Ein Ansatz zur Repräsentation und Verarbeitung großer Korpora

Rüdiger Gleim, Alexander Mehler, Hans-Jürgen Eikmeyer, Hannes Rieser

Universität Bielefeld
Universitätsstraße 25, 33615 Bielefeld
{Ruediger.Gleim,Alexander.Mehler,Hansjuergen.Eikmeyer,Hannes.Rieser}@
uni-bielefeld.de

Zusammenfassung Seit den Anfängen der computergestützten Korpuslinguistik hat sich das Spektrum der untersuchten Daten qualitativ und quantitativ kontinuierlich weiterentwickelt. Dabei ist jener Schub hervorzuheben, der durch die Repräsentation multimedialer und multimodaler Daten zu verzeichnen ist. Eine konstante Herausforderung besteht zudem in der Verarbeitung immer größerer Korpora. Der vorliegende Beitrag diskutiert die Problematik der Vereinbarkeit großer Datenmengen auf der einen Seite und zunehmend komplexeren Annotationen auf der anderen Seite, und zwar unter der Bedingung, dass die resultierenden Korpora mit vertretbarem Aufwand recherchierbar bleiben. Zu diesem Zweck wird mit HyGraphDB ein Datenbanksystem vorgestellt, welches die persistente Erzeugung und Verwaltung komplexer Graphstrukturen ermöglicht. Das HyGraphDB-System wird am Beispiel der Wikipedia evaluiert.

Key words: Textkorpora, Graph eXchange Language, Wikipedia

1 Einleitung

Das Spektrum der Korpora, welche im Rahmen der computergestützten Korpuslinguistik untersucht werden, hat sich seit Veröffentlichung des Brown Korpus (Kuĉera and Francis, 1967) deutlich erweitert. Standen damals Digitalisierungen von Produkten klassischer Printmedien (wie etwa historische Schriftzeugnisse, Buchpublikationen oder Transkriptionen gesprochener Sprache) im Mittelpunkt, so sind es in Zeiten einer umfassenden Verbreitung von Internet und *pervasive computing* zunehmend Korpora von Ausdruckseinheiten der Neuen Medien. Beispielgebend für die hiermit einhergehende Komplexitätssteigerung sind multimodale Datenstrukturen wie sie bei der Untersuchung des *alignment in communication* (Pickering and Garrod, 2004) anfallen. Diesbezügliche Forschungsansätze weisen in Richtung der Entwicklung ausdrucksmächtiger Graphmodelle jenseits von Baumstrukturen und ihrer polyhierarchischen Erweiterungen (Kranstedt et al., 2007).

Auf der anderen Seite ermöglichen steigende Rechenleistungen und Speicherkapazitäten prinzipiell die Verarbeitung immer größerer und komplexerer

Korpora. Eine zweite Herausforderung stellen daher wachsende Korpusumfänge dar. So weist beispielsweise der Bedarf an Referenzkorpora für das maschinelle Lernen in Richtung einer vollständigen Nutzbarmachung frei verfügbarer Dokumentsammlungen wie der Online-Enzyklopädie Wikipedia (Gleim et al., 2006b). Durch ihren einfachen Zugang sowie die Möglichkeit, Artikel kollaborativ zu erstellen, hat sie, je nach Sprache, einen Umfang von mehreren hunderttausend Dokumenten erreicht. Die Wikipedia besticht zudem durch den hohen Grad der durch Hyperlinks induzierten Textvernetzung (Mehler, 2006, Zlatic et al., 2006). Mit der Vielzahl ihrer teils multimedialen Dokumente kann sie daher als ein Referenzbeispiel für die Auslotung der Möglichkeiten und Grenzen der Verarbeitung *von* bzw. Recherche *in* großen Korpora dienen.

