CAMO: Integration of Linked Open Data for Multimedia Metadata Enrichment

Wei Hu[1], Cunxin Jia[1], Lei Wan[2], Liang He[2], Lixia Zhou[2], and Yuzhong Qu[1]

[1] State Key Laboratory for Novel Software Technology, Nanjing University, China
{whu, yzqu}@nju.edu.cn, jiacunxin@smail.nju.edu.cn
[2] Samsung Electronics (China) R&D Center, China
{l.wan, jaimely.he, lixia.zhou}@samsung.com

Abstract. Metadata is a vital factor for effective management, organization and retrieval of multimedia content. In this paper, we introduce CAMO, a new system developed jointly with Samsung to enrich multimedia metadata by integrating Linked Open Data (LOD). Large-scale, heterogeneous LOD sources, e.g., DBpedia, LinkMDB and MusicBrainz, are integrated using ontology matching and instance linkage techniques. A mobile app for Android devices is built on top of the LOD to improve multimedia content browsing. An empirical evaluation is conducted to demonstrate the effectiveness and accuracy of the system in the multimedia domain.

Keywords: Linked Data, multimedia, semantic data integration

1 Introduction

Multimedia metadata and semantic annotation are vital to improve services on multimedia content [21]. The search, browsing and management of multimedia content become very difficult if no or only limited metadata and annotations are provided. Driven by the Linking Open Data Initiative, plenty of open datasets are published and interlinked, in order to enable users to make use of such rich source of information [22].

Looking at the existing multimedia metadata models and standards, they do not provide formal semantics and typically focus on a single media type. For example, EXIF is widely used for image description, but it is incompatible with MPEG-7 [21]. In real world, different media types often coexist in a multimedia presentation, where for example a movie may have a theme music and a poster. We believe that a unified, well-defined ontology (with its mappings to others) is needed in many multimedia application scenarios to gain interoperability. Additionally, metadata from diverse data sources can denote the same multimedia content. Linking and integrating these heterogeneous datasets are challenging, especially when meeting legacy data reserved in relational databases (RDBs) or on the Deep Web. Thus, accurate methods are desired to (semi-)automatically link the overlapping parts of the datasets. The integrated metadata can provide benefits to many multimedia applications like mobile devices, whose market is expected to rise to $9.5 billion by 2014 [7].

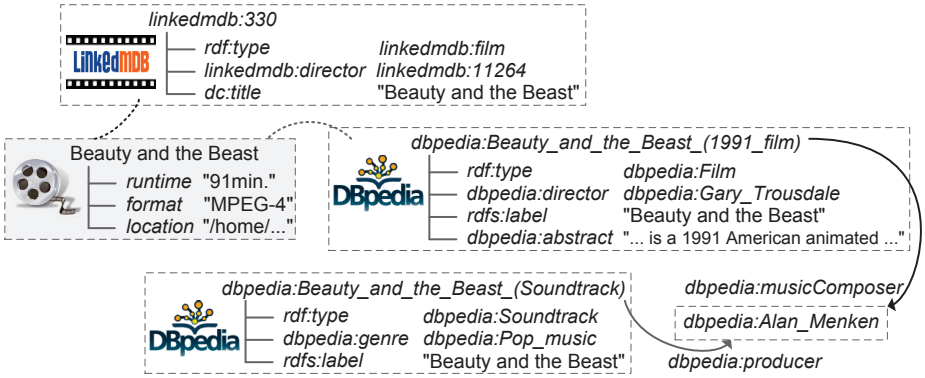


Fig. 1. An example of integrating LOD for multimedia metadata enrichment

A motivating example. Fig. 1 illustrates a real-world example about the movie *Beauty and the Beast*. The original video already has a few low-level metadata like runtime and location. By integrating LOD, e.g., LinkedMDB [10] and DBpedia [17], the description of this movie would be enriched significantly. However, LinkedMDB and DBpedia use different but related ontologies for movie description, thus creating mappings between their classes and properties is important for integrating the descriptions into the movie metadata. Additionally, DBpedia and LinkedMDB refer to the same movie by using different instances, e.g., `dbpedia:Beauty_and_the_Beast_(1991_film)` and `linkedmdb:330` in this example. But it may not be sufficient and accurate to only match their titles/labels, where for example a music `dbpedia:Beauty_and_the_Beast_(Soundtrack)` with exactly the same label should not be linked. □

In this paper, we describe CAMO, a system developed jointly with Samsung for enriching multimedia metadata via integrating LOD. CAMO achieves this by using our ontology matching and instance linkage techniques and adapting them to the multimedia domain. The technical contributions of this paper are threefold: (i) CAMO selects the DBpedia ontology as the mediation model and matches with other ontologies; (ii) CAMO links the instances in DBpedia with other sources and aggregates their descriptions; (iii) CAMO incorporates RDBs with DBpedia to cope with legacy data. We hope that our methods and system can provide reusable experience for applications consuming Linked Data.

We develop a mobile app for browsing and searching multimedia content on Android devices. We perform a user-centered evaluation of CAMO to measure how well it compares with existing apps, in particular with Last.fm, IMDb and Wikipedia mobile apps. We also conduct an experiment on the accuracy of the ontology matching and instance linkage in the multimedia domain. The results demonstrate the advantages of integrating LOD into multimedia metadata for improving the quality of multimedia content services. More information about CAMO is available at `http://ws.nju.edu.cn/camo/`.

