Facilitating Human Intervention in Coreference Resolution with Comparative Entity Summaries

Danyun Xu, Gong Cheng*, and Yuzhong Qu

State Key Laboratory for Novel Software Technology, Nanjing University,
Nanjing 210023, PR China
dyxu@smail.nju.edu.cn, {gcheng,yzqu}@nju.edu.cn

Abstract. A primary challenge to Web data integration is coreference resolution, namely identifying entity descriptions from different data sources that refer to the same real-world entity. Increasingly, solutions to coreference resolution have humans in the loop. For instance, many active learning, crowdsourcing, and pay-as-you-go approaches solicit user feedback for verifying candidate coreferent entities computed by automatic methods. Whereas reducing the number of verification tasks is a major consideration for these approaches, very little attention has been paid to the efficiency of performing each single verification task. To address this issue, in this paper, instead of showing the entire descriptions of two entities for verification which are possibly lengthy, we propose to extract and present a compact summary of them, and expect that such length-limited comparative entity summaries can help human users verify more efficiently without significantly hurting the accuracy of their verification. Our approach exploits the common and different features of two entities that best help indicate (non-)coreference, and also considers the diverse information on their identities. Experimental results show that verification is 2.7–2.9 times faster when using our comparative entity summaries, and its accuracy is not notably affected.

Keywords: #eswc2014Xu, comparative entity summary, coreference resolution, entity consolidation, entity summarization.

1 Introduction

The heterogeneous nature of the Web further motivates the research on data integration, where a primary challenge is how to identify entity descriptions from different data sources that refer to the same real-world entity, which is called coreference resolution, entity consolidation, etc. For instance, DBpedia and GeoNames provide descriptions of many common places but with different identifiers (i.e. URIs); DBpedia and LinkedMDB describe overlapping films.

Apart from a wide variety of automatic methods for solving this problem, recent studies have started to involve human users in this process and, in particular, they solicit user feedback for verifying candidate coreferent entities computed

* Corresponding author.

by automatic methods. Among others, active learning [7] seeks to improve the underlying learning-based approach with a minimized amount of user interaction, e.g. a minimum number of verification tasks; crowdsourcing approaches [13] focus on the assignment of verification tasks to a group of paid users and aim to reduce cost while providing high-quality results; pay-as-you-go approaches [6] promise to provide better information to meet a user's need if, for example, the user helps carry out some verification tasks. Along with these solutions, an equally important issue is how to *improve the efficiency of performing each single verification task*, which has received very little attention. Existing efforts mainly enhance the visualization of entity descriptions [1, 4], but problems arise when entity descriptions are lengthy, e.g. comprising several hundred property-value pairs as in DBpedia, which overload users with too much information and bring about inefficient verification.

To meet the challenge, in this paper, we propose to automatically generate a compact summary of two entity descriptions for verification. Such length-limited *comparative entity summaries* are expected to help users verify more efficiently due to the reduction in length. Meanwhile, if summaries are generated appropriately, the accuracy of verification is expected to be maintained at a high level. To achieve these, our approach extracts, from entity descriptions, the property-value pairs that best reflect the commonality and difference between the two entities, and also carry the largest amount of diverse information on their identities. We will confirm the above two expectations based on real-world verification tasks and, in particular, show that the comparative entity summaries generated by our approach outperform the entity summaries generated for generic use [3].

Our contribution is threefold.

- We propose to help human users more efficiently verify candidate coreferent entities by using comparative entity summaries.
- We analyze and formalize the goodness of a comparative entity summary to optimize from four angles, and transform these objectives into a binary quadratic knapsack problem to solve.
- We implement and evaluate a solution based on real-world verification tasks.

The remainder of this paper is structured as follows. Section 2 gives the problem statement. Section 3 defines the goodness of a summary. Section 4 describes how to generate a good summary. Section 5 presents the experiments. Section 6 discusses related work. Section 7 concludes the paper with future work.

2 Problem Statement

Let $\Sigma_E, \Sigma_C, \Sigma_P, \Sigma_L$ be the sets of all entities, classes, properties, and literals, respectively; and let $\Sigma_V = \Sigma_E \cup \Sigma_C \cup \Sigma_L$, i.e. the set of all possible property values. The description of an entity e, denoted by $d(e) \subseteq (\Sigma_P \times \Sigma_V)$, comprises a set of property-value pairs (a.k.a. features [3]) extracted from RDF data; in fact, an entity and a feature together correspond to an RDF triple. For convenience, given a feature $f \in d(e)$, let $p(f)$ and $v(f)$ return the property and the value of

Table 1: Three Entity Descriptions as a Running Example

TimBL	TBL	Wendy
⟨givenName, "Tim"⟩	⟨name, "Tim Berners-Lee"⟩	⟨fullName, "Wendy Hall"⟩
⟨surname, "Berners-Lee"⟩	⟨type, ComputerScientist⟩	⟨type, ComputerScientist⟩
⟨altName, "Tim BL"⟩	⟨type, RoyalSocietyFellow⟩	⟨type, RoyalSocietyFellow⟩
⟨type, Scientist⟩	⟨sex, "Male"⟩	⟨sex, "Female"⟩
⟨gender, "male"⟩	⟨invented, WWW⟩	⟨birthplace, London⟩
⟨isDirectorOf, W3C⟩	⟨founded, WSRI⟩	⟨founded, WSRI⟩

this feature, respectively. It is worth noting that, in this paper, only the outgoing arcs of an entity in RDF graph are considered as its description for simplicity. However, the extension to both outgoing and incoming arcs is straightforward. As a running example in this paper, Table 1 presents the descriptions of three entities, where TimBL and TBL refer to the same person in the real world, whereas Wendy refers to a different one.

