C3D+P: A Summarization Method for Interactive Entity Resolution

Gong Chenga,∗, Danyun Xua, Yuzhong Qua

aState Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210023, PR China

Abstract

Entity resolution is a fundamental task in data integration. Recent studies of this problem, including active learning, crowdsourcing, and pay-as-you-go approaches, have started to involve human users in the loop to carry out interactive entity resolution tasks, namely to invite human users to judge whether two entity descriptions refer to the same real-world entity. This process of judgment requires tool support, particularly when entity descriptions contain a large number of features (i.e. property-value pairs). To facilitate judgment, in this article, we propose to select, from entity descriptions, a subset of critical features as a summary to be shown and judged by human users. Features preferred to be selected are those that reflect the most commonalities shared by and the most conflicts between the two entities, and that carry the largest amount of characteristic and diverse information about them. Selected features are then grouped and ordered to improve readability and further speed up judgment. Experimental results demonstrate that summaries generated by our method help users judge more efficiently (3.57–3.78 times faster) than entire entity descriptions, without significantly hurting the accuracy of judgment. The accuracy achieved by our method is also higher than those achieved by existing summarization methods.

Keywords: entity summarization, instance matching, interactive entity resolution, object consolidation, semi-automatic data integration

1. Introduction

Data integration, dealing with the heterogeneity of data from different sources, has been a hot research topic for decades, where a fundamental task is to find entity descriptions (or records) that refer to the same real-world entity, which is known as entity resolution, or instance matching or object consolidation in the field of Semantic Web [13], or record linkage or duplication record detection in the field of database [4].

Whereas a wide variety of automatic approaches to entity resolution have been proposed, in consideration of the complexity of the problem, recent studies have started to involve human users in the loop. In particular, they solicit user feedback for judging whether two entity descriptions found by automatic approaches refer to the same real-world entity. Since user feedback is obtained at a cost, existing efforts mainly focus on reducing the cost or giving added incentives to users in various ways. For instance, active learning approaches [15, 12] seek to reduce the amount of user feedback by picking a small number of entity descriptions that, after being judged, will provide the most benefit to the learner. Crowdsourcing approaches [21, 22] directly pay human users to judge, but try to achieve both high-quality results of judgments and a low cost. Pay-as-you-go approaches [11, 9] attract human users to judge by using the results of judgments to facilitate their original tasks, e.g. Web search.

All the above approaches require human users to judge whether two entity descriptions refer to the same real-world entity, which is called an interactive entity resolution task. However, to the best of our knowledge, very little attention has been given to this process of judgment, which not surprisingly requires tool support. D-Dupe [10] is one of the few related efforts, which highlights all the similar features (i.e. property-value pairs) of the two
entities when presenting their descriptions. However, problems still arise when entity descriptions become lengthy, e.g. containing several hundred features, which will overload users with too much information and finally increase the cost.

To address this issue, in this article, we propose to automatically generate a compact summary of two entity descriptions to be judged by human users. Such length-limited summaries are expected to carry critical information for judgment and thus help users judge more efficiently without significantly hurting the accuracy of judgment. To achieve this, we firstly select, from entity descriptions, a subset of features that reflect the most commonalities shared by and the most conflicts between the two entities, and that carry the largest amount of characteristic and diverse information about them. Then we group and order the selected features in order to improve readability and further speed up judgment. To summarize, our contribution is fourfold.

- We propose an abstract framework for selecting features called C3D, which prefers four kinds of features: common, conflicting, characteristic, and diverse ones.
- We present a specific implementation of C3D for selecting features into a summary. In particular, we integrate its four aspects by formulating and solving a multi-objective optimization problem.
- We propose to present a summary by grouping common or conflicting features together and then ordering groups to show more useful ones earlier.
- We empirically evaluate our method by inviting human users to carry out interactive entity resolution tasks based on entity descriptions or their summaries generated by different methods.

Compared with our previous work [23], this article extends it in the following directions.

- We make several improvements to the selection of features (cf. Sect. 7). As demonstrated by the experimental results, the summaries generated by our new method helped human users judge more accurately than those generated by our previous method, and the difference was statistically significant.
- In addition to the selection of features, we also give our attention to the presentation of selected features and propose an approach to grouping and ordering features. As demonstrated by the experimental results, this new presentation helped human users judge more efficiently than a straightforward presentation, and the difference was statistically significant.
- We carry out a more systematic evaluation by comparing our method with more other summarization methods and performing statistical significance tests.

The remainder of this article is structured as follows. Section 2 gives some preliminaries. Section 3 describes the C3D framework for selecting features. Section 4 presents a specific implementation of C3D. Section 5 describes the presentation of features. Section 6 presents the experiments. Section 7 discusses related work. Section 8 concludes the article with future work.

2. Preliminaries

Let \( \Sigma_E, \Sigma_P, \Sigma_C \), and \( \Sigma_D \) be the disjoint sets of all entities, properties, classes (i.e. entity types), and data values (e.g. integers, strings), respectively. An entity has one or more properties. A value of a property can be a data value, an entity, or a class, and we define \( \Sigma_V = \Sigma_D \cup \Sigma_E \cup \Sigma_C \). In particular, a special property called type gives the types of an entity, i.e. the classes that the entity belongs to. Classes usually constitute a subsumptive hierarchy, called a class hierarchy, which characterizes the subclass-superclass relation between them, denoted by \( c_i \sqsubseteq_C c_j \) if \( c_i \) is a subclass of \( c_j \). Analogously, properties constitute a property hierarchy characterizing the subproperty-superproperty relation, denoted by \( p_i \sqsubseteq_P p_j \) if \( p_i \) is a subproperty of \( p_j \). One typical implementation of the above general data model is the Resource Description Framework (RDF) and RDF Schema (RDFS). However, our discussion in the remainder of this article will be not limited to any specific implementation but based on the general data model, though our experiments will be carried out on RDF data.

The description of an entity \( e \) consists of a set of property-value pairs (a.k.a. features) denoted by \( d(e) \subseteq (\Sigma_P \times \Sigma_V) \). For convenience, given a feature \( f \in d(e) \), let \( p(f) \) and \( v(f) \) return the property and the value of this feature, respectively. As a
running example in this article, Table 1 presents the descriptions of three entities, in which TimBL and TBL refer to the same person in the real world, namely Sir Tim Berners-Lee, the inventor of the World Wide Web (WWW), whereas Wendy refers to a different one. So TimBL and TBL are called a match, whereas TimBL and Wendy are called a non-match. In this article, entities, properties, and classes are printed in italic, and data values are in quotes.

Each entity, property, and class is assumed to have a human-readable name, for instance, in our running example, “Tim Berners-Lee” for TBL, “given name” for given name, and “People From London” for PeopleFromLondon. When presenting a feature to human users, for entities, properties, and classes, their names will be shown; and for data values, their string forms will be shown. The length of a feature, denoted by \( l(f) \), is defined as the total number of characters in the name of \( p(f) \) and in the name or string form of \( v(f) \), e.g. \( l(\langle \text{gender, “Male”} \rangle) = 6 + 4 = 10 \).