Wie aber sind solche Korpora effizient verarbeitbar? Seit geraumer Zeit haben sich XML-basierte Sprachen für die Repräsentation und den Austausch linguistischer Korpora durchgesetzt (Stührenberg et al., 2006). Die Verarbeitung großer Korpora stößt jedoch immer wieder an Grenzen der effizienten Berechnung und Speicherverwaltung. Beispielsweise umfasst der XML-Export[1] der deutschen Ausgabe der Wikipedia vom 30.11.2006 ca. 3.8GB. Dieser Export enthält keine explizite Annotation der Links; zudem sind die Artikel in einer MediaWiki-spezifischen Syntax kodiert. Sollen auch diese Daten XML-basiert repräsentiert werden, sind weitere Bearbeitungsschritte mit zusätzlichem Speicherbedarf vonnöten. Umfangreiche XML-Dokumente sind dann aber nur noch mittels effizient programmierter, eigens angepasster SAX-Parser[2] verarbeitbar. Dies führt dazu, dass für texttechnologische Experimente die jeweils relevanten Informationen extrahiert und letztlich doch wieder in proprietären Formaten gespeichert werden. *Sind also große, mehrfachstrukturierte Textkorpora und generische Korpusrepräsentationsformate unvereinbare texttechnologische Zielsetzungen?* Die Untersuchung dieser Frage bildet den Gegenstand des vorliegenden Beitrags. Es geht um die Repräsentation und Verarbeitung großer Korpora mit den Ausdrucksmitteln einer XML-basierten Graphbeschreibungssprache. Ziel ist dabei nicht die Entwicklung eines neuen Repräsentationsformats, sondern die Untersuchung der Frage, inwieweit sich große, auf Graphen abgebildete Korpora effizient maschinell verarbeiten lassen. Zu diesem Zweck beschreiben wir zunächst eine exemplarische Abbildung der Wikipedia auf die Graph eXchange Language (GXL) (Holt et al. (2006)), eine verbreitete XML-basierte Sprache zum Austausch von Graphen. Anschließend stellen wir mit HyGraphDB ein auf BerkeleyDB[3] basiertes Datenbanksystem vor, welches die Verarbeitung von GXL-Dokumenten erlaubt. Schließlich untersuchen wir anhand der Wikipedia, inwieweit sich dieses System für große Korpora eignet.

Sektion 2 thematisiert das als Prüfstein referierte Beispiel der Wikipedia. Sektion 3 stellt die Architektur der HyGraphDB vor und beschreibt ein Experiment, welches die Speicherplatz- und Zeiteffizienz des Systems untersucht. Abschließend werden die gewonnenen Ergebnisse diskutiert und bewertet.

[1] `http://download.wikimedia.org`
[2] z.B. `http://xml.apache.org/xerces-c/`
[3] `http://www.oracle.com/database/berkeley-db/db/index.html`

2 Zur Wikipedia in der texttechnologischen Praxis

Wie in der Einleitung dargestellt wurde, ist eine Entwicklung hin zu immer größeren Korpora bei gleichzeitiger Steigerung der Komplexität der Primärdaten und ihrer Annotationen beobachtbar. So sind zwar syntaktische oder logische Dokumentstrukturen vielfach mit Hilfe von Baumstrukturen explizierbar. Bei Hinzunahme von Annotationen von Kohäsions- oder Kohärenzrelationen müssen jedoch bereits allgemeinere Graphen als Ausdrucksmittel herangezogen werden. Als Beschreibungsmittel sind Graphen insbesondere für multimodale Datenstrukturen relevant. Dabei geht es um Korpora zur Abbildung von Kommunikationshandlungen, die im Hinblick auf ihre sprachlichen, gestischen, visuellen, akustischen und teils auch haptischen Manifestationen zu erfassen sind. Diese werden auf mehreren Ebenen annotiert und interrelationiert (Kranstedt et al., 2007). Die dazu verwendeten Repräsentationen greifen bereits auf hierarchische Hypergraphen zurück, welche die Darstellung heterogener Relationen erlauben. Allgemein gesprochen sind zwei grundlegende Ansätze zur Bewältigung solcher Strukturen zu unterscheiden:

- In Zusammenhang der Entwicklung spezialisierter Annotationswerkzeuge (z.B. Milde and Gut (2004)) werden entsprechend spezialisierte Repräsentationsformate entwickelt, welche an den zu verarbeitenden Gegenstand, wie etwa Gesprächskorpora, angepasst sind. Diese Ausrichtung ermöglicht eine stromlinienförmige Datenverarbeitung, erschwert jedoch die Übertragbarkeit der Formate auf andere Bereiche.
- Die Vertreter des zweiten Ansatzes lösen sich von dem konkreten Anwendungsbereich und zielen auf eine generischere Anwendbarkeit (Schmidt et al. (2006)). Dabei bildet die Graphentheorie eine geeignete formale Basis für die Erstellung der zugehörigen Datenmodelle (Bird and Liberman (1999)).