The remainder of this paper is structured as follows. Section 2 outlines the system architecture of CAMO and the used LOD sources. In Section 3 and 4, we present the methods to match ontologies and link instances with DBpedia, respectively. Section 5 describes the approach for incorporating legacy RDBs. Evaluation is reported in Section 6 and related work is discussed in Section 7. Finally, we conclude this paper and summarize the lessons learned.

2 System Architecture

The architecture of CAMO is illustrated in Fig. 2, which follows a widely-used Client-Server paradigm providing the system with high bandwidth, processing and storage on a large amount of data. For the server side, we choose the DBpedia 3.6 ontology as the mediation and use the Global-as-View solution of data integration, because it is efficient for query rewriting and the used LOD sources are relatively stable. Various LOD sources are all integrated with DBpedia by the ontology matching and instance linkage techniques. The used LOD sources are chosen in terms of popularity. Due to the unstable availability of SPARQL endpoints [4], we currently materialize the original data from their dump files.

DBpedia. DBpedia [17] is a crowd-sourced community effort to extract structured, multi-lingual information from Wikipedia and make this information available on the Web. The reason to choose the DBpedia ontology is that it is generic enough to encapsulate various kinds of multimedia domains and can be matched with a large number of ontologies [14]. Besides the ontology, the instances itself are also comprehensive. This is another reason for choosing it rather than other ontologies such as M3O [21].

DBTune. DBTune[3] is a non-commercial site, which hosts a number of servers providing access to music-related structured data in a Linked Data fashion. We choose three datasets, namely Jamendo, Magnatune and BBC John Peel session, from DBTune because they already provide links to DBpedia. Note that our approaches are also ready for integrating other datasets.

LinkedMDB. The LinkedMDB project [10] aims at publishing an open Semantic Web database for movies, including a large quantity of links to several datasets on the LOD cloud and references to related webpages.

DBTropes. DBTropes[4] transforms numerous movies, books and other pages to RDF with the Skipinions ontology.

MusicBrainz. MusicBrainz[5] is an open music encyclopedia that collects music metadata and makes it available to the public. Different from other sources providing data in RDF, the MusicBrainz Database is built on RDB (although the LinkedBrainz project[6] helps MusicBrainz transform to Linked Data). We will present how to integrate it in Section 5.

[3] `http://dbtune.org/`.
[4] `http://dbtropes.org/`.
[5] `http://musicbrainz.org/`.
[6] `http://linkedbrainz.org/`

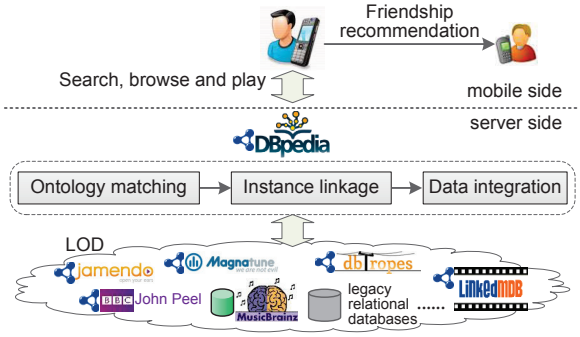


Fig. 2. System architecture of CAMO

For the client side, we build the client on Android-based mobile devices[7] and integrate it with a multimedia player. General users can search and browse multimedia content with the enriched metadata, and the metadata is displayed in an integrated view. Additionally, we provide several value-added functionalities, e.g., a Horn-rule based friendship recommendation based on users' favorites and play histories. The recommendation rules are customizable in terms of application requirements, and we implement a rule-based inference engine by ourselves. We omit the details of this rule-based recommendation engine in this paper.

3 Matching Ontologies with DBpedia

Different LOD sources have different preferences on ontologies, some of which prefer to develop their own ontologies from scratch to meet their requirements rather than reuse existing ones. Among the LOD sources integrated in CAMO, LinkedMDB and DBTropes define their own ontologies. Meanwhile, other LOD sources reuse some famous ontologies as their conceptual models. For example, the Music Ontology [19] is chosen as the underlying ontology by Jamendo, Magnatune and BBC John Peel. Whichever ontology a data source uses, in order to query and browse the distributed multimedia metadata, a necessary phase is to match ontologies for resolving the heterogeneity between them.

To match ontologies with DBpedia, we use Falcon-AO [12], which is an automatic ontology matching system. The methodological steps of matching ontologies with DBpedia are depicted in Fig. 3, where the output is a set of mappings between the classes or properties in two ontologies. The strength of Falcon-AO is that it leverages various powerful matchers, not only the linguistic matchers like V-Doc but also the structural matcher GMO. S-Match [9] is an alternative system for this purpose. We extend Falcon-AO with domain knowledge to support synonym identification in the multimedia domain, e.g., `track` and `song`.

[7] Also because Samsung is a leading company in Android-based mobile devices.

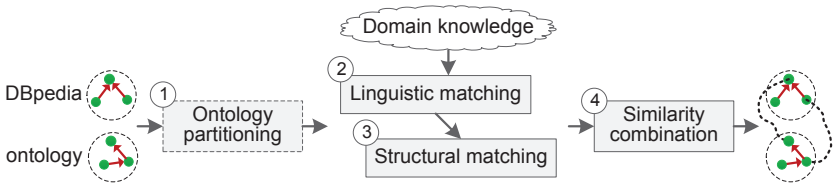


Fig. 3. Methodological steps of matching ontologies with DBpedia

Before the matching step, an optional partitioning step is involved to cope with large ontologies. We propose a divide-and-conquer approach for generating block mappings between large ontologies, which has two main advantages: (i) it avoids the matching algorithms suffering from the lack of memory; and (ii) it decreases the running time without much loss of accuracy, since it is likely that large portions of one or both ontologies have no matched counterparts. Specifically, all the classes and properties of an ontology are firstly partitioned into a set of small blocks based on the structural proximity (e.g., the distance between classes in the hierarchy, the overlapping between the domains of properties). Then, the blocks are matched using some prefound mappings between classes or properties; only the block pairs with high similarity are further matched with the linguistic and structural matchers.

