Content-Aware Master Data Management

Karin Murthy[1], Deepak P.[1], Prasad M. Deshpande[1], Sreekanth L. Kakaraparthy[2], Vedula T. Surya Sandeep[2], Vijaya K. Shyamsundar[2], Sanjay K. Singh[2]

[1]IBM Research - India
Bangalore, India

[2]IBM Software Group
Bangalore, India

{*karinmur*|*deepak.s.p*|*sreekanthkl*|*prasdesh*|*sksingh6*|*vesandee*|*vjkumar*}@*in.ibm.com*

Abstract

Master data management (MDM) provides a means to link data from various structured data sources and to generate a consolidated *master record* for entities such as customers or products. However, a large amount of valuable information about entities exists as unstructured content in documents. In this paper, we show how MDM can be made aware of information from unstructured content by automatically extracting valuable information from documents. We demonstrate for an example application that it is possible to make MDM content-aware without compromising MDM's premise to be the one trusted source for all entity-related information.

1 Introduction

Master data management (MDM) provides a means to link data from various structured data sources and to generate one master record for each entity. For example, MDM delivers an integrated view of key information about entities such as customers, products, and suppliers.

Currently, MDM is limited to integrating data from structured data sources. However, according to various sources [1] as much as 80% of the world's information is unstructured. For example, a large amount of valuable, unstructured information is stored in the form of documents inside Enterprise Content Management (ECM) systems. IBM's InfoSphere Master Content Bridge [2] bridges the gap between MDM and ECM and allows enterprises to link documents with existing master data records. A document is linked to a master record by examing the document's metadata and using key metatdata attributes to identify a match between a document and an MDM record.

However, documents are still treated as opaque objects and the information contained in them is untapped. Using information extraction techniques, valuable information can automatically be extracted from unstructured content. The extracted information benefits MDM in several ways. First, extracted information enables more accurate linking of content to other master data for an entity. Second, the extracted information can be provided to MDM as additional attributes, making each master data record comprehensive. Finally, with more available information, duplicate record detection techniques [5] used by MDM to detect multiple records that map to the same entity, fare significantly better.

To ensure that integrating information from unstructured sources does not compromise MDM's premise to be the one trusted source for all entity-related information, information extraction techniques need to be geared towards delivering trusted and reliable information. In this paper, we show for an example application that making MDM content-aware is feasible.

The remainder of the paper is organized as follows. We first describe an example application for content-aware MDM in Section 2. Section 3 highlights four scenarios of content-aware MDM and Section 4 describes the various components involved. We show in Section 5 that it is possible to extract high quality information and conclude the paper in Section 6.

2 An Example Application

Staffing departments and recruitment agencies deal with large amounts of information in the form of documents. Typically an applicant will supply documents such as CV, cover letter, letters of reference, and copies of transcripts. A large amount of the information contained in those documents is master data for the entity *Applicant*. For example, the applicant's name, phone number, address, birth data, education, and employment history should all be part of the applicant's master record.

In the context of application processing, there are several benefits to apply content-aware master data management which automatically taps into the content of documents.

International Conference on Management of Data
COMAD 2010, Nagpur, India, December 8–10, 2010

First, when uploading a document to an ECM system, the document has to be tagged with metadata such as *name, phone number*, and *email address* in order to retrieve the document at a later point in time. Automatic extraction of information from each document reduces time-consuming and error-prone manual tagging.

Second, many companies do not have a single repository for storing all recruitment-related documents and information. Without MDM providing a single consolidated master record for each applicant, it is easily possible that the same applicant is unnecessarily processed by multiple recruiters. Even with a single repository there is a need to identify duplicate entries. For example, a candidate may apply for different job offerings and send documents multiple times.

Lastly, manually processing large amounts of resumes to fill a position can be very time-consuming and inefficient. Content-aware master data can be used to automatically group and select applicants based on user-specified values (for example, *Years of experience, Highest qualification*, and specific skills) and thus help the recruitment team to preselect promising candidates from the flood of applications.