Each entity, class, and property is assumed to have a human-readable name, which could be given by properties like rdfs:label, foaf:name, and dc:title, or otherwise its local name. When presenting a feature to human users, for entities, classes, and properties, we show their names; and for literals, we show their lexical forms. Both names and lexical forms are strings, or in other words, character sequences. The length of a feature f, denoted by $l(f)$, is then naturally defined as the sum of the length of the name of $p(f)$ and the length of the name or lexical form of $v(f)$. For instance, $l(\langle \texttt{gender}, \text{“male”} \rangle) = 6 + 4 = 10$.

Given the descriptions of two entities e_i and e_j, we define a *comparative entity summary*, or a summary for short, as $\langle S_i, S_j \rangle$ subject to $S_i \subseteq d(e_i)$ and $S_j \subseteq d(e_j)$. That is, a summary consists of a subset of features extracted from each of the two entity descriptions. A summary $\langle S_i, S_j \rangle$ is *feasible* if

$$\sum_{f_m \in S_i} l(f_m) + \sum_{f_n \in S_j} l(f_n) \leq C \,, \tag{1}$$

where C is a character limit defined by the specific application.

Among all feasible summaries, we aim to find the optimum one in terms of some criterion, i.e. one that maximizes some objective function called *Goodness*, which will be discussed in the next section.

3 Goodness of a Summary: A High-level Analysis

In this section, we discuss, at a high level and from four angles, what kinds of features constitute a good summary. We will illustrate our ideas with the three entity descriptions presented in Table 1. Our approach to the generation of such good summaries and a detailed implementation of those low-level measures that are used will be introduced in the next section.

3.1 Commonality

In general, human users identify coreferent entity descriptions based on their common features. So a good summary here should include those features that can be found in both of the two entity descriptions $d(e_i)$ and $d(e_j)$ to compare. For instance, in Table 1, `TimBL` and `TBL` have the same name and gender, indicating that they probably have the same referent. However, as illustrated by this case, there are three challenges to be met.

Comparability between Properties. Due to the heterogeneous nature of the Web, it is more practical to seek and exploit semantically (rather than syntactically) equivalent features. In particular, entities may be described using different properties. These properties may have the same meaning but different names, e.g. `gender` and `sex`; they may also describe not exactly the same but overlapping aspects of an entity, e.g. `givenName` and `name`. In light of these, to find semantically equivalent features, we need to firstly identify which properties are comparable and to what extent. Given two properties p_i and p_j, we use $comp(p_i, p_j) \in [0, 1]$ to denote their *comparability*. Intuitively, $comp(\texttt{gender},\texttt{sex})$ should be as high as 1; $comp(\texttt{givenName},\texttt{name})$ should also be considerably high; for many other pairs of properties such as `sex` and `givenName`, their comparability should be very low, if not 0.

Similarity between Values. For the same reason, we need to measure the *similarity* between two property values v_i and v_j, denoted by $sim(v_i, v_j) \in [-1, 1]$, where 1 indicates they are exactly the same and -1 indicates completely different. For instance, the similarity between "male" and "Male" should be as high as 1; "Berners-Lee" and "Tim Berners-Lee" should also be similar to each other; however, "male" and "Berners-Lee" should be dissimilar. In particular, those pairs of values having positive similarity are of interest to us here.

Likeness to an IFP. Not all the properties are equally useful in indicating coreference. For instance, sharing a common gender is a much weaker indicator than sharing a common name. One extreme is inverse functional properties (IFP) in OWL such as the mailbox of a person, which is one of the strongest properties in indicating coreference because two people sharing a common mailbox must be coreferent according to the semantics of IFP defined in OWL. However, most properties are not defined as IFP, but they indeed exhibit different abilities to indicate coreference, e.g. name being stronger than gender. So for each property p, we use $ifp(p) \in [0, 1]$ to denote its *likeness to an IFP*. Intuitively, $ifp(p) = 1$ if p is exactly an IFP; $ifp(\texttt{name})$ should also be considerably high and, in particular, much higher than $ifp(\texttt{gender})$.

With these three measures, given a pair of features f_m and f_n satisfying $sim(v(f_m), v(f_n)) > 0$, we define their *strength of indicating coreference* as

$$ind_C(f_m, f_n) = comp(p(f_m), p(f_n)) \cdot sim(v(f_m), v(f_n)) \cdot \frac{2 \cdot ifp(p(f_m)) \cdot ifp(p(f_n))}{ifp(p(f_m)) + ifp(p(f_n))}, \tag{2}$$

where in case f_m and f_n have different properties, the harmonic mean of their likeness to an IFP is used.