Linguistic features are widely used for matching ontologies. We employ two linguistic matchers, V-Doc and I-Sub, to calculate the linguistic similarity. V-Doc is a TF-IDF based matcher, representing each class or property as a virtual document (a bag of weighted words). Local descriptions $LD()$ and neighboring information are both considered in V-Doc. For a literal, its description is a collection of words derived from the lexical form; for a named class or property, it is a collection of words extracted from the local name of its URI, `rdfs:label`(s), `rdfs:comment`(s) and other annotations; and for a blank node, it is a collection of words extracted from the information originated from the forward neighbors. To incorporate the descriptions of neighbors in virtual documents, we use three neighboring operations to cover different kinds of neighbors: subject neighbors $SN()$, predicate neighbors $PN()$ and object neighbors $ON()$. Also, synonyms are replaced to refine the documents. Let e be a named class or property. The virtual document of e, denoted by $VD(e)$, is defined as follows:

$$VD(e) = LD(e) + \gamma_s * \sum_{e' \in SN(e)} LD(e') + \gamma_p * \sum_{e' \in PN(e)} LD(e') + \gamma_o * \sum_{e' \in ON(e)} LD(e'), \quad (1)$$

where $\gamma_s, \gamma_p, \gamma_o$ are in $[0, 1]$. The measure in V-Doc to determine if two classes or properties are similar is the cosine similarity of their virtual documents.

I-Sub [24] is an improved string matcher considering not only the commonalities between the descriptions of classes or properties but also their differences.

V-Doc and I-Sub are combined linearly. We find that setting the weightings to 0.8 and 0.2 for V-Doc and I-Sub respectively achieves a good accuracy.

Another popular type of matchers is structure-based. A graph-based matcher GMO is employed in Falcon-AO. GMO transforms each ontology into a directed bipartite graph and measures the structural similarity between the two graphs. GMO accepts as input the mappings that are prefound by the linguistic matchers, and iteratively yields more mappings through similarity propagation on the graphs as output.

To meet different matching scenarios, we design a flexible similarity combination strategy based on the measures of both linguistic and structural comparability. The linguistic comparability is assumed to be more reliable, specifically, if the linguistic comparability is high enough, indicating the matching is almost done, there is no need to run the structural matcher any longer. Nevertheless, when the two linguistic matchers fail to find enough candidates, the structural matcher becomes the primary choice.

4 Linking Instances with DBpedia

Matching ontologies with DBpedia enables it to query and browse multimedia metadata from the global view. However, overlaps among the LOD sources at the instance level are inevitable, due to the distributed nature of the Semantic Web. Hence, instance linkage is helpful to merge all the descriptions in different sources that refer to the same multimedia content. Complementary information from distributed sources helps understand the content more comprehensively.

As of today, a portion of instances among LOD sources have been explicitly interlinked with `owl:sameAs`, however, there still exist plenty of instances that potentially denote the same real-world objects without being linked yet. Linking them manually is an uphill work. Therefore, we propose an automatic method to learn instance links between DBpedia and other LOD sources based on a set of important properties for characterizing instances (referred to as discriminative properties) [11]. The methodological steps are depicted in Fig. 4.

The first step is to construct a training set automatically. Five vocabulary elements, i.e., `owl:sameAs`, `skos:exactMatch`, inverse functional property (IFP), functional property (FP) and (max-)cardinality, are considered, which are widely used to infer the equivalence relation in many instance linkage systems [13]. We implement the Floyd-Warshall algorithm to obtain the transitive closure of the equivalence relation between instances, which contribute "positive examples" to the training set. As reported in [11], an instance only links to a small number of others, so finding positive examples is computationally cheap at large scale.

The instances, which do not hold the equivalence relation, are useful to learn non-discriminative properties as well. However, the number of instances explicitly claimed as different using the vocabulary elements like `owl:differentFrom` is small, which is inadequate for a reasonable-sized training set. Therefore, we approximately regard instance pairs that cannot infer the equivalence relation as denoting different real-world objects. However, this way generates tremendous

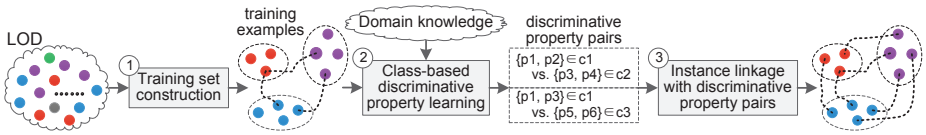


Fig. 4. Methodological steps of linking instances with DBpedia

"negative examples" and most of them are totally orthogonal. Hence, a tailoring strategy is introduced to eliminate superfluous negative examples. Besides, this approximation may involve wrong negative examples to some extent, because positive examples include false negatives. But considering the significant difference between the sizes of positive and negative examples, the number of wrong negatives is typically rare.

