基于语义的数据格式转换

Data Transformation with Semantics

郝亚南　陈少飞　李天柱　袁　方　（保定河北大学数学与计算机学院　071002）

摘要：本文提出了一种基于语义的数据格式转换方法。该方法以 Word 文档为例，采用基于学习的策略，自动地将 Word 文档转换为具有语义信息的 XML 文档，便于用户对大量 Word 文档进行精确的基于语义的查询和管理；该方法支持所见即所得，易于使用。

关键词：格式转换　Word　XML　语义

1　引言

本文在分析现有的各种数据格式转换技术的基础上，提出了基于语义的数据格式转换方法。该方法以 Word 文档为例，利用我们自主开发的信息抽取引擎 PQAgent[2]，采用基于学习的策略，自动地将 Word 文档转换为具有语义的 XML。同时，为了进行高效的查询，又将转换结果转换为关系型数据作为副本。Word 文档只有编辑和显示信息，无语义信息；将之转换为 XML 后，生成具有语义信息的文档数据，从而完成了基于语义的数据格式转换。

2　相关技术

2.1　Word 对象模型

对象是 Microsoft Word 的基本构成单元，用户在 Word 中操作和改变的每一个东西（如文本、图形等）都是一个对象，这些对象的相互关系组成了 Word 中的对象模型[3]。同时，这些对象都有自己的属性和方法，用户可通过 VBA 编程来访问这些已有对象，改变它们的属性及内容，以完成某些较高级的功能。VBA 的执行在 Office 环境中进行。

2.2　PQAgent 系统介绍

我们自主开发了信息抽取原型系统 PQAgent，它具有较好的适应性和较高的性能。它将信息抽取的过程分为四个阶段：附加语义，样本学习，规则优化和信息抽取。

(1) 附加语义。用户根据自己对网页内容的理解，通过创建语义模式，将反映网页内容的语义信息记录下来，作为样本学习阶段的输入。

(2) 样本学习。用户按照语义模式的层次结构，以自顶至底的顺序，依次标记出网页中的数据内容，并选择相应的语义模式，在语义模式的语义项与 HTML 网页中的信息块之间建立对应关系，形成"抽取规则"。

(3) 规则优化。在样本学习阶段完成后，系统对生成的规则进行优化。最终的抽取规则被装配成完整的 XQuery[4] 查询语句，构成复杂对象的抽取规则，存放于和语义模式对应的规则库中。

(4) 信息抽取。采用的方法是首先将 Web 页面的数据转换为 XML 格式。然后，根据该网页对应的语义模式，系统自动从规则库中取出相应的 XQuery 查询语句，输入到 XQuery 查询引擎，对待抽取的网页进行查询，然后将查询结果合并到一个 XML 文档作为抽取结果，放入和该语义模式对应的 XML 文档库中。

3　格式转换原理和方法

3.1　体系结构

我们的格式转换方法基于 B/S 结构，其体系结构如图 1 所示。浏览器端和服务器端的组成分别如下：

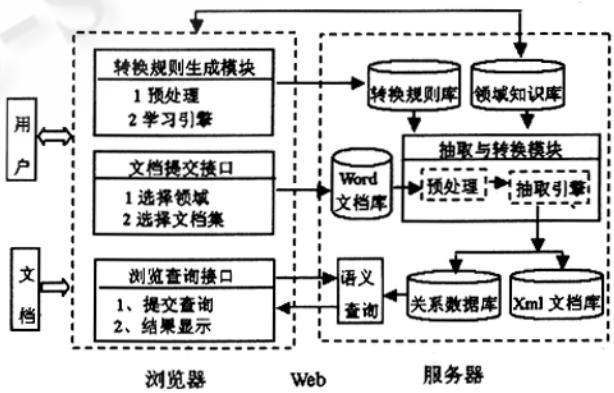

图 1　体系结构

(1) 浏览器端组成。包括转换规则生成模块，文档提交接口和基于语义的查询接口。

转换规则生成模块作为 Word 插件，嵌入到 Word 环境中。用户在浏览器中打开 Word 时，转换规则生成模块便成

为 Word 的一部分。用户在 Word 环境中对样本 Word 文档进行学习,规则生成模块记录用户的操作,自动生成转换规则,并提交到服务器端的转换规则库中。

文档提交接口让用户选择待转换的 Word 文档集合并提交给服务器;基于语义的查询接口供用户输入查询条件,并将查询结果以 HTML 或 Word 文档格式在浏览器中呈现给用户。