Finally, we aim to find a feasible summary $\langle S_i, S_j \rangle$ that reflects the most *commonality*:

$$COMM(\langle S_i, S_j \rangle) = \sum_{\substack{\langle f_m, f_n \rangle \in (S_i \times S_j) \\ sim(v(f_m), v(f_n)) > 0}} ind_C(f_m, f_n) \,. \tag{3}$$

3.2 Difference

A summary only reflecting commonality may be one-sided. For instance, for `TBL` and `Wendy` in Table 1, a "commonality-only" summary probably only includes the three common features shared by them. As a result, a human user is likely to be misled and judge them as coreferent. The problem resides in the fact that such a summary fails to show the difference between them. To fix it, we propose to also choose dissimilar features that can help human users identify non-coreferent entity descriptions. To achieve this, similar to the discussion of commonality, we also consider three factors.

Comparability between Properties. Dissimilar features make sense only when they have comparable properties. Here we reuse the measure *comp* introduced previously.

Dissimilarity between Values. To show the difference between entity descriptions, we need to choose dissimilar values (of comparable properties). Since the measure *sim* previously introduced is in the range $[-1, 1]$ and negative values indicate dissimilarity, here we only consider those pairs of values having negative similarity, and more dissimilar ones are with larger absolute values of *sim*.

Likeness to a FP. Not all the properties are equally effective in indicating non-coreference. For instance, in Table 1, although `TBL`'s `ComputerScientist` and `Wendy`'s `RoyalSocietyFellow` are dissimilar `type`s, it should not be regarded as an indicator of non-coreference; in fact, their `type` properties take exactly the same values. The problem is caused by the multiple values of a property. Actually, the fewer values a property can take, the stronger indicator it is. One extreme is functional properties (FP) in OWL such as the gender of a person, which is one of the most effective properties in indicating non-coreference because one person can have only one gender according to the semantics of FP defined in OWL so that two people sharing different genders must be non-coreferent. Since most properties are not defined as FP, for each property p, we use $fp(p) \in [0, 1]$ to quantify its *likeness to a FP* and characterize its ability to indicate non-coreference. Intuitively, $fp(p) = 1$ if p is exactly a FP; $fp(\texttt{type})$ should not be very high because an entity often has several types.

Then, given a pair of features f_m and f_n satisfying $sim(v(f_m), v(f_n)) < 0$, similar to Eq. (2), we define their *strength of indicating non-coreference* as

$$ind_{NC}(f_m, f_n) = comp(p(f_m), p(f_n)) \cdot |sim(v(f_m), v(f_n))| \cdot \frac{2 \cdot fp(p(f_m)) \cdot fp(p(f_n))}{fp(p(f_m)) + fp(p(f_n))} \,, \tag{4}$$

where in case f_m and f_n have different properties, the harmonic mean of their likeness to a FP is used.

Finally, we aim to find a feasible summary $\langle S_i, S_j \rangle$ that reflects the most *difference*:

$$DIFF(\langle S_i, S_j \rangle) = \sum_{\substack{\langle f_m, f_n \rangle \in (S_i \times S_j) \\ sim(v(f_m), v(f_n)) < 0}} ind_{NC}(f_m, f_n)\,. \tag{5}$$

3.3 Information on Identity

Sometimes two features not explicitly related to each other may also help human users identify coreferent entity descriptions. For instance, in Table 1, ⟨`isDirectorOf`, `W3C`⟩ in `TimBL`'s description and ⟨`invented`, `WWW`⟩ in `TBL`'s description are weak in indicating both coreference and non-coreference according to Eq. (2) and (4), respectively, because their properties, `isDirectorOf` and `invented`, are not comparable. However, if a human user has some knowledge of the World Wide Web, from these two features she can infer that `TimBL` and `TBL` should both refer to Tim Berners-Lee, thereby being coreferent. The inference actually hinges on the fact that these two features can both precisely reflect the identities of these two entities. In other words, both of them carry a sufficiently large amount of information on the identity of an entity.

More generally, we use $inf(f) \in [0, 1]$ to denote the *amount of information on identity* carried by f. Intuitively, $inf(f) = 1$ if f uniquely indicates the identity of an entity; for instance, the director of W3C must be Tim Berners-Lee. By contrast, ⟨`type`, `ComputerScientist`⟩ carries a relatively small amount of information on identity because many people are computer scientists; and ⟨`gender`, "male"⟩ provides very little information. Finally, we aim to find a feasible summary $\langle S_i, S_j \rangle$ that carries the largest *amount of information on identity*:

$$INF(\langle S_i, S_j \rangle) = \sum_{f_m \in S_i} inf(f_m) + \sum_{f_n \in S_j} inf(f_n)\,. \tag{6}$$