Given the descriptions of two entities \( e_i \) and \( e_j \), a summary of them consists of a subset of features selected from each of them, or more formally, \( (S_i, S_j) \) subject to \( S_i \subseteq d(e_i) \) and \( S_j \subseteq d(e_j) \). Given a character limit denoted by \( C \), a feasible summary \( \langle S_i, S_j \rangle \) is one that satisfies

\[
\sum_{f_m \in S_i} l(f_m) + \sum_{f_n \in S_j} l(f_n) \leq C. \tag{1}
\]

In practical applications, features in entity descriptions are often too long to be completely selected and presented without exceeding the application-specific character limit. So in the next section, we will discuss what kinds of features are to be selected to form a feasible summary that is effective in facilitating interactive entity resolution.

3. The C3D Framework for Selecting Features

We propose an abstract framework for selecting features called C3D, which prefers four kinds of features: common, conflicting, characteristic, and diverse ones. In this section, we will describe the four aspects at a high level, illustrated with our running example in Table 1 and based on several low-level measures that will be detailed in the next section. An integration of the four aspects into a specific approach to selecting features will be presented in the next section.

3.1. Common Features

One way that human users identify a match is to seek common features shared by the two entities. For instance, in Table 1, TimBL and TBL have the same name and gender, and thus are likely to refer to the same person in the real world. Such common features shared by the two given entities are useful in indicating a match, and thus are preferred to be selected and put in the summary. However, identifying common features is not a trivial task due to the heterogeneity of data. Besides, not all the common features are equally useful in indicating a match. In the following, we will address these issues from three angles: comparability between properties, similarity between values, and distinguishing ability of properties. Finally we will integrate them and measure to what extent a summary reflects commonalities shared by the two entities.

3.1.1. Comparability between Properties

Before identifying common features, we need to identify common properties. However, entities may have syntactically different but semantically equivalent properties, e.g. TimBL’s gender and TBL’s sex in Table 1. Besides, properties may describe not exactly the same but overlapping aspects, e.g.
TimBL’s given name and TBL’s name. In a heterogeneous environment, it is more practical to seek such comparable properties rather than exactly the same ones. To indicate which properties are comparable and to what extent, let $\text{comp}(p_i, p_j) \in [0,1]$ be the comparability between two properties $p_i$ and $p_j$. Intuitively, the comparability between two semantically equivalent properties such as gender and sex should be high. The comparability between two semantically overlapping properties such as given name and name should also be considerably high. The comparability between two properties describing completely different aspects such as gender and given name should be very low, if not 0.

We will give a specific implementation of $\text{comp}$ in Sect. 4.2.

3.1.2. Similarity between Values

Also due to the heterogeneity of data, comparable properties may have syntactically different but semantically equivalent or similar values. Let $\text{sim}(v_i, v_j) \in [-1,1]$ be the similarity between two property values $v_i$ and $v_j$. Intuitively, the similarity between two semantically equivalent values such as “male” and “Male” in Table 1 should be as high as 1. The similarity between two similar values such as “Berners-Lee” and “Tim Berners-Lee” should also be positive. The similarity between two dissimilar values such as Scientist and “Male” should be negative, and they hardly reflect commonalities shared by the two entities. We will give a specific implementation of $\text{sim}$ in Sect. 4.2.

3.1.3. Distinguishing Ability of Properties

Not all the features having comparable properties and similar values may be equally useful in indicating a match. For instance, in Table 1, TimBL and TBL share a common name and a common gender, the former of which is a much stronger indicator because very few people share a common name but a lot of people share a common gender, or in other words, name has a greater ability to distinguish a person from others than gender. Let $\text{dist}(p_i) \in [0,1]$ be such distinguishing ability of a property $p_i$. Intuitively, the distinguishing ability of properties like twitterName should be as high as 1 because it is usually assumed that a twitter account is used by only one person so that two people sharing a common twitter account should be the same one.\(^1\) The distinguishing ability of name should also be considerably high. The distinguishing ability of gender should be very limited. We will give a specific implementation of $\text{dist}$ in Sect. 4.2.

3.1.4. Integration

Given two entities $e_i$ and $e_j$, by integrating the comparability between properties ($\text{comp}$), similarity between values ($\text{sim}$), and distinguishing ability of properties ($\text{dist}$), for two features $f_m \in d(e_i)$ and $f_n \in d(e_j)$ having similar values, i.e. satisfying $\text{sim}(v(f_m), v(f_n)) > 0$, we define their strength of reflecting commonalities as

$$\text{rcm}(f_m, f_n) = \text{comp}(p(f_m), p(f_n)) \cdot \frac{\text{sim}(v(f_m), v(f_n)) + 2 \cdot \text{dist}(p(f_m)) \cdot \text{dist}(p(f_n))}{\text{dist}(p(f_m)) + \text{dist}(p(f_n))},$$

where in case $f_m$ and $f_n$ have different properties having possibly different distinguishing ability, their harmonic mean is used.

Finally, to reflect the most commonalities shared by $e_i$ and $e_j$, we aim to select their features and form a feasible summary $\langle S_i, S_j \rangle$ that maximizes the total strength of reflecting commonalities:

$$\text{comm}(\langle S_i, S_j \rangle) = \frac{\sum_{(f_m, f_n) \in (S_i \times S_j)} \text{rcm}(f_m, f_n)}{\text{sim}(v(f_m), v(f_n)) > 0}.$$

3.2. Conflicting Features

A summary only reflecting commonalities shared by the two entities in a non-match could be misleading. For instance, in Table 1, TBL and Wendy share two common features about type, though they refer to different people in the real world. A commonality-only summary of their descriptions may consist of only these two common features, and thus mislead users into thinking that the two entities form a match. Actually, TBL’s sex is “Male” whereas Wendy’s sex is “Female”, based on which human users can quickly identify this non-match because the two features conflict. Such conflicting features between two given entities are useful in indicating a non-match, and thus are preferred to be selected and put in the summary. However, similar

\(^1\)Such properties are also known as inverse functional properties in the Web Ontology Language (OWL).
to the identification of common features discussed in Sect. 3.1, identifying conflicting features is not a trivial task either, due to the heterogeneity of data and the fact that not all the conflicting features are equally useful in indicating a non-match. In the following, we will address these issues from three angles: comparability between properties, dissimilarity between values, and value uniqueness of properties. Finally we will integrate them and measure to what extent a summary reflects conflicts between the two entities.

3.2.1. Comparability between Properties

Similar to the identification of common features discussed in Sect. 3.1, before identifying conflicting features, we need to identify comparable properties. We reuse the comparability measure $\text{comp}$ introduced previously.

3.2.2. Dissimilarity between Values

Conflicting features have comparable properties but dissimilar values. We reuse the similarity measure $\text{sim}$ introduced in Sect. 3.1, which returns negative values in $[-1, 0)$ for dissimilar property values, and larger absolute values of $\text{sim}$ for more dissimilar ones.

3.2.3. Value Uniqueness of Properties

Not all the features having comparable properties and dissimilar values are conflicting. For instance, in Table 1, TBL is a PeopleFromLondon and Wendy is a RoyalSocietyFellow. These two features have a common property, namely type, and very dissimilar values, but they are not really conflicting since actually both TBL and Wendy share these two features. In fact, for two comparable properties both having multiple values, we are likely to obtain many pairs of features that have dissimilar values, which not necessarily reflect conflicts between the two entities. Therefore, to seek conflicting features, we focus on the properties taking exactly one value. More formally, given an entity $e$ and a property $p_i$, let $\text{freq}(p_i, e)$ be the frequency of $p_i$ in $d(e)$, namely the number of features in $d(e)$ that have $p_i$ as their property:

$$\text{freq}(p_i, e) = |\{f \in d(e) : p(f) = p_i\}|.$$  (4)

We focus on the properties satisfying $\text{freq} = 1$.

