

Summarizing Entity Descriptions for Effective and Efficient Human-centered Entity Linking

Gong Cheng
gcheng@nju.edu.cn

Danyun Xu
dyxu@smail.nju.edu.cn

Yuzhong Qu
yzqu@nju.edu.cn

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

ABSTRACT

Entity linking connects the Web of documents with knowledge bases. It is the task of linking an entity mention in text to its corresponding entity in a knowledge base. Whereas a large body of work has been devoted to automatically generating candidate entities, or ranking and choosing from them, manual efforts are still needed, e.g., for defining gold-standard links for evaluating automatic approaches, and for improving the quality of links in crowdsourcing approaches. However, structured descriptions of entities in knowledge bases are sometimes very long. To avoid overloading human users with too much information and help them more efficiently choose an entity from candidates, we aim to substitute entire entity descriptions with compact, equally effective structured summaries that are automatically generated. To achieve it, our approach analyzes entity descriptions in the knowledge base and the context of entity mention from multiple perspectives, including characterizing and differentiating power, information overlap, and relevance to context. Extrinsic evaluation (where human users carry out entity linking tasks) and intrinsic evaluation (where human users rate summaries) demonstrate that summaries generated by our approach help human users carry out entity linking tasks more efficiently (22–23% faster), without significantly affecting the quality of links obtained; and our approach outperforms existing approaches to summarizing entity descriptions.

Categories and Subject Descriptors

H.3.1 [Information Storage and Retrieval]: Content Analysis and Indexing—*abstracting methods*; H.1.2 [Models and Principles]: User/Machine Systems—*human factors, human information processing*; H.5.2 [Information Interfaces and Presentation]: User Interfaces—*user-centered design*

Copyright is held by the International World Wide Web Conference Committee (IW3C2). IW3C2 reserves the right to provide a hyperlink to the author's site if the Material is used in electronic media.
WWW 2015, May 18–22, 2015, Florence, Italy.
ACM 978-1-4503-3469-3/15/05.
<http://dx.doi.org/10.1145/2736277.2741094>.

General Terms

Algorithms, Experimentation, Human Factors

Keywords

Class vector model, entity summarization, human-centered entity linking, quadratic multidimensional knapsack

1. INTRODUCTION

The Web is embracing an increasing number of entity-centric knowledge bases, like Google's Knowledge Graph, Facebook's Open Graph, and W3C's Linked RDF Data. To bridge such structured data and traditional unstructured data (i.e. text in webpages) and form a unified Web, a fundamental task is to link entity mentions in text to their corresponding entities in a knowledge base, called entity linking, as illustrated in Fig. 1. Whereas a large body of work has been devoted to automatically generating candidate entities, or ranking and choosing from them [16], manual efforts are still needed for at least two reasons. Firstly, to evaluate automatic approaches to entity linking, human experts need to define gold-standard links by manually carrying out entity linking tasks [4, 10]. Secondly, some crowdsourcing approaches to entity linking directly solicit links from human users via entity linking tasks [5]. However, to the best of our knowledge, *no attention has been focused on facilitating human-centered entity linking*, and thus our aim is to *help human users effectively and more efficiently carry out entity linking tasks*. Thereafter, human users' productivity can be improved, and/or the payment can be reduced.

As illustrated in Fig. 1, when carrying out an entity linking task, the human user is usually provided with a set of candidate entities in the knowledge base, and is responsible for choosing one from them to be linked to, based on the context of the entity mention and the descriptions of candidate entities. The structured description of an entity comprises a set of property-value pairs, called features, and it is often the case that an entity in a modern knowledge base (e.g. Freebase and DBpedia [12]) has hundreds of or even more features. To avoid overloading human users with too much information and help them *more efficiently* choose an entity from candidates, in this paper we attack the problem of *how to summarize descriptions of candidate entities* by selecting and presenting only a subset of features. The selected features are expected to *effectively* substitute entire entity descriptions, without significantly affecting the quality of links produced by human users.

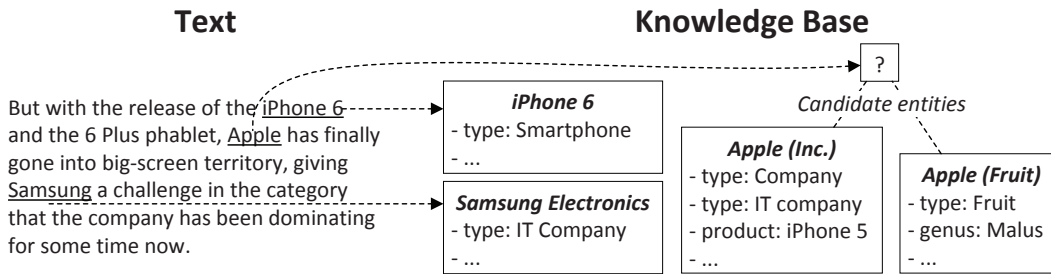


Figure 1: In this example, “iPhone 6” and “Samsung” in text have been linked to corresponding entities in the knowledge base. For “Apple”, two candidate entities have been found to be chosen by human users.

To achieve it, we analyze entity descriptions in the knowledge base and the context of entity mention from three perspectives, and propose to automatically generate *characteristic*, *differential*, and *contextual* summaries by selecting features based on their characterizing and differentiating power, the information overlap between them, and their relevance to the context of entity mention. Accordingly, we develop three approaches from different perspectives, and finally combine them. To evaluate the proposed approaches, we carry out extrinsic evaluation by inviting human users to carry out entity linking tasks based on descriptions of candidate entities or their summaries generated by different approaches, and also carry out intrinsic evaluation by inviting human users to directly rate and comment on summaries generated by different approaches.

The remainder of this paper is structured as follows. Section 2 gives some preliminaries and defines the problem. Section 3, 4, and 5 describe three approaches from different perspectives. Section 6 combines the three approaches into a more powerful one. Section 7 presents experiments. Section 8 discusses related work. Section 9 concludes the paper with future work.

2. PROBLEM STATEMENT

A knowledge base is characterized by an eight-tuple $KB = \langle E, C, P, D, \sqsubseteq_C, \sqsubseteq_P, dom, d \rangle$, where

- E is a set of entities,
- C is a set of classes (a.k.a. entity types),
- P is a set of properties,
- D is a set of data values,
- \sqsubseteq_C is a reflexive and transitive binary relation on C characterizing the subclass-superclass relation,
- \sqsubseteq_P is a reflexive and transitive binary relation on P characterizing the subproperty-superproperty relation,
- $dom : P \rightarrow C$ is a function returning the domain of a property, which is a class, and
- $d : E \rightarrow 2^{P \times (D \cup C \cup E)}$ is a function returning the structured description of an entity comprising a set of property-value pairs (called features) where a value is a data value (e.g. string, integer), a class (if the property is type), or an entity.

For convenience, given a feature f , let $p(f)$ and $v(f)$ return the property and the value of this feature, respectively.

Each entity, class, and property is assumed to have a human-readable name. When presenting a feature to human users, for each entity, class, or property involved, its name will be shown; for a data value, its string form will be shown. The length of a feature f , 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 $v(f)$.