3.4 Diversity of Information

Features in an entity description may share overlapping aspects, e.g. the `givenName` and `altName` of `TimBL` in Table 1. Selecting such overlapping features into a summary will lead to information redundancy. To fully exploit the capacity of a feasible summary, we expect it to provide information that is as diverse as possible. To achieve this, we use $ovlp(f_m, f_n) \in [0, 1]$ to denote the *overlap* between two features f_m and f_n. For instance, the overlap between ⟨`givenName`, "Tim"⟩ and ⟨`altName`, "Tim BL"⟩ is considerably large, whereas ⟨`type`, `Scientist`⟩ and ⟨`gender`, "male"⟩ appear to share no overlap in information. A diverse summary is one containing features sharing small overlap. Therefore, we aim to find a feasible summary $\langle S_i, S_j \rangle$ that maximizes the *diversity of information* it carries:

$$DIV(\langle S_i, S_j \rangle) = \sum_{f_m, f_{m'} \in S_i} -ovlp(f_m, f_{m'}) + \sum_{f_n, f_{n'} \in S_j} -ovlp(f_n, f_{n'})\,. \tag{7}$$

3.5 Goodness

In general, the four objective functions, namely $COMM$, $DIFF$, INF, and DIV, can be conflicting, i.e., sometimes no single feasible summary can simultaneously optimize each objective. To solve this multi-objective optimization problem, one common way of quantifying the trade-offs in satisfying different objectives is to maximize a linear scalarization:

$$\begin{aligned} Goodness(\langle S_i, S_j \rangle) =& \alpha \cdot COMM(\langle S_i, S_j \rangle) + \beta \cdot DIFF(\langle S_i, S_j \rangle) \\ &+ \gamma \cdot INF(\langle S_i, S_j \rangle) + \delta \cdot DIV(\langle S_i, S_j \rangle) \,, \end{aligned} \tag{8}$$

where $\alpha, \beta, \gamma, \delta > 0$ are weights to be tuned in the specific application. We will solve this scalarization in the next section.

4 Generation of a Good Summary

In this section, firstly we introduce how we find a feasible summary that can maximize the scalarization in Eq. (8) by using the binary quadratic knapsack model. Then, we describe our implementation of those low-level measures invoked in the four objective functions.

4.1 Problem Transformation and Solution

The scalarization in Eq. (8) exactly fits the binary quadratic knapsack problem (QKP) [8]. Specifically, given two entities e_i and e_j, we number the features in $d(e_i)$ and $d(e_j)$ from 1 to $|d(e_i)|$ and from $|d(e_i)| + 1$ to $N = |d(e_i)| + |d(e_j)|$, respectively. By introducing a series of binary variables x_m to indicate whether feature f_m is selected into the optimum summary, the problem is formulated as:

$$\begin{aligned} &\text{maximize} \sum_{m=1}^{N} \sum_{n=m}^{N} p_{mn} x_m x_n \\ &\text{subject to} \sum_{m=1}^{N} l(f_m) x_m \leq C \,, \\ &\qquad\qquad x_m \in \{0, 1\}, \, m = 1, \ldots, N \,, \end{aligned} \tag{9}$$

where $l(f_m)$ and C (cf. Eq. (1)) are regarded as the "weight" of feature f_m and the "capacity" of the knapsack, respectively, and "profit" p_{mn} is defined as:

$$p_{mn} = \begin{cases} \alpha \cdot ind_C(f_m, f_n) & \text{if } f_m \in d(e_i), f_n \in d(e_j), sim(v(f_m), v(f_n)) > 0, \\ 0 & \text{if } f_m \in d(e_i), f_n \in d(e_j), sim(v(f_m), v(f_n)) = 0, \\ \beta \cdot ind_{NC}(f_m, f_n) & \text{if } f_m \in d(e_i), f_n \in d(e_j), sim(v(f_m), v(f_n)) < 0, \\ \gamma \cdot inf(f_m) & \text{if } m = n, \\ -\delta \cdot ovlp(f_m, f_n) & \text{otherwise.} \end{cases} \tag{10}$$

Since QKP is strongly NP-hard, we cannot expect to find a fully polynomial-time approximation scheme (FPTAS) unless P=NP. Among heuristic methods that are of interest to practical applications, to the best of our knowledge, a GRASP-based implementation presented in [14] performs the best both in the quality of solutions and in running time. In our experiments, we use this implementation to find near-optimum summaries.

4.2 Implementation of Low-level Measures

Six low-level measures, namely *comp*, *sim*, *ifp*, *fp*, *inf*, and *ovlp*, are invoked in the four objective functions. In the following, we present just one way of implementing them, which is not the core contribution of this paper and can certainly be substituted with others.

ISub [10] returns the similarity between two strings, which is in the range $[-1, 1]$, where 1 and -1 indicate completely similar and dissimilar, respectively. Given two properties or two property values, let *isub* return the ISub similarity between their names or lexical forms. The similarity between property values v_i and v_j is then simply given by

$$sim(v_i, v_j) = isub(v_i, v_j)\,.$$

Analogously, the overlap between features f_m and f_n is defined as

$$ovlp(f_m, f_n) = \max(isub(p(f_m), p(f_n)), isub(v(f_m), v(f_n)), 0)\,.$$

To measure the comparability between two properties, we compute their similarity as a surrogate, which has been extensively studied in the field of ontology matching [9]. We use a learning-based method to measure *comp*, which assumes the existence of some pairs of coreferent entities, denoted by $M \subseteq (\Sigma_E \times \Sigma_E)$. For two properties p_i and p_j, we use the subset of M that are relevant to them:

