Maschinelles Lernen von Beziehungen in Metadaten-Repositories

- Wissensmanagement mit automatischer Klassifikation -

U. Furbach, Universität Koblenz-Landau[1],
M. Groß-Hardt, wizAI[2]
M. Herr, Deutsche Post World Net[3]
B. Thomas, wizAI[4]

Abstract. *Ein Metadaten-Repository verwaltet Metadaten über Systeme, Datenbanken und ihren Daten. Es spielt die Rolle eines „Information Brokers“ und versorgt Anwendungen sowie Benutzer mit Informationen über Datenquellen und ihren Abhängigkeiten. Aufbau und Pflege eines solchen Repositories sind mit hohem Zeit- und Personalaufwand verbunden. In diesem Artikel wird die Problemstellung konkretisiert und es wird anhand von zwei Anwendungsszenarien dargestellt, wie durch Einsatz von automatischen Klassifikationstechniken der Aufwand für die Pflege reduziert werden kann.*

1. Einleitung

In einem großen Unternehmen werden riesige Datenmengen verwaltet. Daten über Kundenstamm, Lieferanten sowie Produkte werden im Allgemeinen in einer Vielzahl von Systemen und Datenbanken abgelegt. Die Anwendungen hierauf wurden häufig unabhängig voneinander entwickelt, so daß die Daten eine heterogene Struktur aufweisen in bezug auf Attributnamen, Datentypen, Semantik etc. Das Erkennen ähnlicher Objekte sowie von Beziehungen zwischen Objekten ist eine zentrale Aufgabe für eine unternehmensweite Anwendungsintegration [Bou98, She90].

Im Rahmen einer unternehmensweiten Wissensmanagement-Strategie ist ein zentrales Repository [Mar00] über die verteilt vorliegenden Systeme

[1] Universität Koblenz-Landau; 56070 Koblenz; E-Mail: uli@uni-koblenz.de
[2] WizAI; Maria Trost 23; 56070 Koblenz; E-Mail: m.gross-hardt@wizai.com
[3] Charles-de-Gaulle-Straße 20; 53113 Bonn; E-Mail: m.herr2@deutschepost.de
[4] WizAI; Maria Trost 23; 56070 Koblenz; E-Mail: b.thomas@wizai.com

und Anwendungen sowie ihrer Daten notwendig, um Änderungen kontrollieren und neuen Anforderungen zeitnah gerecht werden zu können. Ein Metadaten- Repository, als zentraler „Information Broker“, wird deshalb eingesetzt, um Daten über vorhandene Applikationen und den in ihnen gespeicherten Daten zu erfassen. Es dient als „Single Point of Truth“ für Benutzer und Anwendungsentwickler. Ein Metadaten-Repository beinhaltet Wissen über Objekte, (Datenbank-) Tabellen und Beziehungen zwischen den verschiedenen Datenbanken und Anwendungen. Die Qualität der Daten in einem Repository ist ausschlaggebend für seinen Erfolg.

Ein Metadaten-Repository muß somit kontinuierlich gepflegt werden, um die Korrektheit und Vollständigkeit der Daten zu gewährleisten. Das Erfassen und Pflegen von Metadaten erfordert einen hohen zeitlichen und personellen Aufwand. Wenn neue Objekte in das Repository eingepflegt werden, benötigt der Repository-Benutzer Wissen über die bereits vorhandenen Objekte im Repository, sowie über die möglichen Beziehungen zwischen den bereits existierenden Objekten und den neu hinzugefügten Objekten.

Die Firma wizAI hat in Zusammenarbeit mit der Arbeitsgruppe „Künstliche Intelligenz“ an der Universität Koblenz-Landau für die Deutsche Post untersucht, wie die Verwaltung eines Metadaten Repositories durch Einsatz automatischer Klassifikationstechniken (maschineller Lernverfahren [Mit97]) erleichtert werden kann.

Im nächsten Abschnitt wird zunächst der Aufbau der Daten im Repository skizziert; Abschnitt 3 konkretisiert die Problemstellung, und in Abschnitt 4 werden schließlich die Ergebnisse präsentiert.