Figure 1 illustrates four entity descriptions in a knowledge base. In practical human-centered entity linking tasks, features in descriptions of candidate entities are sometimes too long to be completely presented without overloading human users with too much information. To help human users more efficiently choose an entity from candidates, a compact, length-limited summary of entity descriptions is expected. Given k candidate entities $e_1, e_2, \dots, e_k \in E$ to be chosen from and linked to from an entity mention, a structured *summary* of their descriptions consists of a subset of features selected from each entity description, or formally, $\langle S_1, S_2, \dots, S_k \rangle$ subject to $S_i \subseteq d(e_i)$ for $i = 1, 2, \dots, k$. Given a length limit L , a *feasible summary* $\langle S_1, S_2, \dots, S_k \rangle$ satisfies

$$\sum_{f \in S_i} l(f) \leq L, \text{ for } i = 1, 2, \dots, k. \quad (1)$$

There could be many different feasible summaries. In the next sections, we will discuss, from three perspectives, what kinds of features will be selected to form a feasible summary that is effective in facilitating human-centered entity linking.

3. GENERATING CHARACTERISTIC SUMMARIES

Our first approach facilitates human-centered entity linking by selecting features and generating summaries that help human users determine the identity of each candidate entity. In the following, we introduce the basic idea of this approach, and then formulate it, at a high level, as a combinatorial optimization problem and solve it. Finally we discuss a low-level implementation of the measures used in the high-level problem formulation.

3.1 Basic Idea

To help human users choose an entity from k candidates to be linked to, we assume that the user has determined the identity of the entity mention based on its context. Then, entity linking will be successfully carried out if the user is able to correctly determine the identity of each candidate

entity. To this end, we propose to generate a summary that can best reflect the identity of each candidate entity, i.e., to select features that can best characterize each candidate entity or, in other words, have the highest *characterizing power*. For instance, in Fig. 1, the feature “product: iPhone 5” has a very high characterizing power because it is a unique feature of the entity “Apple (Inc.)”. We will measure the characterizing power of a feature by using information theory (cf. Sect. 3.3).

Further, we observe that features in an entity description may carry overlapping information, e.g. “type: Company” and “type: IT company” of the entity “Apple (Inc.)” in Fig. 1. To provide more information to the user within a limited length, we improve the diversity of the generated summary by selecting features from each entity description that not only are characteristic but also share not much *information overlap* between each other. We will measure the information overlap between two features by using logical inference and string/numerical similarity (cf. Sect. 3.3).

Next we will formulate the selection of features as a combinatorial optimization problem that maximizes the characterizing power of and minimizes the information overlap between selected features.

3.2 High-level Problem Formulation

Given k candidate entity descriptions to be summarized, our goal is to maximize the total characterizing power of selected features and also minimize the total information overlap between features selected from the same entity description. Generally, the two objectives could be conflicting; that is, sometimes no single feasible summary can simultaneously optimize both objectives. Optimal summaries need to be formed in the presence of a trade-off between the two objectives. We solve such a multi-objective optimization problem by quantifying the trade-off in satisfying the two objectives by formulating a linear scalarization to be maximized. Formally, let $ch(f) \in [0, 1]$ be the characterizing power of feature f , and let $ov(f_m, f_n) \in [0, 1]$ be the information overlap between feature f_m and f_n . We define the goodness of a feasible summary $\langle S_1, S_2, \dots, S_k \rangle$ as

$$g_{\text{chr}}(\langle S_1, S_2, \dots, S_k \rangle) = \alpha \cdot \sum_{i=1}^k \sum_{f \in S_i} ch(f) + \beta \cdot \sum_{i=1}^k \sum_{\substack{f_m, f_n \in S_i \\ f_m \neq f_n}} (-ov(f_m, f_n)), \quad (2)$$

where $\alpha, \beta > 0$ are the weights of the objectives to be tuned in the specific application, and an implementation of ch and ov will be given in the next subsection. Now our goal is to find a feasible summary that maximizes g_{chr} .

We observe that g_{chr} can be reformulated as k independent sub-objectives to be simultaneously optimized:

$$g_{\text{chr}}(\langle S_1, S_2, \dots, S_k \rangle) = \sum_{i=1}^k g'_{\text{chr}}(S_i), \quad (3)$$

where

$$g'_{\text{chr}}(S_i) = \alpha \cdot \sum_{f \in S_i} ch(f) + \beta \cdot \sum_{\substack{f_m, f_n \in S_i \\ f_m \neq f_n}} (-ov(f_m, f_n)), \quad (4)$$

that is, features can be selected from each candidate entity description separately.

The optimization of g'_{chr} can be formulated as an instance of the binary quadratic knapsack problem (QKP) [14]. Specifically, given a candidate entity e_i , we number the features in its description $d(e_i)$ from f_1 to $f_{|d(e_i)|}$. By introducing a series of binary variables x_m for $m = 1, 2, \dots, |d(e_i)|$ to indicate whether feature f_m is selected into the optimal summary, the optimization of g'_{chr} is reformulated as:

$$\begin{aligned} & \text{maximize} && \sum_{m=1}^{|d(e_i)|} \sum_{n=m}^{|d(e_i)|} p_{m,n} \cdot x_m \cdot x_n \\ & \text{subject to} && \sum_{m=1}^{|d(e_i)|} l(f_m) \cdot x_m \leq L, \\ & && x_m \in \{0, 1\}, m = 1, 2, \dots, |d(e_i)|, \end{aligned} \quad (5)$$

where $l(f_m)$ and L (cf. Eq. (1)) are regarded as the “weight” of feature f_m and the “capacity” of the knapsack, respectively, and $p_{m,n}$ is the “profit” achieved if both feature f_m and f_n are selected, which is defined as:

$$p_{m,n} = \begin{cases} \alpha \cdot ch(f_m) & \text{if } m = n, \\ \beta \cdot (-ov(f_m, f_n)) & \text{if } m \neq n. \end{cases} \quad (6)$$

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 P=NP [14]. So we solve it by using a state-of-the-art heuristic method [20]. Basically, this GRASP-based method consists of iterations made up of successive constructions of a greedy randomized solution and subsequent iterative improvements of it through local search. In particular, a greedy randomized solution is generated by adding elements (i.e. features) to the solution (i.e. the summary) from a list of features ranked by the following greedy function:

$$\frac{g'_{\text{chr}}(S'_i \cup \{f_x\})}{\sum_{f \in S'_i \cup \{f_x\}} l(f)}, \quad (7)$$

where S'_i is the set of features having been added to the summary in previous iterations and f_x is an unselected candidate feature being evaluated. It represents the profit per unit of weight that will be achieved if f_x is added to the summary.

3.3 Low-level Implementation

Now we give an implementation of ch and ov .