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

Then, we look at, within these coreferent entity descriptions, to what extent the values of p_i and p_j can find good matches in each other:

$$comp_L(p_i, p_j) = \frac{1}{|M(p_i, p_j)|} \cdot \sum_{\langle e_s, e_t \rangle \in M(p_i, p_j)} \frac{1}{2}(as(p_i, p_j, e_s, e_t) + as(p_j, p_i, e_t, e_s))$$

$$as(p_i, p_j, e_s, e_t) = \frac{1}{|V(p_i, e_s)|} \cdot \sum_{v_k \in V(p_i, e_s)} \max_{v_l \in V(p_j, e_t)} isub(v_k, v_l)\,,$$

where $V(p_i, e_s)$ returns all the values of p_i in $d(e_s)$. Finally, we define

$$comp(p_i, p_j) = \begin{cases} max(comp_L(p_i, p_j), 0) & \text{if } M(p_i, p_j) \neq \emptyset, \\ max(isub(p_i, p_j), 0) & \text{otherwise.} \end{cases}$$

That is, given no training data for p_i and p_j, their ISub similarity will be used.

Given a property p, inspired by [5], we estimate $ifp(p)$ and $fp(p)$ based on a corpus. Specifically, given a corpus of entity descriptions denoted by D, we have

$$ifp(p) = \frac{|\bigcup_{d(e) \in D} \{v(f) : \exists f \in d(e), (p(f) = p)\}|}{\sum_{d(e) \in D} |\{f \in d(e) : p(f) = p\}|}$$

$$fp(p) = \frac{|\{d(e) \in D : \exists f \in d(e), (p(f) = p)\}|}{\sum_{d(e) \in D} |\{f \in d(e) : p(f) = p\}|} \,.$$

The amount of information on identity carried by feature f is also estimated based on D. According to information theory, we have

$$inf(f) = 1 - \frac{\log |\{d(e) \in D : f \in d(e)\}|}{\log |D|} \,.$$

5 Experiments

To evaluate the proposed approach, we invited human users to verify candidate coreferent entities found between real-world data sets by using entity summaries generated by different approaches, and examined the accuracy of their verification and the time used, to test the following hypotheses.

1. Length-limited entity summaries that are appropriately generated help human users verify candidate coreferent entities more efficiently than their entire descriptions, without significantly hurting the accuracy of verification.
2. Comparative entity summaries that consider commonality and difference produce more accurate verification than traditional generic entity summaries.
3. Comparative entity summaries will produce less accurate verification on non-coreferent entities if their difference is not considered.

5.1 Data Sets and Test Cases

The data sets used were DBpedia (3.9-en), GeoNames (2013-08-27), and LinkedMDB (2010-01-29). In particular, for DBpedia, we imported Mapping-based Types, Mapping-based Properties, Titles, Geographic Coordinates, Homepages, Persondata, PND, and YAGO types. We removed RDF triples containing non-English characters. From GeoNames and LinkedMDB, we removed `rdfs:seeAlso` and `owl:sameAs` links, respectively, because they reveal the expected answers.

From DBpedia and GeoNames (places), and from DBpedia and LinkedMDB (films), we obtained both pairs of coreferent entities (based on `owl:sameAs` links in DBpedia) and pairs of non-coreferent entities, called *positive* and *negative* test cases to be verified, respectively. To generate challenging negative cases, we leveraged the Disambiguation links in DBpedia to find the entities in DBpedia that have a common name. For instance, "Paris" may refer to 103 entities in DBpedia, 24 of which have `owl:sameAs` links to GeoNames. From these links we can reliably obtain 24 positive cases and the remaining $24^2 - 24 = 552$ combinations as negative cases. In this way, 113,587 positive and 743,504 negative cases were generated between DBpedia and GeoNames, and 2,915 positive and 580 negative cases were generated between DBpedia and LinkedMDB.

5.2 Participant Approaches

To test the three hypotheses, we designed four approaches to compare in the experiments: **NOSUMM** which simply returns the entire descriptions of two entities without summarization, and three variants of the proposed approach.

- **GENERIC** fixes $\alpha = \beta = 0$ in Eq. (8) so that only the information on identity and the diversity of information are considered. It actually includes and goes beyond the core of RELIN [3], a state-of-the-art approach to generating generic entity summaries that mainly leverages the information on identity carried by each feature.
- **COMPSUMM** considers all the four terms in Eq. (8) and generates comparative entity summaries.
- **COMPSUMM-C** fixes $\beta = 0$ in Eq. (8) so that, compared with COMPSUMM, it also generates comparative entity summaries but ignores the difference between entities.