Further, not all the features having comparable properties satisfying $\text{freq} = 1$ and having dissimilar values are equally useful in indicating a non-match. For instance, in Table 1, TBL and Wendy have different values of sex and different values of founded. The former is a strong indicator because one person can have only one value of sex, and people sharing different sex must be different. The latter is a weak indicator because one person may have founded multiple organizations. The reason why TBL and Wendy have different values of founded could be that entity descriptions are incomplete, which is reasonable particularly under the open world assumption and is often the case in practice. Actually, in this example, Tim Berners-Lee is also a co-founder of WSRI, which however is missing from TBL’s description. To indicate how likely a property $p_i$ truly takes only one value for an entity in the real world, let $\text{uniq}(p_i) \in [0, 1]$ be $p_i$’s value uniqueness. Intuitively, the value uniqueness of properties like sex should be as high as 1 because each person has one unique value of sex. The value uniqueness of invented and founded should not be very high because it is possible that one person invented multiple things or founded multiple organizations. We will give a specific implementation of $\text{uniq}$ in Sect. 4.2.

3.2.4. Integration

Given two entities $e_i$ and $e_j$, by integrating the comparability between properties ($\text{comp}$), dissimilarity between values ($\text{sim}$), and value uniqueness of properties ($\text{uniq}$), for two features $f_m \in d(e_i)$ and $f_n \in d(e_j)$ satisfying $\text{freq}(p(f_m), e_i) = \text{freq}(p(f_n), e_j) = 1$ and having dissimilar values, i.e. satisfying $\text{sim}(v(f_m), v(f_n)) < 0$, we define their strength of reflecting conflicts as

$$\text{ref}(f_m, f_n) = \text{comp}(p(f_m), p(f_n)) \cdot |\text{sim}(v(f_m), v(f_n))| \cdot \frac{2 \cdot \text{uniq}(p(f_m)) \cdot \text{uniq}(p(f_n))}{\text{uniq}(p(f_m)) + \text{uniq}(p(f_n))},$$  (5)

where in case $f_m$ and $f_n$ have different properties having different value uniqueness, their harmonic mean is used.

Finally, to reflect the most conflicts between $e_i$ and $e_j$, we aim to select their features and form a feasible summary $\langle S_i, S_j \rangle$ that maximizes the total

---

2Such properties are also known as functional properties in OWL.
strength of reflecting conflicts:

\[
\text{conf1}(\langle S_i, S_j \rangle) = \sum_{(f_m, f_n) \in (S_i \times S_j) \text{ freq}(f_m, e_i) = 1 \text{ freq}(f_n, e_j) = 1 \text{ sim}(v(f_m), v(f_n)) < 0} \text{ref}(f_m, f_n) .
\]

\[
(6)
\]

3.3. Characteristic Features

We have discussed the usefulness of common and conflicting features in indicating a match or non-match. Both of them help human users make judgments based solely on the two entity descriptions. However, human users may happen to have some knowledge of these entities, from which they may draw inferences about whether the two entities form a match or non-match. For instance, if a user has some knowledge of the WWW, she can infer, from the feature \(\text{is director of, W3C}\), that TimBL refers to Tim Berners-Lee, the inventor of the WWW. Then, she can infer, from the feature \(\text{invented, WWW}\), that TBL and TimBL refer to the same person. It is worth noting that the two features used here have incomparable properties, and thus reflect neither commonalities nor conflicts according to our discussion in Sect. 3.1 and Sect. 3.2. However, they are still useful because both of them reflect some distinctive characteristics of Tim Berners-Lee, or in other words, the information they carry is sufficient for users to precisely identify the people in the real world that the two entity descriptions refer to. By comparison, the information carried by features about, for instance, type and gender, is insufficient. To indicate how characteristic of a real-world entity a feature \(f\) is, let \(ch(f) \in [0, 1]\) be the characterizing ability of \(f\). Intuitively, the characterizing ability of \(\text{is director of, W3C}\) should be as high as 1 because it precisely indicates that TimBL refers to Tim Berners-Lee. The characterizing ability of \(\text{type, Scientist}\) should not be very high because many people are scientists. The characterizing ability of \(\text{gender, "male"}\) should be very limited because it belongs to around half of the people in the world. We will give a specific implementation of \(ch\) in Sect. 4.2.

To best characterize two entities \(e_i\) and \(e_j\), we aim to select their features and form a feasible summary \(\langle S_i, S_j \rangle\) that maximizes the total characterizing ability:

\[
\text{char}(\langle S_i, S_j \rangle) = \sum_{f_m \in S_i} ch(f_m) + \sum_{f_n \in S_j} ch(f_n) .
\]

(7)

\[3.4. \text{Diverse Features}\]

To fully exploit the capacity of a feasible summary, we expect to remove redundant information and make the summary diverse. For instance, in Table 1, the \(\text{given name}\) and \(\text{alt name}\) of TimBL describe overlapping aspects of the same entity. It is not cost-effective to have both of them in a summary in consideration of the character limit. Such features sharing overlapping information about the same entity are preferred not to be selected and put in the summary together. To indicate how much overlapping information is shared by two features \(f_m\) and \(f_n\), let \(\text{ovlp}(f_m, f_n) \in [0, 1]\) be the information overlap between \(f_m\) and \(f_n\). Intuitively, the information overlap between features having comparable properties and/or similar values such as \(\langle \text{given name, "Tim"}\rangle\) and \(\langle \text{alt name, "Tim BL"}\rangle\) should be considerably high. The information overlap between features describing completely different aspects such as \(\langle \text{given name, "Tim"}\rangle\) and \(\langle \text{type, Scientist}\rangle\) should be very low, if not 0. We will give a specific implementation of \(\text{ovlp}\) in Sect. 4.2.

To best diversify a summary of two entities \(e_i\) and \(e_j\), we aim to select their features and form a feasible summary \(\langle S_i, S_j \rangle\) that minimizes the total information overlap, or equivalently, maximizes the total negated information overlap:

\[
\text{div}(\langle S_i, S_j \rangle) = \sum_{f_m, f_{m'} \in S_i} -\text{ovlp}(f_m, f_{m'}) + \sum_{f_n, f_{n'} \in S_j} -\text{ovlp}(f_n, f_{n'}) .
\]

(8)

It is worth noting that we seek features describing overlapping aspects both when identifying diverse features here and when identifying common features in Sect. 3.1. The difference is that, here we seek features in the same entity description that describe overlapping aspects, and prefer not to select them together, whereas in Sect. 3.1 we seek such features in different entity descriptions and prefer to select them.

4. A Specific Implementation of C3D

In this section, firstly we integrate the four aspects in the C3D framework into a specific approach to selecting features by formulating and solving a multi-objective optimization problem. Then we detail the low-level measures used in C3D.
4.1. Multi-objective Optimization

Selecting features and forming a feasible summary according to each of the four aspects in C3D can be viewed as an optimization problem with \textit{comm}, \textit{conf}, \textit{char}, and \textit{div} defined in Eq. (3), (6), (7), and (8), respectively, as the objective function to be maximized. However, generally the four objectives could be conflicting; that is, sometimes no single feasible summary can simultaneously maximize all the four objectives. Optimal summaries need to be formed in the presence of trade-offs between different objectives. One way to solve such a multi-objective optimization problem is to quantify the trade-offs in satisfying the different objectives by formulating a linear scalarization to be maximized:

\[
goodness(S_i, S_j) = \alpha \cdot \text{comm}(S_i, S_j) + \beta \cdot \text{conf}(S_i, S_j) + \gamma \cdot \text{char}(S_i, S_j) + \delta \cdot \text{div}(S_i, S_j),
\]

where \( \alpha, \beta, \gamma, \delta > 0 \) are the weights of the objectives to be tuned in the specific application. Now the problem becomes finding a feasible summary \( \langle S_i, S_j \rangle \) that maximizes the objective function in Eq. (9).