(2) 服务器端组成。包括 Word 文档抽取与转换模块,语义查询模块;此外,为了便于转换工作的进行,我们在服务器端构造了五个库,即领域知识库,Word 文档库,XML 文档库,关系数据库和转换规则库。

抽取与转换模块负责对用户提交到服务器端的 Word 文档完成基于语义的转换。文档转换模块将 Word 文档转换为 XML 格式,并存于 XML 文档库中;同时,另外生成一份关系数据作为副本,存放到关系数据库中。原来的 Word 文档存放于 Word 文档库中。

领域知识库将文档信息划分成不同领域,便于对不同领域、不同语义的 Word 文档进行管理,便于用户进行数据查询。我们还利用领域知识库对 Word 文档库,XML 文档库,转换规则库进行组织与管理。

Word 文档库存放的是用户提交的原始的 Word 文档;XML 文档库存放的是用户提交的 Word 文档转换后的 XML 结果;关系数据库存放的是 XML 文档库的副本;转换规则库用于存放利用样本学习产生的转换规则,用于转换 Word 文档。这些库中的数据都根据领域知识库中的嵌套关系,按领域分类,组织成树状结构按层次加以管理。

3.2 语义模型

Word 文档本身只有显示和编辑信息,没有语义信息,为了实现语义化的处理,需要为 Word 文档数据定义合适的语义模型。通过对大量的 Word 文档样本进行分析,我们发现,现有的具有较规范文档格式与显示风格的 Word 文档,如电子公文中的"红头文件"等,其数据绝大部分呈现出平面化的特点。所以,我们提出的格式转换方法主要处理平面化的 Word 文档数据。为此,我们选用"平面的关系数据模型"作为我们的语义模型,它用 XML 语法表达,但兼容平面化的关系数据模型。

3.3 工作原理

我们的 PQAgent 系统主要分为学习引擎和抽取引擎两大部分。

本小节,我们将阐述如何将 PQAgent 应用到格式转换领域中,用它处理一般意义的 Word 文档,从 Word 文档中抽取数据,实现基于语义的格式转换。

3.3.1 Word 文档预处理

文档预处理工作在利用 PQAgent 学习引擎学习之前和利用 PQAgent 抽取引擎转换之前都要进行,这都是由系统的预处理模块自动完成。由于我们的信息抽取引擎 PQAgent 是基于 DOM 模型的,它处理的是满足 DOM 规范以及结构良好(Well-formed)的 XML 文档(XHTML Web 页面)。所以如果进行 Word 文档的抽取与转换,必须将它转换为 XML 文档,使之满足 DOM 规范。Word 自身提供了将 Word 文档转换为 HTML 页面的功能,但是,这种转换不仅没有语义信息,而且,转换后的 HTML 文档采用了大量 Word 自定义的命名空间,格式、符号和指令,十分复杂多变,不满足结构良好的要求。我们这一步的主要工作就是对 Word 生成的 HTML 文档规范化,使之满足 XML DOM 规范,进一步能被 PQAgent 处理,实现 Word 文档的转换。

用于 HTML 文档规范化的一个常用工具是 Tidy[5]。Tidy 处理一般性的 HTML 文档效果很好。但是,Word 生成的 HTML 文档十分特殊,其中包含了大量的自定义的命名空间和样式,Tidy 在处理时会把这些内容都去掉。其中的显示风格是我们建立映射关系、生成转换规则的重要组成部分,将之去掉,我们就无法生成准确的转换规则,无法完成 Word 文档的转换。所以,我们只能自己对 Word 生成的 HTML 文档进行预处理。

预处理工作主要集中在如下三点:

(1) 对标签中属性的规范化处理。例如,对未有引号的属性值加上引号使之完整。

(2) 对标签的规范化处理。例如,对标签加以添加及去除操作,使标签必须成对出现,对称且必须正确嵌套。

(3) Word 自定义指令的删除。Word 转换后的 HTML 文档存在大量的特殊的、特定于 Word 文档的指令,不符合标准的 XML 语法,我们将之去除。

将 Word 生成的 HTML 文档经过预处理后,转换后的 XML 文档符合 DOM 规范,结构良好(well-formed),可以进一步输入 PQAgent 进行后续操作与处理。

3.3.2 附加语义

用户需根据样本文档的数据内容,自定义语义模式,来完成附加语义的过程。如图 2 所示,这是一个电子公文中的"红头文件"格式的样本文档。在图中所示的样本"红头文件"中,[标题]、[序号]、[主题词]等信息较为重要,用户较为关心,所以,用户需定义语义模式来反映该类"红头文件"中的密级,保管时间、序号、标题等信息的语义特征。