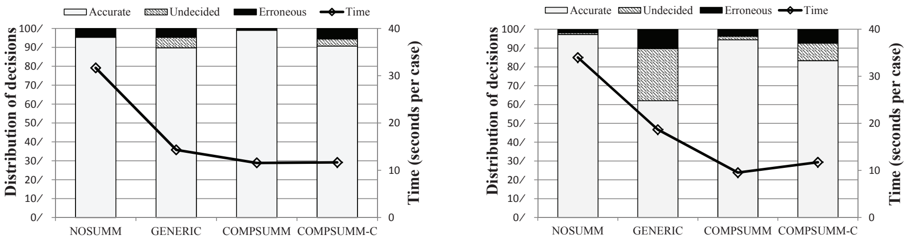
In these approaches, when calculating $comp_L$, we used 1,000 positive cases randomly selected from each pair of data sets as training data (i.e. M). When estimating ifp, fp, and inf, we used all the entity descriptions in each corresponding data set as the corpus (i.e. D).

Tuning the weights $\alpha, \beta, \gamma, \delta$ is a challenge and may depend on the specific data sets. In our experiments, from each pair of data sets, we randomly selected five positive and five negative cases (which were then kept separate from those to be verified in subsequent experiments), and then tuned the weights based on our subjective assessment of the quality of their summaries generated using different weight settings. The weights were tuned one after another. Firstly, γ was fixed to 1, and δ was tuned to obtain GENERIC. Then, α was tuned to obtain COMPSUMM-C. Finally, β was tuned to obtain COMPSUMM. In this way, for DBpedia and GeoNames, we set $\alpha = 6$, $\beta = 4$, $\gamma = 1$, and $\delta = 4$, and for DBpedia and LinkedMDB, we set $\alpha = 8$, $\beta = 4$, $\gamma = 1$, and $\delta = 6$.

5.3 Experimental Design and Evaluation Metrics

We invited 20 students majoring in computer science and technology to the experiments. For each subject, between DBpedia and GeoNames, as a warmup at the beginning, 1 positive and 1 negative case were randomly selected whose entire descriptions were presented to be verified. Then, using each of the four approaches, 3 positive and 3 negative cases were randomly selected and their summaries were generated; all these 24 cases were sorted in random order to be blindly verified by the subject. The character limit for GENERIC, COMPSUMM, and COMPSUMM-C was set to 140, which is around the (estimated) limit of a common snippet in Google search. After verifying each case, the subject's decision could be "coreferent", "non-coreferent", or "not sure". For DBpedia and LinkedMDB, the process was similar.

In a positive case, a "coreferent" and a "non-coreferent" decision are called *accurate* and *erroneous*, respectively. Negative cases are similarly defined. A "not



(a) DBpedia—GeoNames.

(b) DBpedia—LinkedMDB.

Fig. 1: Distribution of decisions and the average time used for verification.

sure” decision is called *undecided*. Then we evaluated the accuracy of verification based on the distribution of decisions. We also measured the efficiency of verification by the average time used for verification.

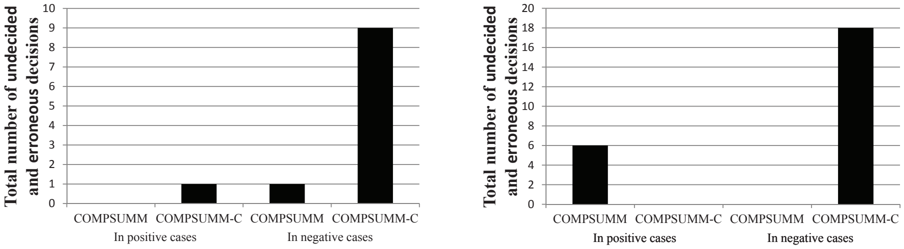
5.4 Results

All the decisions made by two subjects were excluded from the results because they unusually made two or more erroneous decisions when using NOSUMM (i.e. entire entity descriptions) so that we were not very confident about the quality of their decisions and thus excluded all of them.

Figure 1 shows the distribution of the remaining 864 decisions made by 18 subjects and the average time used for verifying a case. Between DBpedia and GeoNames, when using NOSUMM, more than 95% of the decisions were accurate, and when using other approaches that perform summarization, the accuracy rates were all above 90%. Between DBpedia and LinkedMDB, the accuracy rates were also very high when using NOSUMM and COMPSUMM. However, it decreased notably to 83% when using COMPSUMM-C and largely to 62% when using GENERIC. As to the time used, between DBpedia and GeoNames, more than 30 seconds were needed for verifying a case when using NOSUMM, but only less than 15 seconds (i.e. reduced by half or more) were needed when using other approaches that perform summarization. Between DBpedia and LinkedMDB, the results were similar. These results support our first hypothesis, that is, *length-limited entity summaries generated by COMPSUMM help human users verify more efficiently than entire entity descriptions, without notably affecting the accuracy of verification.*

The error rate of using GENERIC was 2.7–5 times higher than using COMPSUMM, depending on the data sets, and the number of undecided decisions was also much larger. These results support our second hypothesis, that is, *comparative entity summaries generated by COMPSUMM produce more accurate verification than generic entity summaries generated by GENERIC.* Besides, using COMPSUMM took even less time than using GENERIC.

Figure 2 shows the total number of undecided and erroneous decisions, divided into positive and negative cases. Between DBpedia and GeoNames, in



(a) DBpedia—GeoNames.

(b) DBpedia—LinkedMDB.