This optimization problem can be reformulated as an instance of the binary quadratic knapsack problem (QKP) [14] as follows. Given two entities \( e_i \) and \( e_j \), we number the features in \( d(e_i) \) and \( d(e_j) \) from \( f_1 \) to \( f_{|d(e_i)|} \) and from \( f_{|d(e_i)|+1} \) to \( f_{|d(e_i)|+|d(e_j)|} \), respectively. By introducing a series of binary variables \( x_m \) to indicate whether feature \( f_m \) is selected into the optimal summary, the problem is reformulated as:

\[
\begin{align*}
\text{maximize} & \sum_{m=1}^{N} \sum_{n=1}^{N} p_{mn} x_m x_n \\
\text{subject to} & \sum_{m=1}^{N} l(f_m) x_m \leq C, \\
& x_m \in \{0, 1\}, \ m = 1, \ldots, N,
\end{align*}
\]

which is defined as:

\[
p_{mn} = \begin{cases} 
\alpha \cdot \text{rcm}(f_m, f_n) & \text{if } f_m \in d(e_i), f_n \in d(e_j), \\
\beta \cdot \text{rcf}(f_m, f_n) & \text{if } f_m \in d(e_i), f_n \in d(e_j), \\
\gamma \cdot \text{ch}(f_m) & \text{if } m = n, \\
\delta \cdot (-\text{ovlp}(f_m, f_n)) & \text{if } f_m, f_n \in d(e_i) \text{ or } d(e_j), \\
0 & \text{otherwise}.
\end{cases}
\]

(11)

A QKP with positive and negative profits (as in our case) does not have any polynomial-time approximation algorithm with a constant approximation ratio unless \( \text{P}=\text{NP} \) [14]. To the best of our knowledge, the heuristic implemented in [25] is a state-of-the-art solution to QKP, which can find good feasible solution in high probability within reasonable time in practice, and thus is to be used in our experiments.

4.2. Implementation of Low-level Measures

Six low-level measures, namely \textit{comp}, \textit{sim}, \textit{dist}, \textit{uniq}, \textit{ch}, and \textit{ovlp}, are used in C3D and are to be specifically implemented. In the following we present a specific implementation of them, which however could be substituted with other implementations.

4.2.1. \textit{comp}

To measure \( \text{comp}(p_i, p_j) \in [0, 1] \), the comparability between two properties \( p_i \) and \( p_j \), we combine a learning-based method and a name-based method.

Firstly, we assume the existence of some matches denoted by \( M \subseteq (\Sigma_E \times \Sigma_E) \) as training data, and learn comparability between properties from them. Specifically, let \( M(p_i, p_j) \) be the subset of \( M \) that use \( p_i \) and \( p_j \) in entity descriptions:

\[
M(p_i, p_j) = \{ (e_s, e_t) : \exists f_m \in d(e_s), f_n \in d(e_t), (p(f_m) = p_i, p(f_n) = p_j) \}.
\]

(12)

Given \( (e_s, e_t) \in M(p_i, p_j) \), let \( \text{sim}(p_i, e_s, p_j, e_t) \in [-1, 1] \) characterize the extent to which each value
of \( p_i \) in \( d(e_s) \) can find a similar value of \( p_j \) in \( d(e_t) \):
\[
\text{sims}(p_i, e_s, p_j, e_t) = \sum_{v_s \in \text{vs}(p_i, e_s)} \frac{\max_{v_t \in \text{vs}(p_j, e_t)} \text{sim}(v_s, v_t)}{|\text{vs}(p_i, e_s)|}
\]
where \( \text{vs}(p_i, e_s) \) returns all the values of \( p_i \) in \( d(e_s) \):
\[
\text{vs}(p_i, e_s) = \{ v_k \in \Sigma_V : \exists f \in d(e_s), (p(f) = p_i, v(f) = v_k) \},
\]
and \( \text{sim} \) is another low-level measure to be detailed later, returning the similarity between two property values and in the range \([-1, 1]\). Then we learn the comparability between \( p_i \) and \( p_j \) from \( M(p_i, p_j) \) by looking at the extent to which the values of \( p_i \) and \( p_j \) can find similar values in each other:
\[
\text{comp}(p_i, p_j) = \frac{1}{2} \left( \frac{\sum_{(e_s, e_t) \in M(p_i, p_j)} \text{sims}(p_i, e_s, p_j, e_t)}{|M(p_i, p_j)|} + \frac{\sum_{(e_s, e_t) \in M(p_i, p_j)} \text{sims}(p_j, e_t, p_i, e_s)}{|M(p_i, p_j)|} \right)
\]
which is in the range \([-1, 1]\).

Secondly, we look at the similarity between the names of \( p_i \) and \( p_j \), and employ the string similarity measure proposed in [17] called ISub. Specifically, let \( \text{isub}(p_i, p_j) \in [-1, 1] \) return the ISub similarity between the names of \( p_i \) and \( p_j \). It is worth noting that such string similarity measures do not disambiguate words or recognize synonyms, and could be substituted with more sophisticated techniques in future work.

Finally, we combine the two methods as follows.
\[
\text{comp}(p_i, p_j) = \begin{cases} 
\max \{ \text{comp}(p_i, p_j), 0 \} & \text{if } M(p_i, p_j) \neq \emptyset, \\
\max \{ \text{isub}(p_i, p_j), 0 \} & \text{otherwise}.
\end{cases}
\]

4.2.2. sim

To measure \( \text{sim}(v_i, v_j) \in [-1, 1] \), the similarity between two property values \( v_i \) and \( v_j \), we mainly consider their string similarity but specially handle numerical values.

Specifically, if both \( v_i \) and \( v_j \) are numerical data values, we compute their similarity as follows.

1. If \( v_i = v_j \), \( \text{sim}(v_i, v_j) = 1 \);
2. otherwise, if \( v_i v_j \leq 0 \), \( \text{sim}(v_i, v_j) = -1 \);
3. otherwise, \( \text{sim}(v_i, v_j) = \frac{\min(|v_i|, |v_j|)}{\max(|v_i|, |v_j|)} \).

In other cases, we treat both \( v_i \) and \( v_j \) as strings; that is, for an entity or class, we take its name, and for a data value, we take its string form. Then we compute their ISub similarity and define \( \text{sim}(v_i, v_j) = \text{isub}(v_i, v_j) \).

4.2.3. dist

To measure \( \text{dist}(p_i) \in [0, 1] \), the distinguishing ability of a property \( p_i \), inspired by [8], we estimate it based on a corpus of entity descriptions denoted by \( \text{ED} \). The idea is that a property will have high distinguishing ability if in general it takes different values in different entity descriptions in \( \text{ED} \), or in other words, it has relatively more distinct values:
\[
\text{dist}(p_i) = \frac{|\bigcup_{d(e) \in \text{ED}} \text{vs}(p_i, e)|}{\sum_{d(e) \in \text{ED}} \text{freq}(p_i, e)}.
\]
where \( \text{vs} \) and \( \text{freq} \) are given by Eq. (14) and (4), respectively.