2. Repräsentation der Daten im Repository

Die Ausgangssituation soll anhand einer Skizze für das Metadaten-Repository (siehe Abbildung 1) verdeutlicht werden. Jede Anwendung wird im Repository durch eine Objektinstanz (*instance of object*) beschrieben. Die Attribute legen jeweils fest, um was für eine Anwendung es sich handelt (*object_type_id*), welche organisatorische Einheit verantwortlich ist (*owner*) etc. Alle Objektinstanzen sind in der OBJECT-Tabelle aufgeführt. Eine Beziehung zwischen zwei Objektinstanzen wird durch eine Referenzinstanz (*instance of reference*) beschrieben. Alle Referenzinstanzen sind in der REFERENCE-Tabelle gruppiert.

Jede Anwendung wird durch eine eigene Objektinstanz beschrieben. Darüberhinaus werden Geschäftsobjekte, und selbst Attribute, die in einer Datenbankanwendung vorkommen, als eigene Objekte beschrieben. Alle

diese Objekte werden in das Repository eingepflegt. Jedes Objekt ist durch eine vordefinierte Menge von Attributen repräsentiert. Die möglichen (Attribut-) Werte für die jeweilige Objektbeschreibung ist dabei nicht fest vorgegeben. Allein das *description*-Attribut kann 2000 Zeichen frei formulierten Text aufnehmen. Obwohl es weitere Informationen im Repository gibt, die bspw. Typen von Objekten beschreiben, z.B. steht object_type_id = 1 für „application + database“, wurden im Rahmen der Studie nur die Objektinstanzen und die Referenzinstanzen betrachtet, um neue Objekte in das vorhandene Repository einzuordnen.

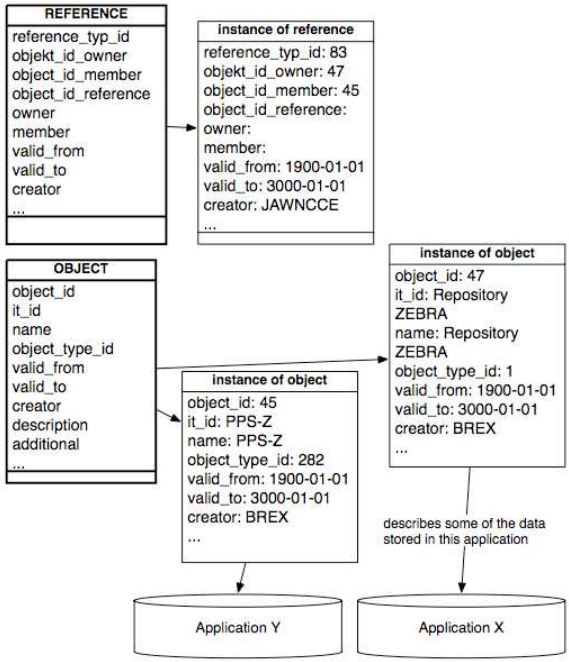

Abb. 1: Aufbau des Repository (Skizze)

2.1 Idee: Erlernen von Klassifikatoren für Objekt-Beziehungen

Die grundlegende Idee für die Automatisierung der Erfassung neuer Objekte besteht darin, Klassifikatoren zu lernen, die entscheiden, ob gewisse Metadaten-Objekte in Beziehung zueinander stehen und falls ja, den Typ der Beziehung feststellen. Konzeptuell betrachtet, handelt es sich hierbei um ein Standard-Klassifikationsproblem, das aus dem Bereich des

Maschinellen Lernens bekannt ist. Im Rahmen eines „Proof of Concept“ hat wizAI das an der Universität Koblenz-Landau entwickelte Klassifikationssystem MIC [Beu01a] eingesetzt, um automatische Klassifikatoren für Beziehungen zwischen Metadaten Objekten zu erlernen. Dieses System unterstützt den Benutzer eines Repositories darin, neue Objekte einzupflegen und schlägt mögliche Beziehungen zu bereits existierenden Objekten im Repository vor. Der Benutzer entscheidet, welche Vorschläge angenommen bzw. verworfen werden. Die Bearbeitung der Vorschläge erfordert weniger Expertenwissen und insbesondere weniger Zeit als das bisherige komplett manuelle Vorgehen.