To measure $ch(f) \in [0, 1]$, the characterizing power of a feature f , we follow information theory and compute the normalized amount of self-information contained in the probabilistic event of observing f in an entity description. We estimate probabilities based on the knowledge base $KB = \langle E, C, P, D, \sqsubseteq_C, \sqsubseteq_P, dom, d \rangle$:

$$\begin{aligned} ch(f) &= \frac{-\log \frac{|\{e \in E : f \in d(e)\}|}{|E|}}{\log |E|} \\ &= 1 - \frac{\log |\{e \in E : f \in d(e)\}|}{\log |E|}, \end{aligned} \quad (8)$$

that is, a feature will have high characterizing power if it is shared by a small number of entities in KB .

To measure $ov(f_m, f_n) \in [0, 1]$, the information overlap between feature f_m and f_n , we perform logical inference

based on the ontological semantics of classes and properties in KB , and also employ a range of similarity measures.

Firstly, we will define $ov(f_m, f_n) = 1$, i.e. maximize the information overlap between f_m and f_n , if one of them can be inferred from the other based on the schema level of KB , namely C , P , \sqsubseteq_C , \sqsubseteq_P , and dom . The following rules of inference are used.

- If $p(f_m) = p(f_n) = \mathbf{type}$, $v(f_m), v(f_n) \in C$, and $v(f_m) \sqsubseteq_C v(f_n)$, then f_n can be inferred from f_m .
- If $p(f_m) \sqsubseteq_P p(f_n)$ and $v(f_m) = v(f_n)$, then f_n can be inferred from f_m .
- If $p(f_n) = \mathbf{type}$, $v(f_n) \in C$, and $dom(p(f_m)) \sqsubseteq_C v(f_n)$, then f_n can be inferred from f_m .

Secondly, if none of the above rules can be applied, we will measure the similarity between properties ($psim$) and the similarity between property values ($vsim$). For $psim$, we measure the ISub string similarity [17] between the names of $p(f_m)$ and $p(f_n)$, which is in the range $[-1, 1]$. For $vsim$, we also measure the string similarity between $v(f_m)$ and $v(f_n)$, but specially handle numerical values. Specifically, if both $v(f_m)$ and $v(f_n)$ are numerical data values, we define $vsim(f_m, f_n) \in [-1, 1]$ as follows.

1. If $v(f_m) = v(f_n)$, then $vsim(f_m, f_n) = 1$;
2. otherwise, if $v(f_m) \cdot v(f_n) < 0$, then $vsim(f_m, f_n) = -1$;
3. otherwise, $vsim(f_m, f_n) = \frac{\min\{|v(f_m)|, |v(f_n)|\}}{\max\{|v(f_m)|, |v(f_n)|\}}$.

If at least one of $v(f_m)$ and $v(f_n)$ is not a numerical data value, we will use the ISub string similarity between their names or string forms to define $vsim(f_m, f_n)$, which is in the range $[-1, 1]$. Finally, we define

$$ov(f_m, f_n) = \max\{psim(f_m, f_n), vsim(f_m, f_n), 0\}. \quad (9)$$

4. GENERATING DIFFERENTIAL SUMMARIES

Our second approach facilitates human-centered entity linking by selecting features and generating summaries that help human users identify the difference between candidate entities. In the following, we introduce the basic idea of this approach, and then formulate it, at a high level, as a combinatorial optimization problem and solve it. Finally we discuss a low-level implementation of the measures used in the high-level problem formulation.

4.1 Basic Idea

Similar to our first approach described in Sect. 3, here we also assume that the user has determined the identity of the entity mention based on its context. Then, to help the user correctly determine the identity of each candidate entity and choose one from them, we propose to generate a summary that can best differentiate candidate entities from each other, i.e., to select features that can best reflect the difference between candidate entities or, in other words, have the highest *differentiating power*. For instance, in Fig. 1, the feature “type: Company” of the entity “Apple (Inc.)” and the feature “type: Fruit” of the entity “Apple (Fruit)” have a very high differentiating power because they provide

very different values of the same property. We will measure the differentiating power of a pair of features from different entity descriptions in a way similar to the measurement of information overlap between features given in Sect. 3.3, by using logical inference, string/numerical similarity, and even more (cf. Sect. 4.3).

Next we will formulate the selection of features as a combinatorial optimization problem that maximizes the differentiating power of selected features.

4.2 High-level Problem Formulation

Given k candidate entity descriptions to be summarized, our goal is to maximize the total differentiating power of features selected from different entity descriptions. Formally, let $di(f_m, f_n) \in [0, 1]$ be the differentiating power of a pair of features f_m and f_n . We define the goodness of a feasible summary $\langle S_1, S_2, \dots, S_k \rangle$ as

$$g_{\text{diff}}(\langle S_1, S_2, \dots, S_k \rangle) = \sum_{i=1}^k \sum_{j=i+1}^k \sum_{\substack{f_m \in S_i \\ f_n \in S_j}} di(f_m, f_n). \quad (10)$$

An implementation of di will be given in the next subsection. Now our goal is to find a feasible summary that maximizes g_{diff} .

The optimization of g_{diff} can be formulated as an instance of the binary quadratic multidimensional knapsack problem (QMKP), namely QKP with multiple constraints. Specifically, given k candidate entities e_1, e_2, \dots, e_k , we number the features in $d(e_i)$, for $i = 1, 2, \dots, k$, from $f_{i,1}$ to $f_{i,|d(e_i)|}$. By introducing a series of binary variables $x_{i,m}$ for $i = 1, 2, \dots, k$ and $m = 1, 2, \dots, |d(e_i)|$ to indicate whether feature $f_{i,m}$ is selected into the optimal summary, the optimization of g_{diff} is reformulated as:

$$\begin{aligned} & \text{maximize} \quad \sum_{i=1}^k \sum_{j=i+1}^k \sum_{m=1}^{|d(e_i)|} \sum_{n=1}^{|d(e_j)|} p_{i,m,j,n} \cdot x_{i,m} \cdot x_{j,n} \\ & \text{subject to} \quad \sum_{m=1}^{|d(e_i)|} l(f_{i,m}) \cdot x_{i,m} \leq L, \quad i = 1, 2, \dots, k, \\ & \quad x_{i,m} \in \{0, 1\}, \quad i = 1, 2, \dots, k, \quad m = 1, 2, \dots, |d(e_i)|, \end{aligned} \quad (11)$$

where $l(f_{i,m})$ and L (cf. Eq. (1)) are regarded as the “weight” of feature $f_{i,m}$ and the “capacity” of each of the k knapsacks, respectively, and $p_{i,m,j,n}$ is the “profit” achieved if both feature $f_{i,m}$ and $f_{j,n}$ are selected, which is defined as:

$$p_{i,m,j,n} = di(f_{i,m}, f_{j,n}). \quad (12)$$

QMKP is at least as hard as QKP. To solve it, we adapt the heuristic method for solving QKP proposed in [20]. One basic modification to the method is to consider multiple constraints throughout the method. Besides, we develop a new greedy function to rank features:

$$\frac{\gamma \cdot g_{\text{diff}}(\langle S'_1, S'_2, \dots, S'_x \cup \{f_{x,y}\}, \dots, S'_k \rangle) + \delta \cdot \text{Potential}}{\sum_{i=1}^k \sum_{f \in S'_i} l(f) + l(f_{x,y})}, \quad (13)$$

where $\langle S'_1, S'_2, \dots, S'_k \rangle$ represents the sets of features having been added to the summary in previous iterations, $f_{x,y}$ is an unselected candidate feature being evaluated, $\gamma, \delta \in [0, 1]$ are

weights to be tuned in the specific application, and

$$\text{Potential} = \sum_{\substack{i=1 \\ i \neq x}}^k \sum_{f \in d(e_i) \setminus S'_i} di(f, f_{x,y}). \quad (14)$$