Auch in anderen Forschungsbereichen — wie dem Software-Engineering oder der Prozessmodellierung — sind generische Sprachen zur Repräsentation von Graphen entwickelt worden, so z.B. die GraphXML (Herman and Marshall (2000)), die GraphML (Brandes et al. (2002)) oder die GXL (Holt et al. (2006)). Der Ansatz einer möglichst allgemeinen, nicht an eine bestimmte Annotationssoftware gebundenen Repräsentation von Graphstrukturen ist für die linguistische Nachhaltigkeit von Korpora von essentieller Bedeutung (Schmidt et al. (2006)). Sie ist daher auch für den vorliegenden Beitrag leitend. Hierzu wird speziell das Beispiel der Wikipedia gewählt, die aufgrund ihrer Größe und ihrer strukturellen Dichte als Referenzbeispiel geeignet ist.

2.1 Wikipedia

Die Wikipedia zählt zu den umfangreichsten Beispielen der kooperativen Textproduktion. Infolgedessen ist sie vielfach Objekt texttechnologischer und informationswissenschaftlicher Untersuchungen geworden. Hierzu zählen Analysen

Tabelle 1. Zur Statistik der deutschen Wikipedia vom 30.11.2006.

Dokumenttyp	Instanzen	Links	Dokumenttyp	Instanzen	Links
Artikel	835.624	16.772.358	Artikel Diskussion	168.729	193.782
Benutzer	79.267	2.741.919	Benutzer Diskussion	70.318	426.453
Wikipedia	7.604	640.683	Wikipedia Diskussion	2.847	9.572
Bild	113.009	105.154	Bild Diskussion	3.761	2.324
MediaWiki	2.074	1.054	MediaWiki Diskussion	96	—
Vorlage	10.152	14.167	Vorlage Diskussion	1.309	1.932
Hilfe	168	8.190	Hilfe Diskussion	119	192
Kategorie	35.945	1.389.881	Kategorie Diskussion	2.526	3.188
Portal	5.106	31.451	Portal Diskussion	1.074	3.277

der Textvernetzung (Capocci et al., 2006, Mehler, 2006, Zlatic et al., 2006) ebenso wie Anwendungen aus dem Bereich der Informationsextraktion, des kollaborativen Schreibens (Holloway et al., 2005) oder der Strukturklassifikation (Gleim et al., 2006a). Diese Beispiele verdeutlichen die Forschungsrelevanz intertextueller Strukturen auch über Artikeldaten hinaus. Neben der Größe der Wikipedia bilden sie daher einen weiteren Prüfstein für die Korpusverarbeitung. Dabei ist zu beachten, dass die Wikipedia neben Primärartikeln eine Reihe weiterer Dokumente umfasst, welche durch Namensräume klassifiziert werden. Im Folgenden wird allgemeiner von *Dokumenten* gesprochen, um diese Namensraum- bzw. Dokumenttyp-Zugehörigkeit zu berücksichtigen. So werden beispielsweise die Typen *Benutzer*, *Vorlage* und *Kategorie* unterschieden. Die Instanzen letzteren Namensraums konstituieren zudem Kategoriensysteme, welche für die Erstellung themenspezifischer Korpora ausgewertet werden können (Gleim et al., 2006b). Dies erlaubt es, themenspezifische Korpora zu extrahieren, die ausschließlich Artikel der Kategorie Musik umfassen. Die Instanzen der Dokumenttypen sind weiterhin mittels Diskussionsseiten kommentierbar, so dass sich der in Tabelle 1 gegebene Überblick über den Umfang der deutschen Wikipedia zum 30.11.2006 ergibt, welcher die Basis für die vorliegende Untersuchung bildete. Die Spalte Links führt die Anzahl der Hyperlinks auf, welche auf Dokumente des jeweiligen Typs verweisen. Die Diskrepanz zwischen der Gesamtzahl der Artikel und der diesbezüglichen offiziellen Angabe rührt vor allem von dem hohen Anteil so genannter Redirect-Knoten her, die lediglich der Weiterleitung dienen, der Übersicht halber aber den Artikeln zugeschlagen wurden — Redirect-Knoten werden mittels HyGraphDB dennoch separat erfasst. Insgesamt umfassen alle Dokumente ca. 450 Mio. Token, wovon etwa 240 Mio. auf Artikel entfallen. Die Wikipedia ist somit sowohl inhaltlich als auch in ihrer Linkstruktur sehr umfangreich.