3. Problemstellung

Die Klassifikationsaufgabe besteht darin, Beziehungen zwischen Metadaten-Objekten zu identifizieren. Ausgangspunkt waren die zur Verfügung stehenden Objektinstanzen, Referenzinstanzen und ein paar Informationen über die Typen von Objekten und Referenzen.

Um den Benutzer bei der Identifikation von Beziehungen zwischen Objekten zu unterstützen, wurden zwei typische Anwendungsszenarien identifiziert.

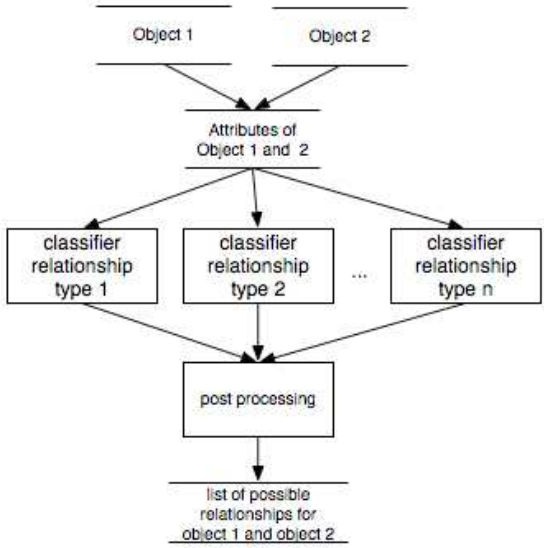

Abb. 2: Klassifikationsszenario 1

Im ersten Szenario hat der Benutzer zwei gegebene Objekte vor sich und möchte mögliche Beziehungen zwischen diesen identifizieren. Das System soll ihm Vorschläge für mögliche Beziehungsarten liefern. Da die Anzahl möglicher Beziehungstypen recht groß sein kann (basierend auf den vorliegenden Beispieldaten waren dies über 100 Beziehungstypen), kann ein derartiger Generator von Vorschlägen das Modifizieren des Repository deutlich beschleunigen. Dieses Szenario ist in Abbildung 2 dargestellt.

Die beiden Objekte werden den zuvor gelernten Klassifikatoren präsentiert. Jeder Klassifikator liefert entweder „Ja“, d.h. die Objekte stehen in Beziehungstyp *X* oder „Nein“, es gibt keine Beziehung vom Typ *X* zwischen den Objekten. Das Resultat des Klassifikationsvorgangs ist eine Liste möglicher Beziehungstypen, die vom System an den Benutzer als Vorschläge weitergereicht werden können. Im zweiten Szenario präsentiert der Benutzer dem System ein einzelnes Objekt, und das System liefert ihm eine Liste anderer Objekte des Repository zusammen mit möglichen Beziehungen, die das gegebene Objekt zu den vorhandenen eingehen kann.