If $\gamma = 1$ and $\delta = 0$, i.e. ignoring Potential, the greedy function given by Eq. (13) will be similar to the one given by Eq. (7), representing the profit per unit of weight that will be achieved if $f_{x,y}$ is added to the summary. In general cases, Potential gives the maximum profit that can be achieved by pairing $f_{x,y}$ with other unselected candidate features, representing the potential profit to achieve with $f_{x,y}$. We introduce Potential because we observe that, different from the definition of profit in Eq. (6) where features can contribute independently, here in the definition of profit in Eq. (12), features will contribute only if they are paired. If Potential is not considered, an arbitrary feature will be selected at the start of iterations, and will direct the selection of features in subsequent iterations. So Potential is introduced to direct the selection toward promising features.

4.3 Low-level Implementation

Now we give an implementation of di .

To measure $di(f_{i,m}, f_{j,n}) \in [0, 1]$, the differentiating power of a pair of features $f_{i,m}$ and $f_{j,n}$, we perform logical inference based on the ontological semantics of classes and properties in KB , and also employ a range of dissimilarity measures.

Firstly, we will define $di(f_{i,m}, f_{j,n}) = 0$, i.e. minimize the differentiating power of $f_{i,m}$ and $f_{j,n}$, if one of them can be inferred from the other based on the schema level of KB , namely C , P , \sqsubseteq_C , \sqsubseteq_P , and dom . We reuse the rules of inference presented in Sect. 3.3.

Secondly, if none of the rules of inference can be applied, we will measure the dissimilarity between property values by reusing $vsim$, the similarity between property values given in Sect. 3.3. Given a pair of features $f_{i,m}$ and $f_{j,n}$, we will define $di(f_{i,m}, f_{j,n}) = 0$ unless all the following conditions are satisfied.

- $p(f_{i,m}) = p(f_{j,n})$, i.e., they have the same property.
- $f_{i,m} \notin d(e_j)$ and $f_{j,n} \notin d(e_i)$, i.e., they are not shared by the two entities but are unique to the entity descriptions they belong to.
- $vsim(f_{i,m}, f_{j,n}) < 0$, i.e., the two property values are dissimilar.

If all the above conditions are satisfied, we will define

$$di(f_{i,m}, f_{j,n}) = vu(p(f_{i,m})) \cdot |vsim(f_{i,m}, f_{j,n})|, \quad (15)$$

where $vu(p(f_{i,m})) \in [0, 1]$ is to weight the result by $p(f_{i,m})$'s value uniqueness, namely how likely $p(f_{i,m})$ takes at most one value for an entity. We estimate it by the reciprocal of the average number of values taken by $p(f_{i,m})$ in an entity description in $KB = \langle E, C, P, D, \sqsubseteq_C, \sqsubseteq_P, dom, d \rangle$:

$$vu(p(f_{i,m})) = \frac{|\{e \in E : \exists f \in d(e), (p(f) = p(f_{i,m}))\}|}{\sum_{e \in E} |\{f \in d(e) : p(f) = p(f_{i,m})\}|}. \quad (16)$$

That is, the fewer values a property usually takes for an entity, the bigger difference an observation of its different values indicates.

5. GENERATING CONTEXTUAL SUMMARIES

Our third approach facilitates human-centered entity linking by selecting features and generating summaries that help human users determine the relevance of each candidate entity to the entity mention and its context. In the following, we introduce the basic idea of this approach, and then propose a class vector model for representing features of candidate entities and the context of entity mention as vectors of classes. Finally we present a way of computing weights in vectors and an algorithm for selecting features.

5.1 Basic Idea

Different from the two approaches described in Sect. 3 and 4, here we do not assume that the user has determined the identity of the entity mention in text. Rather, to help the user choose an entity from candidates, we propose to generate a summary that can best reflect the *relevance* of each candidate entity to the entity mention and its context. For instance, in Fig. 1, the feature “type: IT Company” of the entity “Apple (Inc.)” is highly relevant to the context of entity mention because another entity mention “Samsung” in the context has been linked to the entity “Samsung Electronics” which is also an “IT Company”. We will measure relevance based on a model representing both features of candidate entities and the context of entity mention as vectors of classes (cf. Sect. 5.2) that are appropriately weighted (cf. Sect. 5.3).

Further, instead of simply selecting top-ranked relevant features, we will improve the diversity of the generated summary by diversity-based re-ranking, which is also based on the class vector model (cf. Sect. 5.4).

5.2 Class Vector Model

Term vector model (a.k.a. vector space model or VSM) has been extensively used for relevance ranking in information retrieval [15]. It represents both documents and queries as vectors of terms, and then defines the relevance of each document to a query as the cosine of the angle between the two corresponding vectors (called their cosine similarity). Inspired by this, we propose to represent both features of candidate entities and the context of entity mention as vectors. In our *class vector model* (CVM), each dimension corresponds to not a separate term but a class in the knowledge base. If a class, its subclass, or its instance occurs in a feature (or the context of entity mention), its weight in the vector is non-zero. In CVM, the similarity between two vectors is also defined as their cosine similarity. The relevance of a feature to the context of entity mention is then defined as the cosine similarity between the corresponding vectors.

5.3 CF-IIF Weighting

We weight classes in vectors in CVM by class frequency-inverse instance frequency (CF-IIF), which is adapted from the term frequency-inverse document frequency (TF-IDF) which is often used with VSM.

Let $I(c) \subseteq E$ be the set of class c 's instances in $KB = \langle E, C, P, D, \sqsubseteq_C, \sqsubseteq_P, dom, d \rangle$:

$$I(c) = \{e \in E : \exists f \in d(e), (p(f) = \mathbf{type}, v(f) \sqsubseteq_C c)\}. \quad (17)$$

The number of times a class c , its subclass, or its instance occurs in a feature f of a candidate entity is called c 's class

frequency (cf) in f . Since a property value could be a class, an entity, or a data value, we formalize cf in different cases:

$$cf(c, f) = \begin{cases} 1 & \text{if } v(f) \in C \text{ and } v(f) \sqsubseteq_C c, \\ 1 & \text{if } v(f) \in E \text{ and } v(f) \in I(c), \\ 0 & \text{otherwise.} \end{cases} \quad (18)$$