Wie ist eine solche Datenmenge für texttechnologische Arbeiten geeignet repräsentierbar? In diesem Beitrag soll eine Antwort auf diese Frage aus technischer Sicht gegeben werden (das zugrundeliegende theoretische Graphmodell wird hingegen in Mehler (2007) erläutert): Die Enzyklopädie setzt auf der Software MediaWiki auf, welche zur Datenhaltung MySQL verwendet. Die *Wikipedia Foundation* bietet neben der Online-Version auch einen XML-Dump der Datenbank an, auf welchem die vorliegende Untersuchung basiert. Die Textinhalte

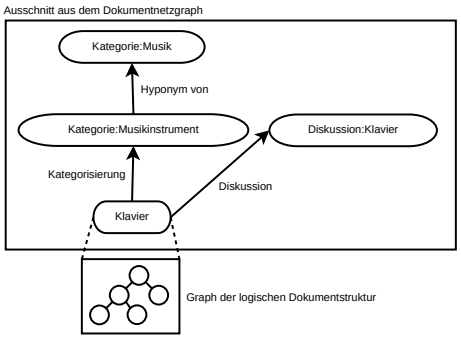

Abbildung 1. Beispiel für Graphrepräsentation der Wikipedia

selbst sind in einer MediaWiki-spezifischen Syntax gespeichert, welche separat zu parsen ist, um die Verlinkung der Artikel und deren Sektionsstruktur zu extrahieren. Diese Form der Repräsentation ist, gemessen am Umfang der Wikipedia (3.8GB), kompakt. Texttechnologische Untersuchungen können jedoch eine Aufbereitung erforderlich machen, welche einen direkten Zugriff auf die textinterne logische Dokumentstruktur bis hinab zur Tokenebene ermöglicht. Eine generische Repräsentation solcher Strukturen, die von speziellen Anwendungen abstrahiert, kann graphentheoretisch vollzogen werden (Mehler, 2007): Die Dokumente der unterschiedlichen Dokumenttypen werden als Knoten eines benannten Graphen gespeichert, dessen Knoten und Kanten typisiert sind. Die logische Dokumentstruktur bis hinunter zur Tokenebene wird als strukturierter Knoten dem jeweiligen Dokumentknoten untergeordnet — es entsteht ein geschachtelter Graph. Abbildung 1 exemplifiziert die Graphrepräsentation des Artikels Klavier, der als separater Knoten erscheint, wobei typisierte Kanten auf das zugehörige Diskussionsdokument sowie auf ein Kategoriendokument verweisen. Der Artikelknoten enthält somit einen eingebetteten Graphen, der seine logische Dokumentstruktur repräsentiert. Auf diese Weise sind prinzipiell Mehrebenenstrukturen erfassbar, und zwar in Form nebengeordneter, eingebetteter Graphen. Dieser Ansatz bewahrt Zugriffsmöglichkeiten auf sämtliche Ebenen der Korpusstrukturierung — ausgehend vom Gesamtkorpus über seine Einzeldokumente bis hin zu den elementaren Sektionen und blätterbildenden Token. Die Explikation dieser Strukturen, die mittels eines Vorverarbeitungsschritts aus der XML-Datei zu explorieren sind, induzieren einen zusätzlichen Speicheraufwand. Sektion 3.3 untersucht, inwieweit die HyGraphDB-API diesen Ansatz unterstützt.

3 HyGraphDB

Im Folgenden wird die Problematik der Verarbeitung von Graphstrukturen diskutiert. Anschließend wird die Architektur der HyGraphDB dargestellt. Die Sektion schließt mit einem Testszenario am Beispiel der Wikipedia.