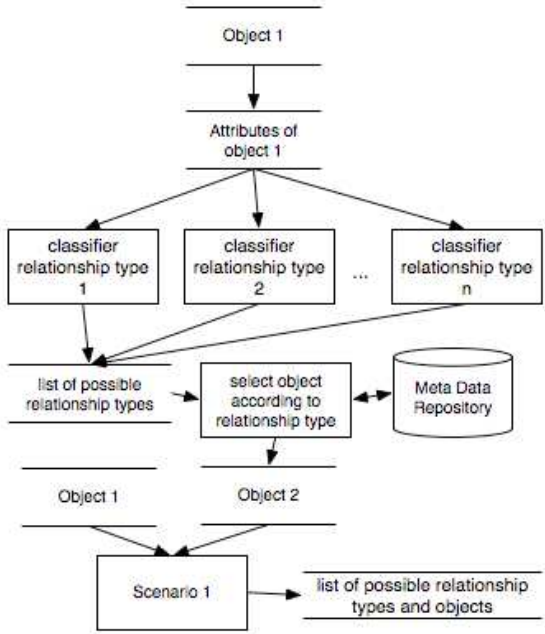

Abb. 3: Klassifikationsszenario 2

Offensichtlich gibt es einen Zusammenhang zwischen Szenario 2 und Szenario 1: Im zweiten Szenario wird jedes Objekt des Repository mit dem neuen Objekt kombiniert und beide zusammen werden dann durch das Szenario 1 verarbeitet. Tatsächlich ist jedoch die Kombination von allen Objekten des Repository mit dem neuen Objekt nicht effizient. Aus diesem Grund wird der Klassifikationsprozess in Szenario 2 in zwei Schritte aufgeteilt (siehe Abbildung 3): Im ersten Schritt, wird das neue Objekt als Eingabe in einen Beziehungstyp-Klassifikator verarbeitet. Im zweiten Schritt, zieht das Klassifikationssystem nur noch solche Objekte aus dem Repository, die über einen der Beziehungstypen aus Schritt 1 mit anderen Objekten in Beziehung stehen können. Das System liefert schließlich zu dem gegebenen neuen Objekt all diejenigen Objekte des Repository, die eine Beziehung mit dem gegebenen Objekt eingehen können. Hierbei wird der Typ einer möglichen Beziehung direkt mit angezeigt.

4. Lernen und Ergebnisse

In diesem Projekte für die Deutsche Post wurden drei Standard Klassifikationsmethoden verwendet: „Naive Bayes Classifiers (NB)“ [Mit97], „Decision Tree (DT)“ [Qui79], sowie „Neuronale Netze (NN)“ [Rum86]. Für jedes Szenario wurden Klassifikatoren mit allen drei Methoden gelernt. Darüberhinaus gibt es für jedes Verfahren gewisse Freiheitsgrade bei der Festlegung positiver und negativer Beispiele, die das Lernen der Klassifikatoren beeinflussen. Hier wurde mit verschiedenen Parametereinstellungen gearbeitet (siehe [Beu03b] für eine ausführliche Beschreibung der Lernkonfigurationen).

Für jedes Szenario und die unterschiedlichen Parametereinstellungen, wurde die Qualität [Chi92] der erlernten Klassifikatoren mittels des F1-Score berechnet. Der F1-Score ist das harmonische Mittel aus Precision und Recall. Precision beschreibt den prozentualen Anteil an korrekten Klassifikation zu allen getroffenen Klassifikationen. Die Precision ist somit ein Maß für die Genauigkeit. Recall beschreibt den prozentualen Anteil der korrekt klassifizierten Beispiel bzgl. der Gesamtmenge der korrekt zu klassifizierenden Daten. Recall beschreibt somit die Vollständigkeit der erlernten Klassifikatioren.

Abbildung 4 zeigt die Ergebnisse: Für das Szenario 1 erhält man informell ausgedrückt in etwa 85% der Fälle, für zwei gegebene Objekte die gewünschten (richtigen) Beziehungstypen angezeigt. Im Szenario 2 werden in über 90% der Fälle die richtigen Objekte mit Beziehungstypen zu einem gegebenen neuen Objekt angezeigt.

	Decision Tree	Naive Bayes	Neural Net
Szenario 1	85,40	70,25	84,02
Szenario 2	92,22	90,69	92,13

Abb. 4: Klassifikationsergebnisse (Durchschnitt aller F1 Ergebnisse)

5. Zusammenfassung

Abschließend werden die wichtigsten Erfahrungen und Ergebnisse aus diesem Projekt noch einmal zusammengefaßt.

In einem Repository werden Metadaten über Anwendungen, Daten und Systeme verwaltet. Die Erstellung eines Metadaten-Repository ist eine Maßnahme, die im Rahmen einer Wissensmanagement-Strategie eingesetzt wird, um einen „Single Point of Truth“ für die im Unternehmen vorhandenen Systeme, Anwendungen und Daten zu erhalten. Der Erfolg des Repository hängt von der Qualität der Daten ab. Erfassung und Pflege der Metadaten ist im Allgemeinen mit einem hohen zeitlichen und personellen Aufwand verbunden.