To calculate c 's class frequency in the context of entity mention, denoted by $cf_{cnt}(c)$, we assume that a set of entities E_{cnt} have been linked to from other entity mentions in the context, and count the number of c 's instances within this set:

$$cf_{cnt}(c) = |I(c) \cap E_{cnt}|. \quad (19)$$

The inverse instance frequency (idf) of a class c is a measure of how much information c provides, i.e., whether c is common or rare across all entities. It is obtained by dividing the total number of entities in KB by the number of c 's instances, and then taking the logarithm of this quotient:

$$idf(c) = \log \frac{|E|}{|I(c)|}. \quad (20)$$

Finally, for a class c in a vector representing a feature f of a candidate entity, its CF-IIF weight is

$$cf(c, f) \cdot idf(c), \quad (21)$$

and for a class c in a vector representing the context of entity mention, its CF-IIF weight is

$$cf_{cnt}(c) \cdot idf(c). \quad (22)$$

5.4 Summary Generation

Instead of simply selecting top-ranked relevant features, we diversify generated summaries by maximizing marginal relevance (MMR) in the selection of features. MMR [1] is a widely used framework for integrating relevance and diversity. According to MMR, features are iteratively selected. In each iteration, unselected candidate features are re-ranked according to a linear combination of their relevance scores and their similarity to previously selected features, and the one having the highest re-ranking score will be added to the summary. Iterations will stop if adding any of the unselected candidate features exceeds the length limit (i.e. L). Specifically, let V_{cnt} and $V(f)$ be the vector in CVM representing the context of entity mention and the vector representing a feature f , respectively. Let S' be the set of features having been added to the summary in previous iterations, and let f_x be an unselected candidate feature being evaluated. The re-ranking score of f_x is defined as its marginal relevance:

$$\epsilon \cdot cs(V(f_x), V_{cnt}) - (1 - \epsilon) \cdot \max_{f \in S'} cs(V(f_x), V(f)), \quad (23)$$

where cs returns the cosine similarity between two vectors which is in the range $[0, 1]$, and $\epsilon \in [0, 1]$ is a weight to be tuned in the specific application.

6. COMBINATION OF APPROACHES

The three proposed approaches select features and generate summaries from different, complementary perspectives. We combine them by integrating the two approaches presented in Sect. 3 and 5 using harmonic mean, and then incorporating the result into an extension of the formulation of QMKP presented in Sect. 4.

Specifically, given k candidate entities e_1, e_2, \dots, e_k , we number the features in $d(e_i)$, for $i = 1, 2, \dots, k$, from $f_{i,1}$ to $f_{i,|d(e_i)|}$. By introducing a series of binary variables $x_{i,m}$ for $i = 1, 2, \dots, k$ and $m = 1, 2, \dots, |d(e_i)|$ to indicate whether feature $f_{i,m}$ is selected into the optimal summary, we formulate an instance of QMKP as:

$$\begin{aligned} & \text{maximize} \quad \sum_{i=1}^k \sum_{j=i}^k \sum_{m=1}^{|d(e_i)|} \sum_{n=1}^{|d(e_j)|} p_{i,m,j,n} \cdot x_{i,m} \cdot x_{j,n} \\ & \text{subject to} \quad \sum_{m=1}^{|d(e_i)|} l(f_{i,m}) \cdot x_{i,m} \leq L, \quad i = 1, 2, \dots, k, \\ & \quad \quad \quad x_{i,m} \in \{0, 1\}, \quad i = 1, 2, \dots, k, \quad m = 1, 2, \dots, |d(e_i)|, \end{aligned} \quad (24)$$

where $l(f_{i,m})$ and L (cf. Eq. (1)) are regarded as the ‘‘weight’’ of feature $f_{i,m}$ and the ‘‘capacity’’ of each of the k knapsacks, respectively, and $p_{i,m,j,n}$ is the ‘‘profit’’ achieved if both feature $f_{i,m}$ and $f_{j,n}$ are selected, which is defined as:

$$p_{i,m,j,n} = \begin{cases} \alpha' \cdot hm(ch(f_{i,m}), cs(V(f_{i,m}), V_{cnt})) & \text{if } i = j \\ & \text{and } m = n, \\ -\beta' \cdot (1 - hm(1 - ov(f_{i,m}, f_{j,n}), \\ \quad 1 - cs(V(f_{i,m}), V(f_{j,n})))) & \text{if } i = j \\ & \text{and } m \neq n, \\ \zeta' \cdot di(f_{i,m}, f_{j,n}) & \text{if } i \neq j, \end{cases} \quad (25)$$

where hm returns the harmonic mean of two values, ch and ov are defined in Sect. 3, di is defined in Sect. 4, cs , V , and V_{cnt} are defined in Sect. 5, and $\alpha', \beta', \zeta' > 0$ are the weights to be tuned in the specific application.

To solve it, we accordingly extend the greedy function in Eq. (13), considering both the profit that will be immediately achieved if adding a feature to the summary and the potential profit that can be achieved by this feature in subsequent iterations. The extension is straightforward, and its details are omitted.

7. EXPERIMENTS

To evaluate the proposed approaches, we carried out both extrinsic and intrinsic evaluation. In extrinsic evaluation, we invited human users to carry out entity linking tasks based on structured descriptions of candidate entities or their summaries generated by different approaches, and then analyzed the accuracy of linking as well as the time used. In intrinsic evaluation, we invited human users to directly rate and comment on summaries generated by different approaches, and then analyzed the ratings and comments received. We also tested the running time of our approach.

7.1 Data Sets

Documents from two text corpora widely adopted for evaluating entity linking were used in our experiments to provide entity mentions, which had been linked to Wikipedia articles and were then mapped to entities in the DBpedia knowledge base [12] as the gold standard for linking.

- **AQUAINT.** The AQUAINT Corpus of English News Text [9] consists of newswire text data in English drawn from the Xinhua News Service, the New York Times News Service, and the Associated Press Worldstream News Service. The authors of [13] randomly selected

50 short documents from this corpus, and then manually linked entity mentions in these documents to corresponding Wikipedia articles.¹ These documents, entity mentions, and links were used in our experiments.

- **IITB.** The authors of [11] built IITB by collecting 104 documents linked to from homepages of popular sites belonging to a handful of domains including sports, entertainment, science and technology, and health, and then manually linked entity mentions in these documents to corresponding Wikipedia articles.² All of these documents, entity mentions, and links were used in our experiments.

We used DBpedia (version 3.9-en³) as the knowledge base to be linked to, which provided four million entity descriptions of various types extracted from Wikipedia. To form gold-standard links, Wikipedia articles linked to from entity mentions in the above text corpora were mapped to corresponding entities in DBpedia according to DBpedia’s “Links to Wikipedia Article” data set. Structured descriptions of these entities were obtained by combining eight of DBpedia’s major data sets, including Mapping-based Types, Mapping-based Properties, Titles, Geographic Coordinates, Homepages, Persondata, PND, and YAGO types. Features containing non-English data values were removed from entity descriptions because subjects in the experiments might not read them. The schema level of the knowledge base was obtained by combining a set of ontologies/schemas used in these entity descriptions, including the DBpedia ontology,⁴ YAGO type hierarchy,⁵ schema.org terms,⁶ and FOAF vocabulary.⁷

7.2 Tasks

An entity linking task to be carried out by subjects consisted of a document randomly selected from a text corpus (i.e. AQUAINT or IITB), an entity mention randomly selected from the document, and three candidate entities in the knowledge base (i.e. DBpedia) to be chosen from and linked to. The three candidate entities included one entity (called the target entity) that was expected to be chosen from and linked to according to the gold standard, and the other two entities (called noise entities) that were randomly selected from a set of entities in DBpedia that shared a common name with the target entity. Such noise entities were obtained from DBpedia’s “Disambiguation links” data set, which was extracted from Wikipedia’s disambiguation pages.