Information extraction used to automatically provide additional information about the MDM entity *Applicant*, must deliver trustworthy information. For example, recruiters do not want to miss out on a great candidate by filtering out a resume based on wrongly extracted information. We show in Section 5 that information extraction can meet the stringent quality requirements of an MDM system for applicant processing.

3 Scenarios of Content-Aware MDM

We highlight four scenarios of content-aware MDM.

3.1 Scenario 1: Validation of metadata

The process of manually adding metadata for each document is error prone. Additional information extracted from each document can be used to validate metadata. Discovered inconsistencies between metadata and extracted data can be used to improve the quality of metadata in the ECM and as such the quality of information propagated to MDM. Figure 1(a) shows an example, where the metadata and the content of the document do not match. In this example, the recruiter accidentally associated metadata from a different applicant when uploading document *doc*3.

3.2 Scenario 2: Detecting master content

Multiple versions of the same document may be uploaded to a single ECM system or multiple version of a document may be uploaded to different ECM systems. To ensure the quality of master data, only the information from the most up-to-date version of the

Entity	Document		Metadata		Extracted Data			
ID	ID	Type	First Name	Last Name	First Name	Last Name	Student ID	Email
e1	doc1	CV	Ben	Doe	Ben	Doe		b.Doe@gmail.com
e1	doc2	Application	Ben	Doe	Ben	Doe	12345	b.Doe@gmail.com
e1	doc3	Application	Ben	Doe	Tom	Smith	9999	tom@yahoo.com

(a) Scenario 1: Validation of metadata

Entity	Document		Metadata		Extracted Data			
ID	ID	Type	First Name	Last Name	First Name	Last Name	Student ID	Email
e3	doc5	CV	Tom	Smith	Tom	Smith		
e3	doc6	CV	Tom	Smith	Tom	Smith		tom@yahoo.com

(b) Scenario 2: Detecting master content

Entity	Document		Metadata		Extracted Data			
ID	ID	Type	First Name	Last Name	First Name	Last Name	Student ID	Email
e1	doc2	Application	Ben	Doe	Ben	Doe	12345	b.Doe@gmail.com
e2	doc4	CV	Benny	Doe	Benny	Doe	12345	b.Doe@gmail.com

(c) Scenario 3: Enhanced duplicate detection

Entity	Document		Metadata	Extracted Data			
ID	ID	Type		First Name	Last Name	Student ID	Email
e1	doc1	CV		Ben	Doe		b.Doe@gmail.com
e1	doc2	Application		Ben	Doe	12345	b.Doe@gmail.com
e1	doc4	CV		Benny	Doe	12345	b.Doe@gmail.com
e3	doc3	Application		Tom	Smith	9999	tom@yahoo.com
e3	doc6	CV		Tom	Smith		tom@yahoo.com

(d) Scenario 4: MDM without metadata

Figure 1: Four scenarios of content-aware MDM

document (that is, for the *master document*) should be part of a master data record. Data extracted from each document can be used to detect the master copy (often the most recent copy) of a document. In Figure 1(b) *doc6* is probably an updated version of the CV contained in *doc5*. Policies specialized for different document types may guide the process of picking the master version of a document.

3.3 Scenario 3: Enhanced duplicate detection

A key feature of MDM is the detection of suspect duplicate entities. Additional information extracted from the content provides a rich source for detecting duplicate entities. Figure 1(c) shows an example where two entities $e1$ and $e2$ actually refer to the same applicant. Even though not all metadata attributes match (specifically, the first name of $e1$ does not match the first name of $e2$), all other additionally extracted information matches. Thus, $e1$ and $e2$ are likely to refer to the same applicant and should be merged. Even though we do not show master data in this example, the actual suspect duplicate processing is done by the MDM system after information from each document has been added to master data.

3.4 Scenario 4: MDM without metadata

Small organizations with no content management system often store documents in a file system without any

metadata. In this case, extracted data can be used as metadata to upload all documents into an ECM system. Alternatively, documents in a file system can also be directly linked to MDM records by extracting appropriate information from each document. Even if no MDM solution is used, the extracted information can be used to group documents by entities. Figure 1(d) shows an example of grouping documents into entities purely based on extracted information.