为完成模式定义,用户需首先打开样本 Word 文档。如图 2 所示,我们采用在 Word 环境中添加输入界面,提供树型输入接口(如图 2 中的模式定义窗口)的方式,让用户在树型视图中根据样本文档的内容,自己定义语义模式。

图 2　应用环境

语义模式的 DTD 树表示如图 3 所示。语义模式对应的 DTD 如下：

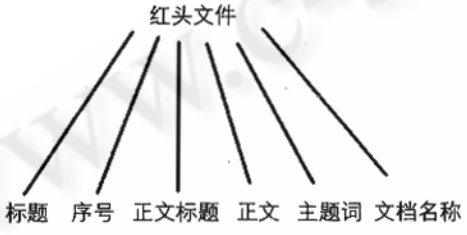

图 3　模式定义

```
<! DOCTYPE 红头文件[
<! ELEMENT 红头文件(标题,序号,正文标题,正文,主题词,文档名称)>
    <! ELEMENT 标题(#PCDATA)>
    <! ELEMENT 序号(#PCDATA)>
    <! ELEMENT 正文标题(#PCDATA)>
    <! ELEMENT 正文(#PCDATA)>
    <! ELEMENT 主题词(#PCDATA)>
    <! ELEMENT 文档名称(#PCDATA)> ]>
```

3.3.3 样本学习

样本学习的主要目的是在创建的语义模式中的语义项与电子公文 Word 文档中的数据项之间建立映射关系，以便完成数据转换。用户学习时，先在以树型视图界面表示的语义模式中选择语义项(例如上面定义的语义模式中的"序号")，然后在样本 Word 文档中标记与语义项对应的数据项(如图 2 样本文档中的文本"×字 001"已被标记)。另外，用户在标记样本 Word 文档时，还可以指明所标记的数据项的语义特征，比如，固定文本，是否是数字，是否是日期等等，这些都可以作为辅助的启发信息，形成辅助规则，进一步保证了转换的准确性。用户标记完成后，系统利用 VBA 访问 Word 文本对象，记录用户在样本 Word 文档中标记的数据内容，并将其提交给 PQAgent 学习引擎。PQAgent 学习引擎首先调用预处理模块对样本 Word 文档进行预处理，经过规则淘汰、合并、优化等步骤，最终生成 XQuery 语句，作为整个 Word 文档语义数据的抽取与转换规则。

学习完成后，系统将转换规则提交到服务器端的转换规则库中，供下一步转换使用。

3.3.4 格式转换

格式转换过程完全由系统自动完成。用户在完成样本 Word 文档的学习之后，系统自动生成转换规则，并存放到服务器端的转换规则库中。

此后，用户从客户端提交 Word 文档到服务器，文档转换模块将在样本学习阶段由 PQAgent 学习引擎产生的转换规则输入给 PQAgent 抽取引擎，抽取引擎调用 XQuery 引擎，从用户指定的 Word 文档中把语义数据抽取出来并转换为符合预定义语义模式 XML 文档。同时，系统将转换后的 XML 保存一份副本存放到关系数据库中。

需要指出的是，采用基于学习的方法转换 Word 文档是有前提条件的，我们要求用户按照标准样式编辑 Word 文档。典型的应用领域，如电子政务、学术论文等，都有其格式规范(包括显示规范和语义规范)，在这样的特殊环境下，用户按照规范的格式来编辑文档，定义语义模式，只需经过一次学习，转换的准确率就可以达到 100%。

Word 文档的格式转换完成后。转换后的 XML 文档可用于数据交换领域；关系数据库可用于高效的查询处理。

3.3.5 运行模式

我们的格式转换方法既可以转换自己编辑生成的文档，又可以转换现有的遗留文档。其运行模式如下：

(1) 编辑文档：用户按照给定的样本文档，按照格式规范来编辑 Word 文档，其显示风格与结构应该同样本文档基本一致。

(2) 样本学习：用户必须首先进行模式定义及样本学习。首先在 Web 页面中选择"学习"功能，系统自动在浏览器中打开 Word 环境。用户可以打开任意一个 Word 文档作为样本。用户选择"模式定义"功能后，系统以树型结构作为输入界面让用户完成语义模式的定义。接着进行样本学习。用户依次标记 Word 文档中的文本内容，并选中语义模式中相应的语义项，完成学习过程。学习完毕后，用户选择"生成规则"功能，系统调用规则生成模块中的 PQAgent 学习引擎，生成转换规则，并按用户选择的领域，将规则存放到