We required the three candidate entities in a task to have at least twenty features in their descriptions, so that it was possible for different approaches to generate significantly different summaries. Only 155 entity mentions from 45 documents in AQUAINT and 487 entity mentions from 98 documents in IITB satisfied this requirement, and were to be used in the experiments.

¹<http://www.nzdl.org/wikification/docs.html>

²<http://www.cse.iitb.ac.in/soumen/doc/CSAW/Annot/>

³<http://wiki.dbpedia.org/Downloads39>

⁴<http://wiki.dbpedia.org/Downloads39?dbpedia-ontology>

⁵<http://wiki.dbpedia.org/Downloads39?yago-type-hierarchy>

⁶<http://schema.rdfs.org/all.rdf>

⁷<http://xmlns.com/foaf/spec/>

7.3 Approaches

Subjects chose entities to be linked to based on their structured descriptions or summaries generated by six approaches to be compared.

- **DESC** returns the entire description of each entity, without performing summarization.
- **CHR** generates characteristic summaries according to Sect. 3.
- **DFF** generates differential summaries according to Sect. 4.
- **CNT** generates contextual summaries according to Sect. 5.
- **COMB** combines the above three approaches according to Sect. 6.
- **RELIN** [3] is a state-of-the-art approach to summarizing entity descriptions for generic purposes. Similar to CHR, it prefers to select characteristic features from each entity description. It also considers relatedness between features, whereas CHR measures information overlap between features.

In CHR, DFF, CNT, COMB, and RELIN, the length limit L in Eq. (1) was set to 100 characters, whereas the average total length of all candidate entity descriptions in all possible tasks was 680.36 characters; that is, the expected space savings produced by summarization was 85.30%. Other parameters in RELIN were set according to [3]. Other parameters in CHR, DFF, CNT, and COMB were empirically tuned based on three tasks randomly selected from each of the two text corpora (which were kept separate from those to be used in subsequent experiments): in CHR, $\alpha = 2$ and $\beta = 5$ in Eq. (6); in DFF (and COMB), $\gamma = 0.8$ and $\delta = 0.2$ in Eq. (13); in CNT, $\epsilon = 0.7$ in Eq. (23); and in COMB, $\alpha' = 2$, $\beta' = 5$, and $\zeta' = 4$ in Eq. (25). In CNT (and COMB), when calculating class frequency in the context of entity mention according to Eq. (19), we let E_{cnt} be the set of all other entities linked to from entity mentions in the document according to the gold standard.

7.4 Extrinsic Evaluation

In extrinsic evaluation, 30 students majoring in computer science and technology were paid to carry out entity linking tasks based on structured descriptions of candidate entities or their summaries generated by different approaches. Each subject was assigned a total of 72 tasks, or 36 tasks from each of the two text corpora. Tasks were randomly selected and were controlled to share no common document in order to minimize the correlation between them. Specifically, based on each of the six approaches that were arranged in random order, the subject was assigned 6 consecutive tasks without knowing the name of the approach (i.e., approaches were blinded). The 6 tasks comprised 1 task as a warmup whose result was not included in subsequent analysis, and 5 tasks whose results were to be analyzed. In each of these tasks, the subject was asked to choose one of the three candidate entities to be linked to from a given entity mention in the document, based on the context of the entity mention (i.e. the content of the document) and the descriptions of candidate entities or their summaries. The subject could

Table 1: Accuracy of Linking and Time per Task on AQUAINT

	Mean (SD)						$F(5, 145)$ (p -value)	LSD post-hoc ($p < 0.05$)
	DESC	CHR	DFF	CNT	COMB	RELIN		
Accuracy	0.91 (0.13)	0.80 (0.17)	0.75 (0.22)	0.71 (0.19)	0.88 (0.13)	0.75 (0.25)	6.586 (0.000)	(1) DESC, COMB > CHR, DFF, CNT, RELIN. (2) CHR > CNT.
Time (s)	59.37 (26.27)	51.26 (24.22)	39.28 (19.77)	52.23 (19.07)	45.63 (16.49)	51.57 (35.36)	3.998 (0.006)	(1) DFF < CHR, CNT, RELIN, DESC. (2) COMB < DESC.

Table 2: Accuracy of Linking and Time per Task on IITB

	Mean (SD)						$F(5, 145)$ (p -value)	LSD post-hoc ($p < 0.05$)
	DESC	CHR	DFF	CNT	COMB	RELIN		
Accuracy	0.89 (0.16)	0.80 (0.18)	0.75 (0.16)	0.78 (0.19)	0.85 (0.17)	0.77 (0.21)	2.598 (0.028)	(1) DESC > CHR, DFF, RELIN. (2) COMB > DFF.
Time (s)	53.88 (19.34)	40.01 (17.94)	47.61 (22.04)	44.23 (18.79)	41.90 (16.34)	43.82 (18.33)	3.335 (0.012)	(1) CHR, CNT, COMB, RELIN < DESC. (2) CHR < DFF.

also respond “not sure” if the provided information was considered insufficient for making a decision.

We defined the accuracy of linking as the percentage of tasks in which the subject chose the target entity instead of a noise entity or “not sure”. Table 1 and 2 present the mean and standard deviation (SD) of the accuracy of linking achieved and the time per task used by all the subjects based on entity descriptions or their summaries generated by each of the six approaches on AQUAINT and IITB, respectively. Repeated measures ANOVA (F and p) revealed that the differences in mean accuracy and mean time based on different approaches on the two text corpora were all statistically significant ($p < 0.05$). In particular, higher accuracy but more time were, as expected, observed on DESC than all the other approaches to summarization. On AQUAINT, Fisher’s Least Significant Difference (LSD) post-hoc analysis revealed that, compared with DESC, the loss of accuracy on CHR, DFF, CNT, and RELIN was statistically significant ($p < 0.05$); that is, summaries generated by the three proposed approaches from different single perspectives and the state-of-the-art generic approach could not effectively substitute entire entity descriptions. By comparison, the loss of accuracy on COMB ($0.91 - 0.88 = 0.03$) was not statistically significant ($p < 0.05$) according to LSD post-hoc analysis; that is, summaries generated by the proposed combined approach were as effective as entire entity descriptions. Meanwhile, compared with DESC, the mean time used for a task on COMB decreased from 59.37 to 45.63 seconds, or by 23%, and LSD post-hoc analysis revealed that this difference was statistically significant ($p < 0.05$). Similar results were observed on IITB. In particular, compared with DESC, the mean time used for a task on COMB significantly ($p < 0.05$) decreased from 53.88 to 41.90 seconds, or by 22%, whereas the loss of accuracy ($0.89 - 0.85 = 0.04$) was not statistically significant ($p < 0.05$). *To conclude, summaries generated by our combined approach helped human users carry out entity linking tasks more efficiently (22–23% faster), without significantly affecting the quality of links obtained, whereas existing approaches to summarizing entity descriptions like RELIN could not achieve this.*

7.5 Intrinsic Evaluation

In intrinsic evaluation, the 30 students directly rated and commented on summaries generated by different approaches. Each subject was assigned a total of 10 tasks, or 5 task from each of the two text corpora. Tasks were randomly selected and were controlled to share no common document in order to minimize the correlation between them. In each task, summaries generated by CHR, DFF, CNT, COMB, and RELIN were all presented, and the subject was blind to the correspondences between summaries and approaches. The subject was asked to compare these summaries and rate each of them on a scale of 1 to 5, indicating its effectiveness in helping make a decision. The subject was also encouraged to comment on summaries having very high or low ratings.