Discriminative properties are important to link instances, which are learned with a class-based way from the training set. We extract the descriptions of the instances in the training set and pairwise compare them with V-Doc. The discriminability of a property pair is measured by information gain, which computes the change in information entropy from the original state of the training set to a state that uses the properties to identify instance links. The information gain measure is widely used for classification. The discriminability of a property pair is refined w.r.t. different classes due to the different preferences of data publishers on the use of properties, and domain knowledge is used for reasoning types. Let $\mathbf{D}$ be the training set (including both positive and negative examples) satisfying that the types of the instances in each instance pair are c_i and c_j, respectively. For a property pair (p_i, p_j), we select all instance pairs (u_i, u_j) in $\mathbf{D}$, denoted by $\mathbf{D}_{(p_i,p_j)}$, where u_i involves p_i, and u_j involves p_j. The discriminability of (p_i, p_j) in $\mathbf{D}$ w.r.t. (c_i, c_j) is measured by the information gain $IG()$ as follows:

$$IG(p_i, p_j) = H(\mathbf{D}) - H(\mathbf{D}_{(p_i,p_j)}), \tag{2}$$

where $H(\mathbf{D})$ measures the information entropy of $\mathbf{D}$, while $H(\mathbf{D}_{(p_i,p_j)})$ measures the information entropy using (p_i, p_j) to classify instance pairs in $\mathbf{D}$.

With discriminative properties, the instance linkage phase can be conducted online. Given two instances to be linked, the first step is to retrieve the types of them. Then, the most discriminative properties w.r.t. the types are queried out (this can be treated as a blocking step). Finally, a link is generated if the linear aggregation of the similarity of the values from the discriminative properties is greater than a pre-fixed threshold.

The descriptions of linked instances in different sources are integrated and displayed in a structured and compressed way. Firstly, the linked instances are grouped and all their descriptions are retrieved. Then, the properties with the same value are clustered together. For the properties matched in the ontology matching phase, their values are merged if matched, and we preferentially show the properties and values in DBpedia. But the descriptions can also be enriched by other data sources. Additionally, with the provenance information for each

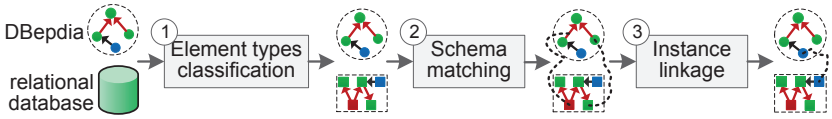


Fig. 5. Methodological steps of integrating RDBs with DBpedia

value, users are capable of determining which source is more trustworthy when encountering inconsistency.

5 Integrating Legacy Relational Databases

Despite the amount of multimedia-related sources in the LOD cloud is considerable, there are still a great deal of legacy data stored in RDBs, such as from the multimedia content providers in Samsung for years, as well as MusicBrainz. Moreover, some data sources (e.g., LinkedMDB and LinkedBrainz) in LOD are published as Linked Data from their relational versions. To address legacy multimedia metadata using the relational model, we propose a lightweight method to integrate RDBs with DBpedia [15]. Fig. 5 shows the methodological steps.

Due to the differences in data models, in the first step an element classifier takes as input an ontology and a relational schema, and classifies the elements in them into different categories. Tables in a relational schema are categorized in two types in terms of their primary keys and foreign keys: entity table and relationship table. An entity table is used to represent a class of instances, and can match a class in the ontology intuitively. A relationship table, which connects entity tables to reveal their relationships, can match an object property in the ontology. Columns in an entity table can be regarded as non-foreign keys and foreign keys. Non-foreign keys can match datatype properties, while foreign keys can match object properties. This step is also called reverse engineering.

The next step is to match the elements in each category by reusing the V-Doc matcher. As mentioned before, V-Doc is a linguistic matcher considering both local descriptions and neighboring information. The virtual documents of the elements in the relational schema and the ontology are built and compared with the cosine similarity measure. The elements holding the similarity greater than a pre-defined threshold are considered as element mappings, which are expressed using the W3C R2RML language.

Instances in the RDB are linked using a similar way to that for LOD. The tables and columns matched in the previous schema matching step are treated as classes and properties. Then, instance links are generated by comparing the values of class-based discriminative properties and aggregating their similarity. This step can also be online as long as the discriminative properties are learned.

There are also some existing systems, e.g., D2RQ,[8] which support SPARQL queries against non-RDF databases; however, they are not well suitable for inte-

[8] `http://d2rq.org/`

grating heterogeneous data sources. Additionally, D2R is an alternative system that requires user-defined mappings to match with RDBs.

6 Evaluation

CAMO is a system to integrate LOD for multimedia metadata enrichment. To evaluate its effectiveness and accuracy, we conduct two kinds of experiments: (i) the usability and effectiveness of the mobile app of CAMO are compared with several popular apps in a user-centered way, and (ii) the integration accuracy is evaluated in the multimedia domain using the well-known precision and recall.

The mobile app is deployed on a Samsung Galaxy S3 with 4.8 inch screen, 1GB RAM and Android OS 4.0, and the data are stored on a server with two Xeon E7440 2.4GHz CPUs and 4GB memory for JVM, using Apache Jena and PostgreSQL. The integration approaches also run on the server. The user interface of CAMO is shown in Fig. 6, where the integrated metadata for the movie *Beauty and the Beast* is displayed with the provenance.