3.1 Problematik und Anforderungen

Bei der Konzeption eines korpusbasierten Experiments stößt man auf zwei markante Grenzen: *Speicherbedarf* und *Verarbeitungszeit*. Designentscheidungen laufen letztlich auf einen Trade-off zwischen diesen beiden Größen hinaus, deren Abwägung an eine Reihe von Einzelentscheidungen geknüpft ist:

- Oben wurde bereits die Unterscheidung zwischen Baumstrukturen und allgemeineren Graphstrukturen bis hin zu Hypergraphen und geschachtelten Graphen skizziert, wobei das Referenzbeispiel der Wikipedia eine Festlegung im Hinblick auf die Ausdrucksmächtigkeit letzterer Strukturen erfordert. Dies gilt insbesondere dann, wenn über das Instrumentarium einfacher Kookkurrenzanalysen hinaus die logische Dokumentstruktur oder gar die Textvernetzungsstruktur zum Bezugspunkt von Textanalysen gemacht werden soll. Dies ist wiederum insofern relevant, als hiermit eine Öffnung in Richtung der Erfassung auch multimodaler Korpora einhergeht, welche durch eine vergleichbare Komplexität gekennzeichnet sind.
- Eine weitere Bezugsgröße zur Abwägung zwischen Speicherbedarf und Verarbeitungszeit bildet der Datenzugriff. Ein beschleunigter Zugriff auf einzelne Graphelemente und die Traversierung der Graphstruktur ist zumeist nur mittels redundanter Datenhaltung erreichbar. Ein Beispiel für den hierdurch erzielbaren Zeitvorteil bildet die Beziehung zwischen Graphen und ihren Knoten. Theoretisch genügen Referenzen von Knoten auf die Graphen, denen sie angehören. Eine Abfrage nach dem Graph eines Knotens ist dann effizient beantwortbar. Soll dagegen die gesamte Knotenmenge eines Graphen bestimmt werden, müssen bei letzterem Ansatz *alle* Knoten dahingehend untersucht werden, ob sie dem Zielgraphen angehören. Um solche Abfragen effizient zu halten, ist folglich eine Indizierung vonnöten, die jedoch den Speicherbedarf erhöht.
- Ein verwandtes Thema bildet das Update von Korpora. Im Falle statischer Graphstrukturen sind Speichertechniken einsetzbar, die eine kompakte Speicherung der Strukturinformationen ermöglichen (etwa mittels Kompressionsverfahren). Handelt es sich jedoch um Korpora, deren Annotationen einer laufenden Veränderung unterliegen, scheidet letztere Möglichkeit aus.

Prinzipiell existieren viele Möglichkeiten, Graphstrukturen im Arbeitsspeicher zu verwalten, von einfachen Adjazenzmatrizen bis hin zu Objekten eines ausgereiften Klassendesigns. Spätestens wenn die Größe der Graphen den verfügbaren Speicherplatz übersteigt, ist eine persistente Datenhaltung unabdingbar. Einen naheliegenden Ansatz bilden XML-basierte Serialisierungen etwa mittels der GXL. Der vermeintlich einfachen Handhabbarkeit solcher Dokumente steht eine Reihe von Nachteilen gegenüber. Für Updates muss das gesamte Dokument eingelesen, geändert und wieder gespeichert werden. Sollen mehrere Benutzer gleichzeitig am Dokument arbeiten, entstehen Zugriffskonflikte. Schließlich sind Datenänderungen nur über entsprechende Sicherungskopien rückgängig zu machen, da eine geeignete Transaktionsverwaltung fehlt. Die HyGraphDB beruht daher auf einer Datenbankarchitektur, ohne auf die Möglichkeiten geschachtelter Relationen und ihrer Serialisierung mittels XML-Dokumenten zu verzichten.

3.2 Architektur

Die HyGraphDB ist eine in C++ geschriebene Bibliothek zur Verarbeitung persistenter Graphstrukturen. Sie ist nicht für Endanwender konzipiert, sondern zielt auf die Entwicklung von Werkzeugen für graphbasierte Korpusrepräsentationen. Die Ausdrucksmächtigkeit der berücksichtigten Graphstrukturen ist equivalent zur GXL und umfasst kanten- und knotentypisierte, geordnete, gerichtete, hierarchische Hypergraphen. Desweiteren können alle Graphelemente mit beliebig geschachtelten Attributen annotiert werden. Die Funktionen umfassen alle typischen Graphoperationen wie das Einfügen von Graphen, Knoten, Kanten, Relationen, Typisierungen und Attributen. Diese, hier allgemein als *Elemente* zusammengefassten Einheiten können im Rahmen des GXL-Schemas Modifikationsoperationen unterzogen werden. Teilgraphen können beispielsweise kopiert oder innerhalb einer Struktur "umgehängt" werden. Das intendierte Anwendungsspektrum von HyGraphDB umfasst den kompletten Korpuslebenszyklus, von der initialen Korpuserstellung über die Korpusaufbereitung und -annotation bis hin zur Verfügbarmachung von Suchanfragen, Analysen und Exporte in einschlägige Formate. Hierzu werden Module für den Import/Export von Dokumenten texttechnologisch relevanter Formate entwickelt. Dies umfasst unter anderem die GXL sowie die Austauschformate der Annotationstools Anvil, Elan, EXMARaLDA und Praat (siehe Rohlfing et al. (2006) zu einer Vergleichsstudie dieser Tools).