We calculated the average rating given by each subject to the summaries generated by each approach. Table 3 presents the mean and standard deviation (SD) of the average rating given by all the subjects to different approaches on AQUAINT and IITB. Repeated measures ANOVA (F and p) revealed that the differences in rating on the two text corpora were both statistically significant ($p < 0.05$). In particular, COMB received the highest rating on both text corpora, followed by CHR. Both of their ratings were higher than the other three approaches (namely DFF, CNT, and RELIN). LSD post-hoc analysis revealed that these differences were statistically significant ($p < 0.05$). *To conclude, summaries generated by our combined approach were considered more effective in helping human users carry out entity linking tasks than existing approaches to summarizing entity descriptions like RELIN. In particular, characterizing power and information overlap (which make up CHR) were dominating factors. Differentiating power and relevance to context were also complementary factors.*

We also summarized all the major comments that were made by at least fifteen subjects (50%).

- As to CHR, on IITB, 16 subjects (53%) commented that their decision was facilitated by some highly distinguishing features.
- As to DFF, on IITB, 15 subjects (50%) commented that different types helped them filter out noise entities easily. However, on AQUAINT, 18 subjects (60%)

Table 3: Ratings of Summaries on AQUAINT and IITB

	Mean (SD)					$F(4, 116)$ (p -value)	LSD post-hoc ($p < 0.05$)
	CHR	DFE	CNT	COMB	RELIN		
AQUAINT	3.75 (0.60)	2.78 (0.98)	3.03 (0.82)	4.02 (0.52)	3.20 (0.75)	18.499 (0.000)	(1) COMB > CHR > DFF,CNT,RELIN.
IITB	4.01 (0.55)	2.86 (0.89)	2.84 (0.87)	4.27 (0.48)	3.29 (0.76)	35.890 (0.000)	(1) COMB > CHR > RELIN > DFF,CNT.

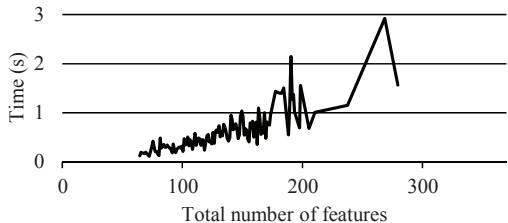


Figure 2: Average running time of the heuristic solution to QMKP in COMB on a task from AQUAINT.

commented that, apart from different types, almost no useful information was provided.

- As to COMB, on AQUAINT and IITB, 24 subjects (80%) and 27 subjects (90%) commented that their decision was facilitated by some highly distinguishing features, respectively. On IITB, 16 subjects (53%) also commented that comprehensive information was provided, which was very helpful.

7.6 Discussion

We conclude extrinsic and intrinsic evaluation with the following insights.

Firstly, as shown in both Table 1 and 2, in extrinsic evaluation, the mean accuracy of linking achieved based on DESC was high but did not reach 1.00; that is, subjects sometimes made wrong decisions even based on entire entity descriptions. A major reason was that some entity linking tasks were inherently difficult. For instance, in some task, an entity mention “Nice” was expected to be linked to the city in France but not the town in California, because in the document, Saint-Laurent-du-Var was also mentioned and this town was a suburb of the city of Nice in France. Subjects made wrong decisions in this task because they neither had the above background knowledge nor could find it in the document or in Nice’s entity description. To solve it, more information needs to be provided to the subject by, for instance, expanding entity descriptions wisely.

Secondly, in intrinsic evaluation, more than half (53%) of subjects commented that “comprehensive” information was provided by COMB on IITB. Here we tend to interpret comprehensive information as multiple paths that could lead to decision. COMB achieved this because it combined three different perspectives, generating characteristic, differential, and contextual summaries. Also for this reason, in extrinsic evaluation, it outperformed CHR, DFF, and CNT, all of which were only from single perspectives.

Thirdly, whereas the expected space savings produced by summarization in our experiments was 85.30%, in extrinsic evaluation, the mean time used for a task based on COMB

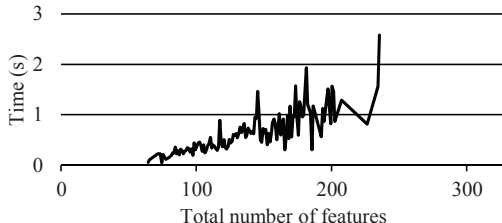


Figure 3: Average running time of the heuristic solution to QMKP in COMB on a task from IITB.

decreased by only 22–23% compared with DESC. A major reason was that, apart from descriptions of candidate entities or their summaries, subjects consistently needed to spend time reading the context of entity mention, which could be the entire document typically containing thousands of characters in our experiments. Therefore, it motivates us to summarize not only candidate entity descriptions but also the context of entity mention in future work.

7.7 Running Time

We tested the running time of our approach, although summaries could be generated off-line in practical applications and thus its running time would not be a main concern. We only tested the running time of COMB, the most complex approach we proposed. It was written in Java and ran on an Intel Xeon E3-1225 v2 with 512MB memory for JVM. We found that its running time was dominated by the heuristic solution to QMKP. Figure 2 and 3 show the average running time of this computation on 500 random tasks from AQUAINT and IITB, respectively; and the results are grouped by the total number of features of all the three candidate entities in a task. Our implementation of COMB was reasonably fast, since it took less than 3 seconds in all the cases and less than 2 seconds in most cases.

8. RELATED WORK

8.1 Entity Linking

Entity linking is the task of linking an entity mention in text to its corresponding entity in a knowledge base. A large body of work has been devoted to automatically generating candidate entities, or ranking and choosing from them [16]. However, here we will not discuss any of these specific automatic approaches to entity linking, because the goal of our work is orthogonal to all of them and we focus on human-centered entity linking. Nevertheless, *our work can be beneficial to the evaluation of automatic approaches to entity linking* because such evaluation is usually based on manually defined gold-standard links obtained from human experts carrying out entity linking tasks [4, 10], and our approach

can facilitate human-centered entity linking by substituting entire descriptions of candidate entities with compact, equally effective summaries, thereby improving human experts' productivity.