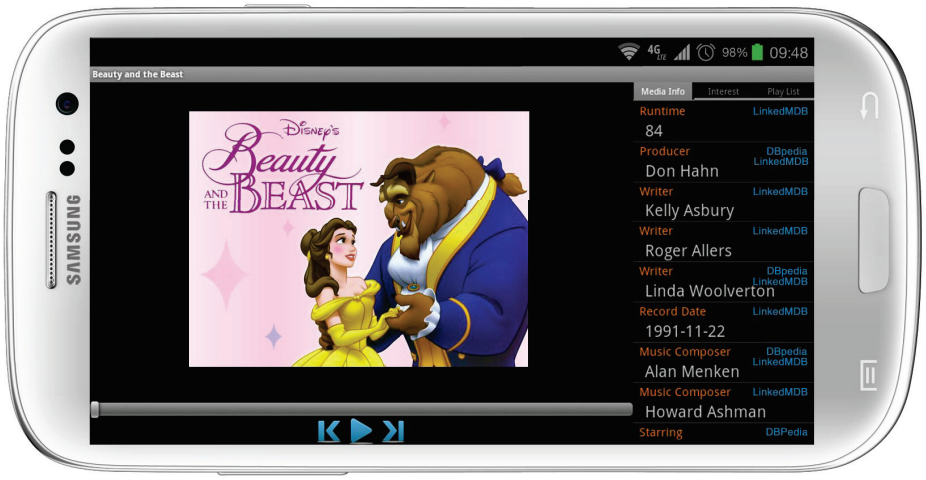


Fig. 6. User interface of CAMO on a mobile phone

6.1 Evaluation on Usability and Effectiveness

Experimental methodology. We choose three mobile apps in the Google Play app store for comparison: Last.fm, IMDb and Wikipedia, whose underlying datasets are very similar to those integrated in CAMO (see Section 2). We introduce the three apps briefly as follows:

– Last.fm provides a mobile app that has the capability to learn about users' favorite artists, find nearby concerts and share music tastes with the Last.fm library. Last.fm uses the FOAF ontology for user profiles and MusicBrainz for music metadata.
– IMDb is a database of movie, TV and celebrity information. There are more than two million movie and TV titles in IMDb. The IMDb Android app has the features like search, rating and browsing.
– The Wikipedia Android app is open-sourced and developed mainly by JavaScript. Users of supported mobile devices are automatically redirected to the mobile version of Wikipedia.

We design six testing tasks of three groups: music, movie and cross-domain, which are listed in Table 1. Users are asked to use the four mobile apps aforementioned to complete a randomly designated task of each group. The tasks in the music group are performed on Last.fm and CAMO, while the tasks in the movie group are assigned to compare IMDb and CAMO. Cross-domain tasks regarding both music and movie are accomplished by Wikipedia and CAMO. The cross-domain tasks may also be done by the collaboration of Last.fm and IMDb, but it is burdensome for users, so we leave this out of consideration. The tasks are chosen due to their high popularity among university students.

Table 1. Tasks for usability and effectiveness assessment

Domain	Task description
Music	$T1$. X is a *Lady Gaga*'s song whose name is started with letter "P". Please find the album of X.
	$T2$. X is a *Coldplay*'s song whose name is started with letter "Y". Please find the writer of X.
Movie	$T3$. X is the producer of *The Godfather*. Please find X's name and any two films for which X won the Academy Award.
	$T4$. X is the music composer of *The Terminator*. Please find X's name and any two films of which X was also the music composer.
Cross-domain	$T5$. X is the director of *Michael Jackson*'s movie *Michael Jackson's That Is It*, and Y is the album of *Michael Jackson*'s song *Beat It*. Please find the names of X and Y, respectively.
	$T6$. X is the distributor of *Will Smith*'s movie *The Pursuit of Happiness*, and Y is an *Will Smith*'s album named "Born to Reign". Please find X's name and the release date of Y.

We invite 50 users to participate in the evaluation. 10 of them are graduate students in our group and have adequate knowledge of the Semantic Web and LOD; another 22 users are undergraduate students randomly picked up in our university; the rest 18 users are software engineers in Samsung. The users with different backgrounds reflect diversity.

Before starting a task, the users are asked to score the familiarity and difficulty about the task. The time limit for each task is 5 minutes. When finishing the task, a user is required to fill a System Usability Scale (SUS) questionnaire and a post-task questionnaire. SUS is a reliable and low-cost way to assess system usability. The post-task questionnaire is shown in Table 2. These questions are designed to evaluate the quality, diversity and coverage of the underlying metadata for the four mobile apps.

Table 2. Post-task questionnaire

Question description	Score (1–5)
$Q1$. The app has an accurate description about content.	1 for strongly disagree, and 5 for strongly agree.
$Q2$. The app has a comprehensive coverage about content.	
$Q3$. The app helps me easily find content that I am interested in.	
$Q4$. The app provides few redundant and irrelevant information.	
$Q5$. The app often shows me some unexpected facts in browsing.	