Fig. 2: Total number of undecided and erroneous decisions.

positive cases, there were very few undecided or erroneous decisions when using COMPSUMM and COMPSUMM-C, whereas in negative cases, 9 ones were made when using COMPSUMM-C, much more than using COMPSUMM. Between DBpedia and LinkedMDB, there was no undecided or erroneous decision when using COMPSUMM in negative cases, whereas using COMPSUMM-C made 18 ones. These results support our third hypothesis, that is, *comparative entity summaries generated by COMPSUMM-C which ignore the difference between entities produce less accurate verification on non-coreferent entities*. However, using COMPSUMM made 6 undecided or erroneous decisions in positive cases between DBpedia and LinkedMDB, which is unexpected and will be discussed later.

To sum up, all the three hypotheses have been confirmed. In particular, verification using comparative entity summaries generated by our approach (i.e. COMPSUMM) is 2.7–2.9 times faster than using entire entity descriptions (i.e. NOSUMM), when their accuracy rates differ insignificantly (-2.8% to 3.7%).

5.5 Discussion

A closer analysis of the undecided and erroneous decisions provided the following insights into the problem and the participant approaches.

Firstly, even using entire entity descriptions (i.e. NOSUMM), human users still occasionally made erroneous decisions. Actually, sometimes it is really difficult to make a decision. For instance, a place may have slightly different longitudes and latitudes in DBpedia and GeoNames; two different places may be very close in name and location. These greatly challenge coreference resolution.

Secondly, generic entity summaries (i.e. GENERIC) led to a large number of undecided decisions between DBpedia and LinkedMDB. A major reason is that the features selected into such a summary are often not comparable even though they are highly informative. For instance, a film may be with its writer and an actor on one side, but with its producer and another actor on the other side. The proposed comparative summaries exactly target this issue.

Thirdly, using comparative summaries that ignore the difference between entities (i.e. COMPSUMM-C) was prone to inaccurate decisions in negative cases.

It is because non-coreferent entity descriptions may share common features, which make commonality-only summaries misleading. For instance, given two different films having a common producer, a common director, but different editors, if ignoring the difference between them, their common producer and director will be selected into a comparative summary, which seems to indicate coreference. The proposed approach considers both commonality and difference, which exactly target this issue.

Last but not least, considering both commonality and difference (i.e. COMPSUMM) led to several inaccurate decisions in positive cases between DBpedia and LinkedMDB. It is because in coreferent entity descriptions, a property of high likeness to FP may occasionally have more than one value, and different values of this property may be selected on different sides due to their dissimilarity, which is misleading. For instance, a film having two editors (which is not often the case) may be misleadingly with one editor on one side, but with the other editor on the other side. This motivates us to improve our measure of difference between entities in future work.

5.6 Performance Testing

We tested the performance of our implementation on an Intel Xeon E3-1225 v2 with 512MB memory for JVM. Prior to testing, $comp_L$, ifp, fp, and inf were precomputed, and all the relevant data was loaded into memory. From DBpedia and GeoNames, 1,000 test cases were randomly selected, on which the average running time of COMPSUMM was 24 ms per case. Similarly, for DBpedia and LinkedMDB, the average running time was 35 ms per case.

6 Related Work

6.1 Coreference Resolution

More and more solutions to coreference resolution solicit user feedback for verifying candidate coreferent entities. Active learning [7] seeks to pick a set of candidate coreferent entities that, when verified, will provide the most benefit to the learner. Further, pay-as-you-go data integration [6] considers the benefit not only to the overall quality of data integration but also to the user's current task (e.g. a search). Crowdsourcing approaches [13] pay a group of users to verify candidate coreferent entities, and intend to achieve both high-quality results and a low cost. In all these approaches, the verification of candidate coreferent entities requires tool support. However, to the best of our knowledge, very little attention has been paid to it. D-Dupe [1] exactly addresses this issue with a layout highlighting the common features shared by the entities. In the field of ontology matching, tools like COGZ [4] primarily focus on the various layouts of class hierarchies to help human users verify candidate mappings between classes. *All these tools mainly concern the visualization of descriptions, whereas what we study is summarization or extraction, which complements existing tools well.*

Some low-level measures used in our approach are borrowed from automatic methods for coreference resolution, e.g. [5, 10]. Not surprisingly, both resolving entity coreference and generating comparative entity summaries involve similarity measurement. However, *they address different, though related, research problems and, in particular, our approach 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 length limit so as to not overload human users with too much information.

6.2 Entity Summarization

Entity summaries have proven to be useful as snippets in search engine results pages [2, 15], where they indicate the relevance of an entity to a keyword query. Recent studies mainly focus on the more general problem of entity summarization and generate a summary of an entity description for generic use. Among others, RELIN [3] employs a random surfer model to rank features mainly based on their informativeness but also based on the relatedness between them. DIVERSUM [11] proposes to improve the diversity of an entity summary by choosing features having different properties. Thalhammer et al. [12] prefer the features of an entity that are shared with its nearest neighbors, where the distance between entities is derived from usage data. The generic entity summaries generated by these approaches can of course be used in the verification of candidate coreferent entities. However, as demonstrated by our experimental results, *verification will be more accurate if using our comparative entity summaries that are specifically designed for this task*.