4.2.4. uniq

To measure \( \text{uniq}(p_i) \in [0, 1] \), the value uniqueness of a property \( p_i \), we straightforwardly estimate it based on a corpus of entity descriptions \( \text{ED} \). A property will have high value uniqueness if on average it takes a small number of values in an entity description in \( \text{ED} \):
\[
\text{uniq}(p_i) = \frac{|\{d(e) \in \text{ED} : \exists f \in d(e), (p(f) = p_i)\}|}{\sum_{d(e) \in \text{ED}} \text{freq}(p_i, e)}
\]
where \( \text{freq} \) is given by Eq. (4).

4.2.5. ch

To measure \( \text{ch}(f) \in [0, 1] \), the characterizing ability of a feature \( f \), we estimate it based on a corpus of entity descriptions \( \text{ED} \) by using information theory. The idea is to compute the normalized amount of self-information contained in the probabilistic event of observing \( f \) in an entity description in \( \text{ED} \). Specifically, a feature will have high characterizing ability if it belongs to a small number of entity descriptions in \( \text{ED} \):
\[
\text{ch}(f) = -\log \left( \frac{|\{d(e) \in \text{ED} : f \in d(e)\}|}{|\text{ED}|} \right) \frac{\log |\text{ED}|}{\log |\text{ED}|}
\]

\[= 1 - \log \left( \frac{|\{d(e) \in \text{ED} : f \in d(e)\}|}{|\text{ED}|} \right) \]
4.2.6. $ovlp$

To measure $ovlp(f_m, f_n) \in [0,1]$, the information overlap between two features $f_m$ and $f_n$, first we exploit the ontological semantics of classes and properties. Specifically, if $p(f_m) = p(f_n) =$ type and $v(f_m) \subseteq_C v(f_n)$ (or $v(f_n) \subseteq_C v(f_m)$), i.e., $f_m$ and $f_n$ give two types of an entity that hold the subclass-superclass relation, we will define $ovlp(f_m, f_n) = 1$ because one of them can be inferred from the other and thus they share maximized overlapping information. Similarly, we will also define $ovlp(f_m, f_n) = 1$ if $v(f_m) = v(f_n)$ and $p(f_m) \subseteq_P p(f_n)$ (or $p(f_n) \subseteq_P p(f_m)$).

In other cases, we look at the $ISub$ string similarity between property names (i.e. $isub$) and the similarity between property values (i.e. $sim$), both of which have been discussed previously:

$$ovlp(f_m, f_n) = \max\{isub(p(f_m), p(f_n)), sim(v(f_m), v(f_n)), 0\}. \quad (20)$$

5. Presenting Features

Having selected features and formed a summary, in this section, we discuss the presentation of these features. A straightforward solution is to show the features selected from each entity description in alphabetical order, as illustrated in Table 2. However, to improve readability, firstly we propose to group features so that common or conflicting features are placed together. Secondly, to further speed up judgment, we order groups so that more useful ones are shown earlier. Table 3 shows an example. In the following, we will elaborate on these two steps.

5.1. Grouping Features

To improve the readability of a summary, we group common or conflicting features together when presenting features because these features are useful in indicating a match or non-match when they are compared with each other. To obtain groups, we construct an undirected bipartite graph as illustrated in Fig. 1, where vertices comprise two disjoint sets corresponding to the features in the summary selected from two entity descriptions, and an edge connects two vertices (i.e. two features) if their strength of reflecting commonalities (cf. Eq. (2)) or strength of reflecting conflicts (cf. Eq. (5)) is larger than a threshold $\epsilon$. Then, features in each non-trivial connected component (i.e. containing more than one vertex) of this graph will form a regular group reflecting some commonalities or conflicts. Besides, all the features in trivial connected components (i.e. containing an isolated vertex) will form a miscellaneous group to be placed at the end of the summary because they do not notably reflect commonalities or conflicts. For instance, the summary presented in Table 3 comprises two regular groups and a miscellaneous group induced by the bipartite graph in Fig. 1.

5.2. Ordering Groups

Further, to improve the efficiency of judgment, we order all the regular groups to show more useful ones earlier. The ranking score of a regular group of features is given by the “profit per unit weight” achieved by these features. Specifically, since a regular group can be viewed as a summary in itself, we compute its ranking score by reusing Eq. (9) and then normalize the result by dividing it by the total length of the features in the group. Finally, we order all the regular groups according to their ranking scores in descending order.

6. Experiments

To evaluate the proposed method, we invited human users to carry out interactive entity resolution tasks by judging whether two entities from different real-world data sets formed a match or non-match. The judgment was based on entity descriptions or their summaries generated by different methods. By examining the accuracy of judgment and the time used, we evaluated the effectiveness of different methods in facilitating interactive entity resolution. We also tested the running time of our method.

6.1. Hypotheses

In the experiments, we mainly aimed to test the following three hypotheses.
Table 2: Presenting features straightforwardly (i.e. in alphabetical order)

<table>
<thead>
<tr>
<th>TimBL TBL TBL</th>
</tr>
</thead>
<tbody>
<tr>
<td>⟨gender, “male”⟩</td>
</tr>
<tr>
<td>⟨given name, “Tim”⟩</td>
</tr>
<tr>
<td>⟨is director of, W3C⟩</td>
</tr>
<tr>
<td>⟨surname, “Berners-Lee”⟩</td>
</tr>
<tr>
<td>⟨name, “Tim Berners-Lee”⟩</td>
</tr>
<tr>
<td>⟨sex, “Male”⟩</td>
</tr>
</tbody>
</table>

Table 3: Presenting features as ordered groups (separated by dotted lines)

<table>
<thead>
<tr>
<th>TimBL TBL TBL</th>
</tr>
</thead>
<tbody>
<tr>
<td>⟨given name, “Tim”⟩</td>
</tr>
<tr>
<td>⟨surname, “Berners-Lee”⟩</td>
</tr>
<tr>
<td>⟨gender, “male”⟩</td>
</tr>
<tr>
<td>⟨is director of, W3C⟩</td>
</tr>
<tr>
<td>⟨name, “Tim Berners-Lee”⟩</td>
</tr>
<tr>
<td>⟨sex, “Male”⟩</td>
</tr>
<tr>
<td>⟨invented, WWW⟩</td>
</tr>
<tr>
<td>⟨founded, W3C⟩</td>
</tr>
</tbody>
</table>

1. Summaries generated by our implementation of C3D help subjects judge matches/non-matches more efficiently than entire entity descriptions, without significantly hurting the accuracy of judgment.

2. Summaries generated by our implementation of C3D help subjects judge matches/non-matches more accurately than those generated by existing methods for summarizing entity descriptions.

3. Summaries presented as ordered groups of features help subjects judge matches/non-matches more efficiently than those presented straightforwardly (i.e. in alphabetical order).

6.2. Data Sets

Entity descriptions from the following three real-world RDF data sets were used in the experiments.

- DBpedia\(^3\) (version 3.9-en) provides four million entity descriptions of various types like persons, places, creative works, and organizations. The following data sets were used: Mapping-based Types, Mapping-based Properties, Titles, Geographic Coordinates, Homepages, Persondata, PND, and YAGO types. Features containing non-English data values were removed because subjects might not understand them.