Table 3. Scores of SUS

CAMO	87.88
Last.fm	79.81
IMDb	**89.62**
Wikipedia	84.04

Table 4. Scores of post-task questionnaire

	Music		Movie		Cross-domain	
	CAMO	Last.fm	CAMO	IMDb	CAMO	Wikipedia
$Q1$.	**4.92**	4.53	**5.00**	4.92	**5.00**	**5.00**
$Q2$.	**4.69**	2.38	**4.46**	4.38	4.62	**4.69**
$Q3$.	**4.69**	2.92	**4.62**	3.46	**4.77**	3.23
$Q4$.	**4.31**	4.23	**4.38**	3.77	**4.54**	3.31
$Q5$.	**3.61**	2.54	3.54	**4.00**	3.85	**4.15**

Results and discussions. Table 3 lists the average SUS scores of CAMO, Last.fm, IMDb and Wikipedia respectively: 87.88 ($SD = 8.28, median = 90$), 79.81 ($SD = 5.44, median = 80$), 89.62 ($SD = 5.67, median = 90$) and 84.04 ($SD = 4.95, median = 85$). Repeated measures ANOVA indicates that the differences are statistically significant ($p < 0.01$). LSD post-hoc tests ($p < 0.05$) indicate that IMDb and CAMO are more usable than Wikipedia, and Wikipedia is more usable than Last.fm. Although SUS is not an absolute and overall criterion, this result reflects that CAMO is user-friendly in a sense. Besides, all the users are very familiar with the tasks and think them easy.

The result of post-task questionnaire is shown in Table 4. Due to the high-quality multimedia metadata, the scores of all the apps in $Q1$ are close to each other. Last.fm gets the lowest score in $Q1$ and $Q2$ because it gives only limited information about artists. Additionally, 35 users (70%) tell that they are very confused when clicking a song leads to browse the artist of the song rather than the song itself. The competitive performance of CAMO in $Q1$ and $Q2$ reflects the advantages of integrating multimedia metadata from multiple sources.

CAMO outperforms the other apps in $Q3$ and repeated measures ANOVA reveals that the difference is statistically significant ($p < 0.01$). 42 users (84%) think that they can find information that they want from CAMO without much effort. Although all the four apps are capable of keyword search, some features of CAMO makes it more prone to locate content. The first feature is browsing between content via links, which is not implemented in Last.fm. In contrast to Wikipedia, CAMO organizes and displays multimedia metadata in a structured way. Moreover, the properties in CAMO are more plentiful than those of IMDb.

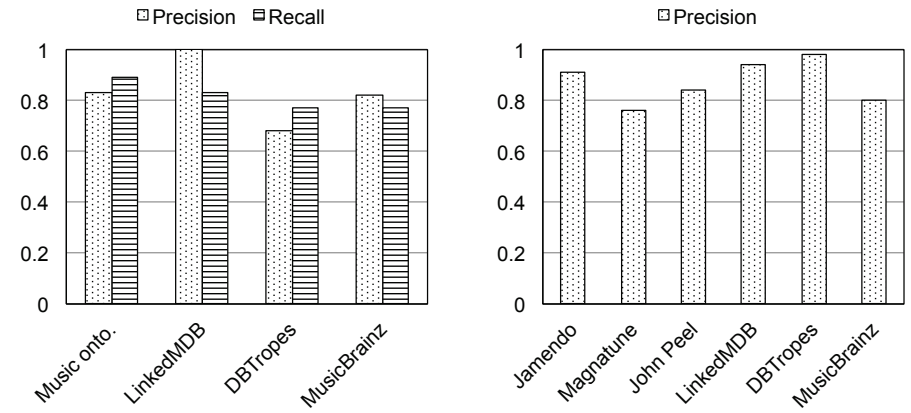
By taking benefits from the structured nature of RDF data, CAMO achieves a higher score in $Q4$. CAMO performs better than Last.fm in $Q5$ but not as well as IMDb and Wikipedia. 28 users (56%) tell that IMDb contains a large amount of interesting, user-generated content like movie reviews and ratings, which are not considered in the current version of CAMO.

We also analyze the result of questionnaire according to the typology of the users. Generally, we see that different users have different focuses. The software engineers are more interested in the usability and performance of the apps, while the students pay more attention to the content quality. Also, the students with different background hold diverse opinions. Taking $Q2$ for example, 12 students (10 graduate students and 2 undergraduates) having Semantic Web knowledge approve that CAMO has a better coverage of integrated data than the others. On the contrary, the rest 20 students neglect this advantage more or less.

6.2 Evaluation on Integration Accuracy

We also carry out two experiments to test the integration accuracy of CAMO: one for ontology matching, while the other for instance linkage. In our previous works [11,12,15], we verify the underlying methods of CAMO systematically on a number of widely-used benchmarks from OAEI, and the results demonstrate the effectiveness of these methods. In this evaluation, we particularly focus on assessing them in the multimedia domain. Due to the lack of "golden standard", the generated results are judged manually by two software engineers in Samsung from a practical viewpoint. It is worth mentioning that the evaluation process is time-consuming, error-inevitable and even subjective sometimes. Still, we believe the evaluation is important to make progress in real use.

CAMO discovers 78 mappings between DBpedia and the other ontologies within about 4 minutes, including 18 mappings between DBpedia and the MusicBrainz relational schema. The precision and recall are in Fig. 7(a), where for LinkedMDB and DBTropes, only the classes and properties instantiated in the RDF data are matched because of the unavailability of their ontologies, but this causes the complete semantics loss and narrows the searching space. As compared with the accuracy of BLOOMS [14] on matching DBepdia with the Music Ontology (precision = 0.39, recall = 0.62), CAMO achieves a better accuracy (precision = 0.83, recall = 0.89). Note that this comparison is for reference only, because the results are judged by different people. It is also shown that CAMO finds a bulk of correct mappings between DBpedia and MusicBrainz. We also



(a) Precision & recall of matching ontologies/relational schema with DBpedia

(b) Precision of sampled instance links between DBpedia and others

Fig. 7. Integration accuracy of CAMO

receives feedback from the judges that a small amount of mappings holding the subclass relationship should be involved to support query reformulation.