Another line of research involves humans in the loop, and combines algorithmic and manual efforts instead of pursuing a fully automatic solution. For instance, ZenCrowd [5] dynamically generates entity linking tasks and publishes them on an online crowdsourcing platform, and employs a probabilistic framework to integrate links contributed by human users which are possibly inconsistent. *Our work well complements existing crowdsourcing techniques for entity linking* because ZenCrowd does not focus on tool support on the user side for carrying out each single entity linking task, whereas our approach exactly targets this issue, and improves user experience by largely reducing the amount of information that needs to be consumed by human users, without significantly affecting the quality of links obtained. Therefore, the payment can also be reduced.

8.2 Summarizing Entity Descriptions

Summarizing structured descriptions of entities, a.k.a. entity summarization, has proven to be useful in many applications. For instance, in entity search engines, entity descriptions in search results pages are summarized to concisely indicate their relevance to the keyword query, by selecting features that contain query keywords [2] or by selecting features via more sophisticated machine learning techniques [21]. In interactive entity resolution, entity descriptions are summarized to help human users more efficiently judge whether they refer to the same real-world entity, by selecting features that reflect commonalities shared by and conflicts between them, and that carry characteristic and diverse information about them [19]. Different from all these efforts, *our work aims to summarize entity descriptions for a different task, namely human-centered entity linking. Accordingly, our approach deals with different types of data (e.g. context of entity mention) and thus employs different techniques (e.g. class vector model).*

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 based on their informativeness as well as the relatedness between them. DIVERSUM [18] improves the diversity of a summary by not selecting features sharing a common property. In the database community, all data held in a database about a particular data subject is summarized based on affinity of relations and attributes [6, 7, 8]. Although summaries generated by these approaches can also be used or adapted for the task of human-centered entity linking, *summaries generated by our specifically developed approach have proven to be more effective* according to the accuracy of linking achieved and the ratings given by human users in extrinsic and intrinsic evaluation, respectively.

9. CONCLUSIONS AND FUTURE WORK

To facilitate human-centered entity linking, we have proposed to substitute entire descriptions of candidate entities with compact summaries. Extrinsic evaluation demonstrates that summaries generated by our approach help human users carry out entity linking tasks more efficiently (22-23% faster), without significantly affecting the quality of links obtained. Next, we are interested in whether and how

our summaries are also beneficial to automatic approaches to entity linking, which will be explored in future work.

Our characteristic, differential, and contextual summaries select features based on their characterizing and differentiating power, the information overlap between them, and their relevance to the context of entity mention. This multi-perspective approach outperforms approaches from single perspectives and a state-of-the-art generic approach, in terms of accuracy of and time for linking in extrinsic evaluation, and in terms of rating in intrinsic evaluation. For future work, our experimental design can be extended in several directions, e.g., to test with different numbers of candidate entities (including not-in-the-list) in extrinsic evaluation, and to solicit multi-dimensional ratings in intrinsic evaluation. Another attempt will be to create gold-standard summaries to enable automatic evaluation without humans in the loop.

10. ACKNOWLEDGMENTS

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. The authors would like to thank all the participants and anonymous reviewers.

11. REFERENCES

- [1] J. Carbonell and J. Goldstein. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Proc. SIGIR*, pages 335–336, August 1998.
- [2] G. Cheng and Y. Qu. Searching linked objects with Falcons: Approach, implementation and evaluation. *Int'l J. Semant. Web Inf. Syst.*, 5(3):49–70, July–September 2009.
- [3] G. Cheng, T. Tran, and Y. Qu. RELIN: Relatedness and informativeness-based centrality for entity summarization. In *Proc. ISWC*, pages 114–129, October 2011.
- [4] M. Cornolti, P. Ferragina, and M. Ciaramita. A framework for benchmarking entity-annotation systems. In *Proc. WWW*, pages 249–260, May 2013.
- [5] G. Demartini, D. Difallah, and P. Cudré-Mauroux. ZenCrowd: Leveraging probabilistic reasoning and crowdsourcing techniques for large-scale entity linking. In *Proc. WWW*, pages 469–478, April 2012.
- [6] G.J. Fakas. A novel keyword search paradigm in relational databases: Object summaries. *Data Knowl. Eng.*, 70(2):208–229, February 2011.
- [7] G.J. Fakas. Size-1 object summaries for relational keyword search. *Proc. VLDB Endowment*, 5(3):229–240, November 2011.
- [8] G.J. Fakas, Z. Cai, and N. Mamoulis. Versatile size-1 object summaries for relational keyword search. *IEEE Trans. Knowl. Data Eng.*, 26(4):1026–1038, April 2014.
- [9] D. Graff. The AQUAINT corpus of English news text LDC2002T31. Linguistic Data Consortium, September 2002.
- [10] B. Hachey, W. Radford, J. Nothman, M. Honnibal, and J. Curran. Evaluating entity linking with Wikipedia. *Artif. Intell.*, 194:130–150, January 2013.
- [11] S. Kulkarni, A. Singh, G. Ramakrishnan, and S. Chakrabarti. Collective annotation of Wikipedia

- entities in Web text. In *Proc. KDD*, pages 457–466, June–July 2009.
- [12] J. Lehmann, R. Isele, M. Jakob, A. Jentzsch, D. Kontokostas, P. Mendes, S. Hellmann, M. Morse, P. van Kleef, S. Auer, and C. Bizer. DBpedia - A large-scale, multilingual knowledge base extracted from Wikipedia. *Semant. Web J.*, to appear.
- [13] D. Milne and I. Witten. Learning to link with Wikipedia. In *Proc. CIKM*, pages 509–518, October 2008.
- [14] D. Pisinger. The quadratic knapsack problem - A survey. *Discrete Appl. Math.*, 155(5):623–648, March 2007.
- [15] G. Salton, A. Wong, and C. Yang. A vector space model for automatic indexing. *Commun. ACM*, 18(11):613–620, November 1975.
- [16] W. Shen, J. Wang, and J. Han. Entity linking with a knowledge base: Issues, techniques, and solutions. *IEEE Trans. Knowl. Data Eng.*, 27(2):443–460, February 2015.
- [17] G. Stoilos, G. Stamou, and S. Kollias. A string metric for ontology alignment. In *Proc. ISWC*, pages 624–637, November 2005.
- [18] M. Sydow, M. Piłkuła, and R. Schenkel. The notion of diversity in graphical entity summarisation on semantic knowledge graphs. *J. Intell. Inf. Syst.*, 41(2):109–149, October 2013.
- [19] D. Xu, G. Cheng, and Y. Qu. Facilitating human intervention in coreference resolution with comparative entity summaries. In *Proc. ESWC*, pages 535–549, May 2014.
- [20] Z. Yang, G. Wang, and F. Chu. An effective GRASP and tabu search for the 0-1 quadratic knapsack problem. *Comput. Oper. Res.*, 40(5):1176–1185, May 2013.
- [21] L. Zhang, Y. Zhang, and Y. Chen. Summarizing highly structured documents for effective search interaction. In *Proc. SIGIR*, pages 145–154, August 2012.