4 Components

In this section, we briefly describe the components of content-aware MDM.

4.1 Overview of Components

The following five components are necessary to realize content-aware MDM:

MDM: An MDM system to manage master data. We use IBM's InfoSphere Master Data Management Server.[1]

ECM: An ECM system to store documents. We use IBM's FileNet Content Manager.[2]

IE: An information extraction module. We use SystemT[3] with some extensions to enable high-precision information extraction.

Metadata Validator (MV): A new component that supports Scenario 1 in Section 3 by validating whether extracted information matches available metadata.

Master Content Updater (MCU): A new component that supports Scenarios 2, 3 and 4 in Section 3 by updating MDM with additional information available due to the upload of a document in ECM.

In the following, we describe IE, MV and MCU in more detail.

4.2 Metadata Validator (MV)

The MV component is entrusted with the responsibility of checking any inconsistencies between the extracted information and user-entered metadata. An inconsistency can be a direct attribute mismatch (for example, the metadata entered is *Name: Benny* and the extracted information is *Name: Charles*) or a violation of domain-specific constraints (for example, the meta data entered is *Birth Date: 10.01.1970* and the extracted information is *Years of Experience: 30*). All discovered inconsistencies are sent to the user for validation. The user changes metadata and/or extracted information accordingly and if necessary provides feedback to the IE component about incorrect annotations. Figure 2 depicts the various steps in the process by arrows, with time advancing in downward direction.

[1] www-01.ibm.com/software/data/infosphere/mdm_server

[2] www-01.ibm.com/software/data/content-management/filenet-content-manager

[3] www.alphaworks.ibm.com/tech/systemt

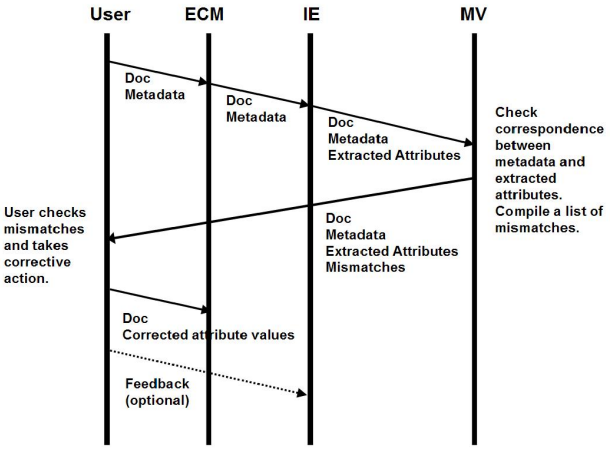

Figure 2: Metadata Validation

Although not depicted here, the user intervention is skipped if no inconsistencies are found.

4.3 Master Content Updater (MCU)

The MCU component keeps MDM updated with the latest information in ECM. For example, if a new resume for an applicant containing a new qualification is uploaded to ECM, the attribute *Highest Qualification* in the respective MDM record needs to be updated. MCU is aware of the primary key attribute(s) (for example, the email address of an applicant) by which MDM records can be retrieved. When a document is added to ECM and the set of available information after the validation process is a proper superset of the primary key, MCU is triggered. MCU then retrieves the complete MDM record, and checks if there is any need for updating the record. If so, updates are made and sent back to MDM. In case, no MDM record is returned for a primary key, MCU triggers the creation of a new MDM record. The overall process is illustrated in Figure 3. The broken arrow denotes that when the user uploads a document to ECM, it first passes through MV before it reaches MCU. The last step in the diagram is skipped if no update to the MDM record is necessary.

4.4 Information Extraction (IE)

The IE component is responsible for extracting relevant information from unstructured documents. The IE component is based on SystemT [3] which includes AQL, a declarative rule language to write annotators to extract information from text. Rule-based annotators are known to achieve high accuracies [4]. In addition to building domain-specific annotators (as described in Section 5), we use two other techniques to further enhance the trustworthiness of extracted information.