For instance linkage, CAMO spends nearly a whole day to generate more than 60 thousand links, where a half of them come from the `owl:sameAs` links that already exist in the LOD sources. Notice that the instance linkage phase is done online for the actual system. Since there are too many instance links to evaluate the recall, only the precision is measured on 100 sample links between DBpedia and each LOD source at present. The result is shown in Fig. 7(b). As compared to the precision of the system [10] on linking DBpedia with LinkedMDB (precision = 0.98), CAMO gets a slightly worse result (precision = 0.94). The reason may be that the result in [10] is made by carefully adjusting the threshold, while CAMO has to balance the threshold for the whole multimedia domain. We estimate that the recalls of the two systems are close, because they find a similar number of links. It is also observed that the most discriminative properties for DBpedia are `rdfs:label` and `dbpedia:releaseDate`.

7 Related Work

Roughly speaking, a semantic data integration process consists of three phases: ontology matching, instance linkage and data fusion [3]. A number of works have been proposed to address the issues in each phase [6,5,3], which exploit many kinds of features in ontologies and instances. Recent works also apply machine learning and crowdsourcing to complex data integration tasks [13]. We discuss the semantic data integration works relevant to the multimedia domain.

The survey in [22] investigates the techniques to generate, expose, discover, and consume Linked Data in the context of multimedia metadata, and discusses representative applications and open issues with the goal of bringing the fields of multimedia and Linked Data together. BLOOMS [14] is a system to generate schema-level links between LOD datasets based on the idea of bootstrapping information already in the LOD cloud. It conducts a comprehensive evaluation on many LOD datasets, and our system achieves comparable accuracy in multimedia ontology matching.

To link open multimedia data, the LinkedMDB project [10] is a movie data triplification project and supplies a high quality source of RDF data that links to several well-known LOD sources. The work in [20] introduces an automatic method to interlink music datasets on the Web by taking both the similarity of web resources and of their neighbors into account. Multipedia [8] studies how to enrich ontology instances with candidate images retrieved from search engines. The work in [23] analyzes the relationship between instance linkage and ontology matching and describes a framework for instance linkage taking advantages of ontology matching, which inspires our study. By using DBpedia and LOD, BBC integrates data and links documents to build more meaningful navigation paths across BBC domains, such as BBC Music [16]. Along the same lines as BBC, we develop a set of sophisticated methods to match ontologies, link instances and integrate legacy RDBs in the multimedia domain.

Tabulator [2] and Sig.ma [25] are two representative "desktop" browsers for Linked Data, which provide integrated data views for general users. As mobile devices penetrate everyone's life, more and more mobile apps emerge to exploit Linked Data. DBpedia Mobile [1] is a location-aware client, which supports users to discover, search and publish Linked Data using mobile devices. dbrec [18] is a music recommender system built on top of DBpedia. Additionally, LinkedTV, seevl.fm, wayOU, Who's Who and many others in AI Mashup Challenges give us valuable experiences for developing CAMO.

8 Conclusion and Lessons Learned

In this paper, we describe how LOD is integrated for multimedia metadata enrichment. We develop CAMO, a system that leverages ontology matching and instance linkage techniques for data integration and supports users to browse and search multimedia content on mobile devices. We perform an empirical test to evaluate how CAMO competes with three relevant mobile apps. At the time of writing this paper, we are working on combining the proposed approaches with Samsung Hub to make it better to find and browse multimedia content from a simple, seamless app. During the development and use of CAMO, three specific lessons are learned, and we would like to share them with the community:

Ontology matters. The first lesson learned concerns the importance of ontology. Ontologies stay at the heart of semantic data integration, and in our architecture the global ontology gives a conceptual view over the schematically heterogeneous source schemas. To support high-level query reformula-

tion, a trade-off exists between the ontology's expressiveness and ease of use. Furthermore, the global ontology must cover a wide range of application domains. We use DBpedia in the system, but we want to extend it compatible with existing multimedia metadata models and standards.

Data integration quality. Another lesson learned is about the quality of LOD and the accuracy of integration. The LOD cloud is far from perfect to build applications using it directly. The situation becomes even worse when integrating legacy RDBs. The ambiguous semantics and incorrect/missing data affect the accuracy of integration. Furthermore, all ontology matching and instance linkage techniques have strengths and weaknesses. So, we have to resort to domain experts to establish some links manually. However, human interaction is expensive and often difficult to perform at large scale. Machine learning is a possible way to leverage human computation for improving the accuracy and adaptability of data integration. Additionally, semantic query reformation may require complex mappings, which is not well supported in the current system.

Mobile app design. As is often the case with mobile devices, a limited screen size makes it difficult to efficiently present information and help users view the information. Therefore, a concise and aggregated description of multimedia content is very important. Although we merge the values of matched properties, the method is somehow straightforward without considering inconsistency, and the ranking scheme of properties and values still needs to be studied. Also, user feedback indicates that a user-friendly interface and content are vital to attract users' interests. In the current version of CAMO, we do not expand into NLP, but integrating user-generated content should be considered in the future.

Acknowledgments. This work is supported by the National Natural Science Foundation of China (Nos. 61370019, 61223003 and 61321491), the Natural Science Foundation of Jiangsu Province (No. BK2011189), and the Samsung Electronics. We are greatly thankful to our students and the software engineers in Samsung for participating in the evaluation.