- GeoNames\(^4\) (version 2013-08-27) provides eight million descriptions of places. The entire data set was used, but features about rdf:s:seeAlso were removed because they might leak the expected results of judgments.

- LinkedMDB\(^5\) (version 2010-01-29) provides tens of thousands of descriptions of films and related entities like actors and directors. The entire data set was used, but features about owl:sameAs were removed because they might leak the expected results of judgments.

The class hierarchy and property hierarchy used in the experiments were obtained from the ontologies used by these data sets.

6.3. Tasks

Interactive entity resolution tasks were established based on two pairs of data sets: DBpedia and GeoNames on descriptions of places, and DBpedia and LinkedMDB on descriptions of films. A task comprised two places (or films), one from DBpedia and the other from GeoNames (or LinkedMDB), that were known to be either a match or a non-match. The subject did not know this result when carrying out the task, and was exactly invited to judge whether the two entities formed a match or non-match. To establish challenging tasks, the Disambiguation links in DBpedia were leveraged to find places (or films) in DBpedia having a common name. For instance, as illustrated in Fig. 2, the name “Paris” was found to refer to 24 places in DBpedia having owl:sameAs links to places in GeoNames, indicating that these

\(^3\)http://dbpedia.org/
\(^4\)http://www.geonames.org/
\(^5\)http://www.linkedmdb.org/
Figure 2: Establishing challenging tasks consisting of matches and non-matches obtained from Disambiguation links and \texttt{owl:sameAs} links in DBpedia.

24 pairs of entities formed 24 matches. The remaining $24^2 - 24 = 552$ combinations formed non-matches. In this way, from DBpedia and GeoNames, 113,587 matches and 743,504 non-matches of places were obtained. From DBpedia and LinkedMDB, 2,915 matches and 580 non-matches of films were obtained. Each of these matches/non-matches formed a task.

6.4. Methods

Subjects judged matches/non-matches based on summaries generated by six methods.

- **NoS** simply returns the entire description of each entity. Features from each entity description are presented straightforwardly (i.e. in alphabetical order).

- **RELIN** [3] is a state-of-the-art method for summarizing entity descriptions for generic purposes. It prefers to mainly select characteristic features from each entity description. Features from each entity description are presented straightforwardly.

- **CD** implements two out of the four aspects in the C3D framework proposed in this article by fixing $\alpha = \beta = 0$ in Eq. (9), namely only preferring characteristic and diverse features. In this regard, CD can be conceived of as a stronger method for summarizing entity descriptions for generic purposes than RELIN. Features from each entity description are presented straightforwardly.

- **Conf** [23] is a preliminary version of C3D. Features from each entity description are presented straightforwardly.

- **C3D** fully implements the C3D framework proposed in this article. Features from each entity description are presented straightforwardly.

- **C3D+P** fully implements the C3D framework proposed in this article. Different from all the above methods, C3D+P presents features in a summary as ordered groups according to Sect. 5.

In C3D and C3D+P, to measure $\text{comp}$ according to Sect. 4.2.1, one thousand matches were randomly selected from each pair of data sets as training data (i.e. $M$). In CD, C3D, and C3D+P, to measure $\text{dist}$, $\text{uniq}$, and $\text{ch}$ according to Sect. 4.2, all the entity descriptions in each data set comprised the corpus for estimation (i.e. $ED$).

The weights $\alpha$, $\beta$, $\gamma$, $\delta$ in Eq. (9) in CD, C3D, and C3D+P, as well as the threshold $\epsilon$ in Sect. 5.1 in C3D+P, needed to be tuned for the specific data sets. To achieve this in the experiments, five matches and five non-matches were randomly selected from each pair of data sets (which were kept separate from those to be used in subsequent experiments), based on which $\alpha$, $\beta$, $\gamma$, $\delta$, $\epsilon$ were empirically tuned according to our subjective assessment of the quality of the summaries generated using different settings. Finally, for places in DBpedia and GeoNames, we set $\alpha = 4$, $\beta = 1$, $\gamma = 1$, $\delta = 2$, $\epsilon = 0.01$; and for films in DBpedia and LinkedMDB, we set $\alpha = 6$, $\beta = 3$, $\gamma = 1$, $\delta = 4$, $\epsilon = 0.01$.

The configuration of RELIN and Conf precisely followed the instructions given by their papers.

### Table 4: Average length of an entity description involved in tasks

<table>
<thead>
<tr>
<th>Data Set</th>
<th>#features</th>
<th>#characters</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBpedia</td>
<td>30.98</td>
<td>575.83</td>
</tr>
<tr>
<td>GeoNames</td>
<td>16.95</td>
<td>404.34</td>
</tr>
<tr>
<td>LinkedMDB</td>
<td>28.49</td>
<td>725.08</td>
</tr>
</tbody>
</table>


In RELIN, CD, Conf, C3D, and C3D+P, the character limit of a summary was set to 140, which is around the (estimated) limit of a common snippet in Google Search. Compared with the average length of an entity description involved in tasks as presented in Table 4, for places in DBpedia and GeoNames, the expected space savings produced by summarization in our experiments was 85.72% (1 - \frac{140}{575.83+394.34}), and for films in DBpedia and LinkedMDB, the expected space savings was 89.24% (1 - \frac{140}{575.83+725.08}).

6.5. Experimental Design

Twenty-three students majoring in computer science and technology were invited to the experiments.

Based on summaries of places in DBpedia and GeoNames generated by one of the six methods, each subject carried out twelve interactive entity resolution tasks. Specifically, the subject firstly carried out two random tasks as a warmup, comprising one match and one non-match, the results of which were not included in subsequent analysis. Then the subject carried out ten random tasks, comprising five matches and five non-matches. In each of these tasks, the subject judged whether the two entities formed a match or non-match, and responded “match”, “non-match”, or “not sure”. The response as well as the time used for judgment were recorded. The subject repeated the above process based on each of the six methods, which were arranged in random order. It is worth noting that the random assignment of tasks in the experiments was controlled to ensure that no task was carried out more than once by a subject.

Finally, the entire process was repeated based on summaries of films in DBpedia and LinkedMDB generated by the six methods.

To conclude, each subject carried out a total of \((2 + 10) \times 6 - 2 = 144\) tasks. The results in subsequent sections were obtained from a total of \(10 \times 6 \times 23 = 2,760\) tasks.

6.6. Evaluation Metrics

Two metrics were measured in the experiments: accuracy and time.

A subject’s judgment would be called accurate if the response was the same as the expected answer, i.e. “match” for a match or “non-match” for a non-match. The accuracy of judgment achieved by a subject based on a method was defined as the proportion of accurate responses and was in the range \([0, 1]\).

Time referred to the average time used by a subject for carrying out one single task based on a method.

Thus, a method would be better if subjects achieved higher accuracy of judgment and used less time based on summaries generated by this method.

6.7. Results

Table 5 shows the mean and standard deviation (SD) of the accuracy of judgment achieved and the time used by all the subjects based on summaries of places in DBpedia and GeoNames generated by each of the six methods. Repeated measures ANOVA \((F \text{ and } p)\) was used to test the statistical significance of the differences between the mean values. When the differences were statistically significant, LSD post-hoc analysis was performed to reveal the differences. Analogously, Table 6 shows these results on films in DBpedia and LinkedMDB.

In both Table 5 and 6, repeated measures ANOVA revealed that the differences in mean accuracy of judgment achieved and in mean time used by the subjects were both statistically significant \((p < 0.01)\). Therefore, in the following, we will directly look at the results of LSD post-hoc analysis.