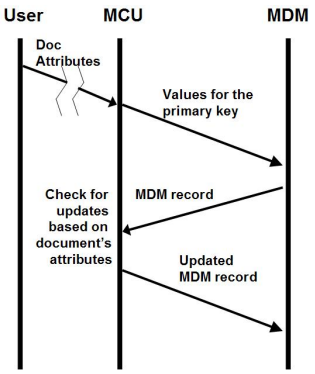

Figure 3: Master Content Updater

Person-name assumptions	Counter example
First letter of words capitalized	Chandra kanth Gurram
Full name has at least two words	MADHUSUDHANA
Dots appear after initials	Rajesh. V

Table 1: Some invalidated person-name assumptions

Firstly, we use existing metadata or already extracted data to guide the extraction process in discovering more trustworthy information. Secondly, we explore minimally querying the user for feedback on low-confidence extractions to improve and maintain extraction accuracy even in the presence of changing trends.

5 Evaluation

Based on input from IBM Staffing, we developed rule-based annotators [3] for the following attributes: *Name, Email, Phone Number, Birth date, Highest qualification, Year of highest qualification, Current employer* and *Total years of experience*. To build and test the annotators, we received a sample of 38 Indian resumes from IBM Staffing and retrieved another 12 Indian resumes from the web. Half of the resumes were used for building and improving the annotators, the other half was used for test purposes. We manually labeled all documents and report precision and recall numbers against the manually labeled data.

We observed that the generic name annotator commonly used for western names does not work well for the names in Indian resumes. Table 1 shows some examples of assumptions that do not hold. To compensate, we incorporate metadata into the extraction process. For example, we use the metadata attribute *Email address* and the fact that the first part of an email is often composed from parts of a person's name to guide the extraction process.

We also observed that the generic annotator for organizations does not work well for Indian organizations. Given the large number of organizations and spelling variations, it is not feasible to maintain a complete dictionary of all organizations. Instead we add additional domain-specific rules to recognize organizations in a resume. For example, company names in resumes are often preceded by words such as *for, at, in, with*.

Annotator	Precision	Recall
Person Name (generic)	33	32
Person Name (using metadata)	92	48
Phone Number (generic)	100	80
Phone Number (domain-specific)	100	92
Email (generic)	100	100
Date of Birth	100	92
Highest Qualification	96	96
Year of Qualification	100	96
Current Employer (generic Org annotator)	91	76
Current Employer (domain-specific Org annotator)	100	88
Years of Experience	95	80

Table 2: Precision and recall for various annotators

Table 2 summarizes the precision and recall we achieve for the various attributes implementing different rule-based annotators.

6 Conclusion

We have demonstrated for an example application that it is possible to harness content for master data management. Specifically, we showed that it is possible to extract reliable structured information from content. Once the information is extracted reliably, it can be used to link with other master data for an entity, to detect master content, to enhance detection of duplicate entities, and to validate metadata associated with documents. With these techniques the concept of content-aware MDM that combines both structured and unstructured data while maintaining the trustworthiness of the data becomes a reality.

References

[1] `clarabridge.com/default.aspx?tabid=137&ModuleID=635&ArticleID=551`.

[2] `www-01.ibm.com/software/data/infosphere/mdm_server/master-content.html`.

[3] L. Chiticariu, R. Krishnamurthy, Y. Li, S. Raghavan, F. Reiss, and S. Vaithyanathan. Systemt: An algebraic approach to declarative information extraction. In *ACL*, 2010.

[4] L. Chiticariu, R. Krishnamurthy, Y. Li, F. Reiss, and S. Vaithyanathan. Domain adaptation of rule-based annotators for named-entity recognition tasks. In *EMNLP*, 2010.

[5] A. K. Elmagarmid, P. G. Ipeirotis, Vassilios, and S. Verykios. Duplicate record detection: A survey. *Transactions on Knowledge and Data Engineering*, 2007.