References

1. Becker, C., Bizer, C.: Exploring the Geospatial Semantic Web with DBpedia Mobile. Journal of Web Semantics 7(4), 278–286 (2009)
2. Berners-Lee, T., Hollenbach, J., Lu, K., Presbrey, J., Pru d'ommeaux, E., Schraefel, M.C.: Tabulator Redux: Browsing and Writing Linked Data. In: WWW Workshop on LDOW (2008)
3. Bleiholder, J., Naumann, F.: Data Fusion. ACM Computing Survey 41(1), 1–41 (2008)
4. Buil-Aranda, C., Hogan, A., Umbrich, J., Vandenbussche, P.-Y.: SPARQL Web-Querying Infrastructure: Ready for Action? In: Alani, H., et al. (eds.) ISWC 2013, Part II. LNCS, vol. 8219, pp. 277-293 (2013)

5. Elmagarmid, A.K., Ipeirotis, P.G., Verykios, V.S.: Duplicate Record Detection: A Survey. IEEE Transactions on Knowledge and Data Engineering 19(1), 1–16 (2007)
6. Euzenat, J., Shvaiko, P.: Ontology Matching (2nd Edition). Springer (2013)
7. Fernando, N., Loke, S.W., Rahayu, W.: Mobile Cloud Computing: A Survey. Future Generation Computer Systems 29(1): 84–106 (2013)
8. García-Silva, A., Jakob, M., Mendes, P.N., Bizer, C.: Multipedia: Enriching DBpedia with Multipedia Information. In: K-CAP 2011, pp. 137–144 (2011)
9. Giunchiglia, F., Shvaiko, P., Yatskevich, M.: S-Match: An Algorithm and an Implementation of Semantic Matching. In: Davies, J., et al. (eds.) ESWS 2004. LNCS, vol. 3053, pp. 61–75 (2004)
10. Hassanzadeh, O., Consens, M.: Linked Movie Data Base. In: WWW Workshop on LDOW (2009)
11. Hu, W., Chen, J., Qu, Y.: A Self-Training Approach for Resolving Object Coreference on the Semantic Web. In: WWW 2011, pp. 87–96 (2011)
12. Hu, W., Qu, Y., Cheng, G.: Matching Large Ontologies: A Divide-and-Conquer Approach. Data and Knowledge Engineering 67(1), 140–160 (2008)
13. Isele, R., Bizer, C.: Active Learning of Expressive Linkage Rules Using Genetic Programming. Journal of Web Semantics 23, 2–15 (2013)
14. Jain, P., Hitzler, P., Sheth, A.P., Verma, K., Yeh, P.Z.: Ontology Alignment for Linked Open Data. In: Patel-Schneider, P.F., et al. (eds.) ISWC 2010, Part I. LNCS, vol. 6496, pp. 402–417 (2010)
15. Jia, C., Hu, W., Bai, W., Qu, Y.: SMap: Semantically Mapping Relational Database Schemas to OWL Ontologies. Journal of Computer Research and Development 49(10), 2241–2250 (2012)
16. Kobilarov, G., Scott, T., Raimond, Y., Oliver, S., Sizemore, C., Smethurst, M., Bizer, C., Lee, R.: Media Meets Semantic Web – How the BBC Uses DBpedia and Linked Data to Make Connections. In: Aroyo, L., et al. (eds.) ESWC 2009. LNCS, vol. 5554, pp. 723–737 (2009)
17. Lehmann, J., Isele, R., Jakob, M., Jentzsch, A., Kontokostas, D., Mendes, P.N., Hellmann, S., Morsey, M., van Kleef, P., Auer, S., Bizer, C.: DBpedia – A Large-scale, Multilingual Knowledge Based Extracted from Wikipedia. Semantic Web Journal (2014)
18. Passant, A.: dbrec – Music Recommendations Using DBpedia. In: Patel-Schneider, P.F., et al. (eds.) ISWC 2010, Part II. LNCS, vol. 6497, pp. 209–224 (2010)
19. Raimond, Y., Abdallah, S., Sandler, M., Giasson, F.: The Music Ontology. In: International Conference on Music Information Retrieval, pp. 417–422 (2007)
20. Raimond, Y., Sutton, C., Sandler, M.: Automatic Interlinking of Music Datasets on the Semantic Web. In: WWW Workshop on LDOW (2008)
21. Saathoff, C., Scherp, A.: M3O: The Multimedia Metadata Ontology. In: SAMT Workshop on SeMuDaTe (2009)
22. Schandl, B., Haslhofer, B., Bürger, T., Langegger, A., Halb, W.: Linked Data and Multimedia: The State of Affairs. Multimedia Tools and Applications 59(2), 523–556 (2012)
23. Scharffe, F., Euzenat, J.: Linked Data Meets Ontology Matching – Enhancing Data Linking through Ontology Alignments. In: International Conference on Knowledge Engineering and Ontology Development, pp. 279–284 (2011)
24. Stoilos, G., Stamou, G., Kollias, S.: A String Metric for Ontology Alignment. In: Gil, Y., et al. (eds.) ISWC 2005. LNCS, vol. 3729, pp. 623–637 (2005)
25. Tummarello, G., Cyganiak, R., Catasta, M., Danielczyk, S., Delbru, R., Decker, S.: Sig.ma: Live Views on the Web of Data. Journal of Web Semantics 8(4), 355–364 (2010)