6.7.1. Testing Hypothesis 1 (C3D versus NoS)

As shown in Table 5, the mean accuracy of judgment achieved by the subjects based on entire descriptions of places in DBpedia and GeoNames (i.e. NoS) and the mean accuracy achieved based on summaries generated by our implementation of C3D were 0.94 and 0.96, respectively. Both of them were very high and, according to LSD post-hoc analysis, no statistically significant \((p < 0.05)\) difference was found between them. Compared with the mean time used by the subjects for carrying out a task based on entire descriptions of places (i.e. NoS), the mean time used based on summaries generated by our implementation of C3D decreased from 27.66 to 10.07 seconds, or was 2.75 times faster. LSD post-hoc analysis revealed that this difference (i.e. C3D < NoS) was statistically significant \((p < 0.05)\). Table 6 provides very similar results on films in DBpedia and LinkedMDB. In particular, tasks were carried out 26.71/10.81 = 2.47 times faster based on C3D than based on NoS, and this difference was also statistically significant \((p < 0.05)\).
Table 5: Accuracy and time on places in DBpedia and GeoNames

<table>
<thead>
<tr>
<th></th>
<th>NoS</th>
<th>RELIN</th>
<th>CD</th>
<th>Conf</th>
<th>C3D</th>
<th>C3D+P</th>
<th>Mean (SD)</th>
<th>F(5,110)</th>
<th>LSD post-hoc (p &lt; 0.05)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy</td>
<td>0.94</td>
<td>0.76</td>
<td>0.90</td>
<td>0.93</td>
<td>0.96</td>
<td>0.97</td>
<td>0.94</td>
<td>19.432</td>
<td>(1) C3D+P &gt; CD,Conf</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>19.432</td>
<td>0.00</td>
<td>(2) C3D &gt; CD &gt; RELIN.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(3) NoS &gt; RELIN.</td>
</tr>
<tr>
<td>Time (s)</td>
<td>27.66</td>
<td>16.40</td>
<td>12.29</td>
<td>10.04</td>
<td>10.07</td>
<td>7.74</td>
<td>27.66</td>
<td>45.569</td>
<td>(1) C3D+P &lt; Conf,C3D</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>45.569</td>
<td>0.00</td>
<td>&lt; CD &lt; RELIN &lt; NoS.</td>
</tr>
</tbody>
</table>

These results support our first hypothesis, that is, summaries generated by our implementation of C3D help subjects judge matches/non-matches more efficiently than entire entity descriptions (2.47–2.75 times faster, being a statistically significant difference), without significantly hurting the accuracy of judgment (i.e. being not a statistically significant difference).

6.7.2. Testing Hypothesis 2 (C3D versus RELIN, CD, and Conf)

Firstly, as shown in Table 5, compared with the mean accuracy of judgment achieved by the subjects based on summaries of places in DBpedia and GeoNames generated by RELIN, which is a state-of-the-art method for summarizing entity descriptions for generic purposes, the mean accuracy achieved based on summaries generated by our implementation of C3D increased from 0.76 to 0.96, or by 0.20. LSD post-hoc analysis revealed that this difference (i.e. C3D > RELIN) was statistically significant (p < 0.05). Table 6 provides very similar results on films in DBpedia and LinkedMDB-B: from 0.73 based on CD to 0.95 based on C3D, or by 0.22, the difference between which was also statistically significant (p < 0.05).

Thirdly, as shown in Table 6, compared with the mean accuracy of judgment achieved by the subjects based on summaries of films in DBpedia and LinkedMDB generated by Conf, which is a preliminary version of C3D, the mean accuracy achieved based on summaries generated by our implementation of C3D increased from 0.90 to 0.95, or by 0.05. LSD post-hoc analysis revealed that this difference (i.e. C3D > Conf) was statistically significant (p < 0.05). Table 5 provides similar results on places in DBpedia and GeoNames: from 0.93 based on Conf to 0.96 based on C3D, or by 0.03, the difference between which was however not statistically significant (p < 0.05).

These results support our second hypothesis, that is, summaries generated by our implementation of C3D help subjects judge matches/non-matches more accurately than those generated by existing methods for summarizing entity descriptions (in particular, increasing the accuracy by 0.15–0.20 compared with RELIN, being a statistically significant difference).
6.7.3. Testing Hypothesis 3 (C3D+P versus C3D)

As shown in Table 5, the mean accuracy of judgment achieved by the subjects based on summaries of places in DBpedia and GeoNames generated by our implementation of C3D and presented straightforwardly and the mean accuracy achieved based on summaries generated by our implementation of C3D and presented as ordered groups of features (i.e. C3D+P) were 0.96 and 0.97, respectively.

Both of them were very high, and according to LSD post-hoc analysis, no statistically significant (p < 0.05) difference was found between them.

Compared with the mean time used by the subjects for carrying out a task based on summaries generated by our implementation of C3D and presented straightforwardly, the mean time used based on summaries generated by our implementation of C3D and presented as ordered groups of features (i.e. C3D+P) decreased from 10.07 to 7.74 seconds, or was 1.30 times faster. LSD post-hoc analysis revealed that this difference (i.e. C3D+P < C3D) was statistically significant (p < 0.05). Table 6 provides very similar results on films in DBpedia and LinkedMDB. In particular, tasks were carried out 10.81/7.07 = 1.53 times faster based on C3D+P than based on C3D, and this difference was also statistically significant (p < 0.05).

These results support our third hypothesis, that is, summaries presented as ordered groups of features help subjects judge matches/non-matches more efficiently than those presented straightforwardly (1.30–1.53 times faster, being a statistically significant difference).

Further, by comparing the results of NoS and C3D+P in Table 5 and 6, LSD post-hoc analysis revealed that no statistically significant (p < 0.05) difference was found between the accuracy achieved based on entity entity descriptions (i.e. NoS) and the accuracy achieved based on summaries generated by our C3D+P, but tasks were carried out from 27.66/7.74 = 3.57 to 26.71/7.07 = 3.78 times faster based on C3D+P than based on NoS, and this difference (i.e. C3D+P < NoS) was statistically significant (p < 0.05). These results more strongly support our first hypothesis.

6.8. Discussion

Apart from testing the three hypotheses, the following observations made from the experimental results also deserve to be noticed.

Firstly, as shown in both Table 5 and 6, the mean accuracy of judgment achieved by the subjects based on entire descriptions of places or films (i.e. NoS) was high but did not reach 1.00. That is, subjects occasionally made inaccurate judgments even based on entire entity descriptions. A major reason is that entity resolution is an inherently difficult task. For instance, there were cases where the two descriptions of a place in DBpedia and GeoNames provided (slightly) different longitudes and latitudes, whereas there were also cases where two different places had very similar names and locations. In such cases, subjects were prone to inaccurate judgments.

Secondly, our implementation of C3D outperformed the methods for summarizing entity descriptions for generic purposes (i.e. RELIN and CD) mainly because features selected into such a generic summary were often not comparable even though they carried a large amount of information. For instance, there were cases where features about the writer and an actor of a film were selected from its description in LinkedMDB, but features about the producer and another actor of this film were selected from its description in LinkedMDB. Due to such informational mismatches, subjects often showed hesitation, then had to make a guess, and thus were prone to inaccurate judgments. This also explained why, according to Table 5 and 6, tasks were carried out significantly slower based on RELIN and CD than based on C3D. By comparison, our implementation of C3D exactly targeted this issue and selected comparable features into a summary, and thus performed better in the experiments.