7 Conclusion

In consideration of the growing trend toward human intervention in coreference resolution through verifying candidate coreferent entities, we have addressed the improvement of the efficiency of such a verification task. Our solution extracts and presents a compact summary of entire entity descriptions in order to help human users spend less time verifying. We have defined the goodness of such a comparative entity summary from four angles. These four objectives exactly fit the binary quadratic knapsack problem, which can be efficiently solved by an effective heuristic method. We have presented an implementation of our approach, and demonstrated its effectiveness based on real-world verification tasks. The experimental results show that the comparative entity summaries generated by our approach can, as expected, help human users verify more efficiently without notably affecting the accuracy of their verification. In particular, they outperform non-comparative entity summaries generated for generic use.

To improve our approach, in the future, we will particularly explore how to automatically (or more systematically) configure the weights of different objectives. We will also extend the experiments. Specifically, we will examine how the length of a summary will influence the accuracy and efficiency of verification, and will experiment with more challenging verification tasks in different domains.

Acknowledgments. We thank all the participants and reviewers. This work was supported in part by the NSFC under Grant 61100040, 61170068, and 61223003, and in part by the JSNSF under Grant BK2012723 and BK2011189.

References

1. Bilgic, M., Licamele, L., Getoor, L., Shneiderman, B.: D-Dupe: An Interactive Tool for Entity Resolution in Social Networks. In: 2006 IEEE Symposium on Visual Analytics Science and Technology, pp. 43–50. IEEE, Washington, DC (2006)
2. Cheng, G., Qu, Y.: Searching Linked Objects with Falcons: Approach, Implementation and Evaluation. Int'l J. Semant. Web Inf. Syst. 5(3), 49–70 (2009)
3. Cheng, G., Tran, T., Qu, Y.: RELIN: Relatedness and Informativeness-based Centrality for Entity Summarization. In: Aroyo, L., Welty, C., Alani, H., Taylor, J., Bernstein, A., Kagal, L., Noy, N., Blomqvist, E. (eds.) ISWC 2011, Part I. LNCS, vol. 7031, pp. 114–129. Springer, Heidelberg (2011)
4. Falconer, S.M., Storey, M.-A.: A Cognitive Support Framework for Ontology Mapping. In: Aberer, K., Choi, K.-S., Noy, N., Allemang, D., Lee, K.-I., Nixon, L., Golbeck, J., Mika, P., Maynard, D., Mizoguchi, R., Schreiber, G., Cudré-Mauroux, P. (eds.) ISWC/ASWC 2007. LNCS, vol. 4825, pp. 114–127. Springer, Heidelberg (2007)
5. Hogan, A., Zimmermann, A., Umbrich, J., Polleres, A., Decker, S.: Scalable and Distributed Methods for Entity Matching, Consolidation and Disambiguation over Linked Data Corpora. J. Web Semant. 10, 76–110 (2012)
6. Madhavan, J., Jeffery, S.R., Cohen, S., Dong, X., Ko, D., Yu, C., Halevy, A.: Web-scale Data Integration: You Can Only Afford to Pay As You Go. In: 3rd Biennial Conference on Innovative Data Systems Research, pp. 342–350. cidrdb.org (2007)
7. Ngonga Ngomo, A.-C., Lehmann, J., Auer, S., Höffner, K.: RAVEN - Active Learning of Link Specifications. In: 6th International Workshop on Ontology Matching, pp. 25–36. CEUR-WS.org (2011)
8. Pisinger, D.: The Quadratic Knapsack Problem - A Survey. Discrete Appl. Math. 155(15), 623–648 (2007)
9. Shvaiko, P., Euzenat, J.: Ontology Matching: State of the Art and Future Challenges. IEEE Trans. Knowl. Data Eng. 25(1), 158–176 (2013)
10. Stoilos, G., Stamou, G., Kollias, S.: A String Metric for Ontology Alignment. In: Gil, Y., Motta, E., Benjamins, V.R., Musen, M.A. (eds.) ISWC 2005. LNCS, vol. 3729, pp. 624–637. Springer, Heidelberg (2005)
11. Sydow, M., Pikuła, M., Schenkel, R.: The Notion of Diversity in Graphical Entity Summarisation on Semantic Knowledge Graphs. J. Intell. Inf. Syst. 41(2), 109–149 (2013)
12. Thalhammer, A., Toma, I., Roa-Valverde, A.J., Fensel, D.: Leveraging Usage Data for Linked Data Movie Entity Summarization. In: 2nd International Workshop on Usage Analysis and the Web of Data. (2012)
13. Wang, J., Kraska, T., Franklin, M.J., Feng, J.: CrowdER: Crowdsourcing Entity Resolution. Proc. VLDB Endowment 5(11), 1483–1494 (2012)
14. Yang, Z., Wang, G., Chu, F.: An Effective GRASP and Tabu Search for the 0-1 Quadratic Knapsack Problem. Comput. Oper. Res. 40(5), 1176–1185 (2013)
15. Zhang, L., Zhang, Y., Chen, Y.: Summarizing Highly Structured Documents for Effective Search Interaction. In: 35th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 145–154. ACM, New York (2012)