服务器端的转换规则库中。

(3) 存储与转换:用户编辑 Word 文档完成后,进入文档提交页面,选择相应的领域,提交文档到服务器端的 Word 文档库中。完成提交后,系统首先将 Word 文档存为 HTML 格式,并调用 Word 文档预处理模块进行预处理过程,然后,调用转换模块中的 PQAgent 抽取引擎完成转换。转换后的 XML 文档存放于 XML 文档库中,其副本存放到关系数据库中。

(4) 语义查询:用户首先在 HTML 页面中选择要查询的领域,例如,选择"电子公文"领域,表示欲在电子公文领域中查询"红头文件"的文档数据。接着,再输入查询条件,比如查询主题词包含"发展"的红头文件。最后提交,查询条件被上传到服务器。系统自动将查询结果以 HTML 页面或 Word 文档格式返回到客户端,而且是具有语义信息的数据。

(5) 重新显示与编辑:在语义查询的同时,用户也可以浏览原来的整个 Word 文档。Word 文档被系统从服务器端的 Word 文档库中提取,并下载到客户端的浏览器中,Word 环境在浏览器中打开,用户可以查看初始的 Word 文档样式。同时,用户也可以重新对 Word 文档进行编辑,编辑完成后,可以再次提交到服务器,系统自动重新完成转换,并更新 XML 文档库和关系数据库中相应的数据内容。

4 实验环境及测试

根据上文阐述的基于语义的格式转换原理,利用我们已有的自主开发的信息抽取引擎 PQAgent,我们构建了基于 Web 的 Word 文档转换实验环境。整个实验环境采用基于 Browser/Server(浏览器/服务器)的 B/S 结构。实验对象为电子政务领域中的电子公文。编程环境为 Visual Basic Application (VBA),Visual InterDev 6.0 和 IIS 服务器。

我们选择了 7 种电子公文"红头文件"格式作为样本,在实验环境中进行了性能测试。这 7 种电子公文"红头文件"是 7 种不同的电子公文规范格式。我们根据这 7 种规范格式,分别编辑 7 个 Word 文档,定义了 7 种语义模式。测试结果显示:对于这 7 种文档格式,转换的准确率均为 100%。

我们的格式转换方法主要是利用已有的信息抽取引擎 PQAgent。转换结果的准确率取决于 PQAgent 的抽取能力。PQAgent 系统既能处理复杂的嵌套对象,又能处理简单的平面对象,处理平面对象的能力尤其强,对于相对较为规则的 Web 页面,抽取的准确率可以达到 100%。典型的应用领域,如电子政务、学术论文等,都有其格式规范(包括显示规范和语义规范),且数据绝大多数都呈现出平面化的特点,在这样的特殊环境下,用户按照规范的格式来编辑文档,定义语义模式,只需经过一次学习,转换的准确率就可以达到 100%。

5 小结

本文提出了基于语义的格式转换方法,以 Word 文档为例,将 Word 文档附加语义信息,转换成 XML 格式。该方法对 Word 文档的转换采用了学习的策略:不仅能转换按规定格式编辑的文档,还能转换大量遗留文档,均为自动完成,且可用于 PDF 等其他数据格式的转换。该方法支持所见即所得,应用环境没有特殊性,易于被用户使用;转换后的 XML 文档具有语义信息,既可以进行精确的基于语义的查询,又可以对大批量的 Word 文档进行基于语义的查询和管理,大大提高工作效率。进一步的工作主要有:用户对 XML 文档库或关系数据库中的数据进行更新后,如何反映到原始的文档中,如何自动完成对原始文档的更新;如何转换结构复杂的 Word 文档;如何减少用户的工作量,提高系统的智能性;电子文档的安全性问题;如何将该种格式转换技术应用于其他格式的文档如 PDF、PS 等格式,这些问题需要进一步研究。

参考文献

1 Word and XML [EB/OL]. http://www.gca.org

2 陈少飞:Web 信息抽取规则的优化及规则的 XQuery 表达,河北大学 2000 级硕士学位论文。

3 Office SDK. MSDN, 2001 年 4 月。

4 XML Query. http://www.w3.org/XML/Query

5 D.Raggett. Clean Up Your Web Pages with HTML TIDY. http://www.w3.org/People/Raggett/tidy/1999.

6 ROBERT BAUMGARTNER, SERGIO FLESCA, GEORG GOTTLOB. Supervised wrapper generation with Lixto [Z]. In Proceedings of the 27th International Conference on Very Large Database. Roma, Italy, 2001.