Thirdly, our implementation of C3D outperformed its preliminary version (i.e. Conf) particularly on films in DBpedia and LinkedMDB. One reason is the improvement made in identifying conflicting features. For instance, the editor property of a film has a very high distinguishing ability since a film usually has only one editor. However, occasionally there were cases where a film had two editors, but in its summary generated by Conf, the feature about one editor of this film was selected from its description in LinkedMDB, and the feature about the other editor was selected from its description in LinkedMDB, thereby seeming to form conflicting features due to the high distinguishing ability of editor and misleading the subjects into judging them a non-match. By comparison, when identifying conflicting features, our implementation of C3D focused on the properties taking exactly one value, and thus fixed the above issue and performed better in the experiments.
Table 7: Average running time (ms) of C3D+P in a task

<table>
<thead>
<tr>
<th>Features</th>
<th>On places</th>
<th>On films</th>
</tr>
</thead>
<tbody>
<tr>
<td>Selecting features</td>
<td>23.43</td>
<td>32.19</td>
</tr>
<tr>
<td>Presenting features</td>
<td>0.10</td>
<td>0.09</td>
</tr>
</tbody>
</table>

6.9. Running Time

Last but not least, we tested the running time of C3D+P on an Intel Xeon E3-1225 v2 with 512MB memory for JVM. Prior to testing, some low-level measures including \textit{comp}, \textit{dist}, \textit{uniq}, and \textit{ch} were precomputed, and all the relevant data was loaded into memory. C3D+P was then applied to generate summaries for 1,000 random tasks on places in DBpedia and GeoNames, and 1,000 random tasks on films in DBpedia and LinkedMDB. Table 7 presents the average running time of selecting and presenting features in a task, showing that our implementation of C3D+P was reasonably fast.

7. Related Work

Since this article proposes a summarization method for interactive entity resolution, in this section we will separately review the literature on entity summarization and on entity resolution.

7.1. Entity Summarization

Summarizing entity descriptions, or entity summarization, has proven to be useful in many applications. In particular, in semantic search engines like Sindice [1] and Falcons [2], entity descriptions in search engine results pages are summarized to mainly indicate their relevance to the keyword query by selecting and showing features that contain query keywords. In a recent work, Zhang et al. [26] used machine learning to rank and select properties (and thus features) for effective search interaction. Different from this line of research, in this article, we study a summarization method for a different task, namely interactive entity resolution. Therefore, our method focuses on not query relevance but how to help human users accurately and efficiently judge matches/non-matches, and thus uses different techniques.

Another line of research aims to summarize entity descriptions for generic use, i.e. not for any specific task. For instance, RELIN [3] employs a random surfer model to rank features mainly based on their informativeness but also considering the relatedness between them. Thalhammer et al. [20] preferred to select the features of an entity that are shared with its nearest neighbors, and they measured the distance between entities based on usage data such as ratings of film entities. DIVERSUM [18] improves the diversity of a summary by not choosing features sharing a common property. Fakas et al. [5, 6, 7] summarized entity descriptions represented as records in relational database by investigating the affinity of relations and their attributes.

The generic summaries generated by these methods can be used for the task of interactive entity resolution. However, as demonstrated by our experimental results, the summaries generated by our method which were specifically for this task helped human users judge matches/non-matches more accurately than generic summaries such as those generated by RELIN.

Research efforts have also been put into evaluation and empirical study of entity summarization. For instance, Thalhammer et al. [19] tried to establish a ground truth for evaluation based on a quiz game. Xu et al. [24] empirically analyzed how humans rank and select features by linking features in entity descriptions in DBpedia to their mentions in the abstracts of the corresponding Wikipedia articles. Implications drawn from the findings may inspire us to improve our method in the future.

7.2. Entity Resolution

Entity resolution, a.k.a. instance matching or object consolidation in the field of Semantic Web [13], or record linkage or duplicate record detection in the field of database [4], refers to the task of finding entity descriptions that refer to the same real-world entity. A large body of work has been devoted to developing automatic approaches to solving this task [4, 13]. Some of them [8, 17] have been incorporated into our method. However, compared with automatic approaches, our method is designed to help human users make a decision rather than to make a decision by itself, and thus pays attention to human factors, e.g. to consider a character limit so as to not overload human users with too much information.

In fact, our method is closely related to many semi-automatic approaches to entity resolution. Among others, active learning approaches [15, 12] seek to pick a set of candidate matches that, after being judged to be true or false, will provide the most benefit to the learner. Crowdsourcing approaches [21, 22] pay a group of human users to judge matches/non-matches, and thus intend to
achieve both high-quality results of judgments and a low cost. Pay-as-you-go approaches [11, 9] consider not only the benefit to the overall quality of entity resolution in the system but also the benefit to the human user’s original task such as Web search. In all these approaches, human users are involved in the loop to judge matches/non-matches, which requires tool support. Summaries generated by our method can exactly facilitate such interactive entity resolution. However, to the best of our knowledge, very little attention has been paid to this problem in the literature. D-Dupe [10] is one of the few related attempts, which simply highlights all the similar features of the two entities when presenting their descriptions. By comparison, our method considers both how to select and how to present features.

Compared with our previous work [23] which is a preliminary version of C3D, in this article, we extend it in several directions. Firstly, we have made several improvements to the selection of features. For instance, when identifying conflicting features, our C3D framework focuses on the properties taking exactly one value. When measuring the similarity between two property values, our implementation of C3D specifically processes numerical data values. When measuring the information overlap between two features, our implementation of C3D exploits the ontological semantics of classes and properties. As demonstrated by our experimental results, the summaries generated by our implementation of C3D helped human users judge matches/non-matches more accurately than those generated by it preliminary version. Secondly, in addition to the selection of features, we have also given our attention to the presentation of selected features and have proposed an approach to grouping and ordering features. As demonstrated by our experimental results, this new presentation helped human users judge matches/non-matches more efficiently than a straightforward presentation.

8. Conclusions

To help human users carry out interactive entity resolution tasks, we have proposed an abstract framework for selecting features from entity descriptions into a summary and have presented a specific implementation. Further, we have proposed to present features in a summary as ordered groups. Experimental results show that summaries generated by our method help users judge matches/non-matches more efficiently than entire entity descriptions, without significantly hurting the accuracy of judgment. The accuracy achieved is also higher than those achieved by existing summarization methods. Therefore, our method well complements many existing semi-automatic approaches to entity resolution such as active learning, crowdsourcing, and pay-as-you-go approaches; the summaries generated by our method can facilitate the process of judgment in these approaches.

We have evaluated our method on place and film entities. Although it is a domain-independent method, its current implementation presented in this article requires configuring the weights of different objectives. In the experiments, we manually tuned these weights based on several random examples, but in the future, we will explore semi-automatic ways of tuning. We will also try to improve our method by experimenting with more sophisticated ways of combining and implementing low-level measures.

Acknowledgments

The authors would like to thank all the participants in the experiments. This work was supported in part by the 863 Program under Grant 2015AA015406, in part by the NSFC under Grants 61223003, 61170068, and 61370019, and in part by the JSNSF under Grant BK2012723.

References


Figure-graph

<given name, “Tim”> <name, “Tim Berners-Lee”>
<surname, “Berners-Lee”> <sex, “Male”>
<gender, “male”> <invented, WWW>
<is director of, W3C> <founded, W3C>