Die Idee, die dem hier beschriebenen Projekte zugrunde lag, besteht darin, mittels automatischer Klassifikationsverfahren, Beziehungen zwischen Objekten in einem Metadaten-Repository zu erkennen und dem Repository-Benutzer als Vorschläge zu präsentieren.

Der Einsatz maschineller Lernverfahren und die Untersuchung der vorliegenden Problemstellung als Textklassifikationsproblem haben erstaunlich gute Ergebnisse geliefert. Mit einer Präzision von ~85% Prozent für Szenario 1 und ~90% für Szenario 2 kann der Einsatz von Klassifikationsverfahren für die genannte Aufgabe generell empfohlen werden.

Folgende Erkenntnisse wurden beim Einsatz von Klassifikationsverfahren für ein Metadaten Repository gemacht: Eine gute Precision allein sollte nicht ausschlaggebend dafür sein, ein Lernverfahren einem anderen vorzuziehen. In den Szenarien, die hier vorgestellt wurden, bekommt der Benutzer mögliche Beziehungstypen vorgeschlagen und kann dem System sagen, ob diese korrekt sind (handelt es sich hierbei tatsächlich um eine Beziehung oder nicht?). In diesen Szenarien sind sogenannte „falsch negativ“ klassifizierte Beziehungen (solche die existieren, jedoch nicht angezeigt werden), problematischer als „falsch positiv“ Klassifizierte. Erstere werden dem Benutzer nicht präsentiert, und können deshalb nicht manuell hinzugefügt werden. Die falsch positiv Klassifizierten werden dem

Benutzer gezeigt und dieser kann sie entsprechend verwerfen. Mittels automatischer Klassifikation kann der Vorgang für das Erfassen und Pflegen der Daten deutlich vereinfacht werden. Beziehungen werden automatisch erkannt und können in das Repository eingepflegt werden. Grundsätzlich handelt es sich hierbei jedoch um ein semi-automatisches Verfahren, da die generierten Vorschläge vom Repository-Administrator letztendlich noch überprüft werden. Der hier beschriebene Ansatz kann zukünftig noch optimiert werden, indem einerseits die Repräsentation der Daten für die Klassifikationsverfahren modifiziert wird und andererseits indem mehr strukturelle Information über die Metadaten aus dem Repository ausgenutzt wird.

Literatur

[Beu01a] G. Beuster. MIC-A System for Classification of Structured and Unstructured Texts. Master thesis, University Koblenz, 2001.

[Beu03b] G. Beuster and U. Furbach and M. Gross-Hardt and B.Thomas: Automatic Classification for the Identification of Relationships in a Meta-Data Repository. In 6th International Conference on Discovery Science (DS 2003), Sapporo, Japan, Proceedings, pages 282-289. Lecture Notes in Artifical Intelligence, Springer-Verlag, 2003.

[Bou98] A. Bouguettaya, B. Benatallah, and A. K. Elmagarmid: Interconnecting Heterogeneous Information Systems. Kluwer Academic Publishers, 1998.

[Chi92] N. Chinchor. Muc-4 evaluation metrics. In Fourth Message Understanding Conference, pages 22-29. Morgan Kaufmann, 1992.

[Mar00] D. Marco. Building and Managing the Meta Data Repository: A Full Lifecycle Guide. John Wiley & Sons, 2000.

[Mit97] T. M. Mitchell. Machine Learning. McGraw-Hill International Editions, 1997.

[Qui79] J. Quinlan. Discovering rules by induction from a large collection of examples. In D. Michie, editor, Expert systems in the Micro-Electronic Age, pages 168-201. Edinburgh University Press, Edinburgh, 1979.

[Rum86] D. D. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by backpropagating errors. Nature, pages 533-536, 1986.

[She90] A. Sheth and J.Larson. Federated database systems for managing distributed, heterogeneous, and autonomous databases. ACM Computing Surveys, Vol.22, pages 183-236, 1990.