Die HyGraphDB erzeugt eine Abbildung der zu verarbeitenden Graphstrukturen in B-Bäume, welche mittels der BerkeleyDB verwaltet werden. Diese sorgt für die persistente Speicherung der Daten und stellt das Instrumentarium für Transaktionen und den konsistenten Mehrbenutzerbetrieb nach dem ACID-Schema (Atomicity, Consistency, Isolation, Durability) bereit. Auf diese Weise ist es möglich, beliebige Sequenzen von Graphoperationen in einer Transaktion zu kapseln. Die API der HyGraphDB umfasst eine Reihe von Standardindexierungen von GXL-Attributen. Auf diese Weise ist es zum Beispiel möglich, alle String-Attribute eines bestimmten Typs zu ermitteln, die einen bestimmten Inhalt aufweisen. Diese Indexierungen sind optional und werden nur dann berechnet, wenn sie explizit eingebunden werden. Neben den Standardvorgaben können auch eigene Indexierungsfunktionen geschrieben und eingebunden werden. Dadurch sind korpusspezifische Datenzugriffe optimierbar, ohne dass ein Export in ein spezialisiertes Format notwendig wird.

Die HyGraphDB-API ermöglicht zusammen mit den Suchabfragen auf der Basis der letztgenannten Indizes die Ausprogrammierung sehr spezieller Suchanfragen. Dies betrifft etwa dokumentstruktursensitive Kollokationsanalysen und die Suche nach Strukturmustern. Die Entwicklung bzw. Adaption einer entsprechenden Anfragesprache ist Gegenstand der gegenwärtigen Entwicklungsarbeit.

3.3 Ein Testszenario

Dieser Abschnitt beschreibt eine Evaluierung des aktuellen Entwicklungsstands der HyGraphDB. Das Experiment gliedert sich in zwei Teile: Zunächst wird eine Teilmenge der Wikipedia in die Datenbank importiert. Dabei soll neben der

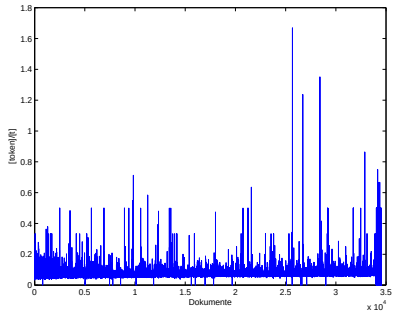

Abbildung 2. Laufzeitverhalten des Imports bzgl. der eingefügten Token/ms

benötigten Gesamtzeit und dem erforderlichen Speicherplatz untersucht werden, wie sich die Einfügezeiten pro Artikel im Laufe des Imports verändern. Daraus können Rückschlüsse für das Laufzeitverhalten bei größeren Datenmengen gezogen werden. Der zweite Teil besteht in einem *Ex*port der Graphstrukturen in die GXL. Neben der Laufzeit ist dabei das Größenverhältnis der dateibasierten GXL-Repräsentation und der HyGraphDB relevant.

Als Testdaten selektieren wir sämtliche Artikel, welche direkt oder mittelbar der Kategorie *Literatur* angehören. Dieses Korpus umfasst 34.528 Dokumente, 239.493 inzidente Kanten sowie 21.198.460 Tokens. Die Zieldatenstruktur innerhalb der Datenbank entspricht einem Graphen, der alle importierten Artikel als Knoten und deren Links als Kanten enthält. Der Import erfolgt zweistufig: Zunächst wird der Dokumentnetzgraph erstellt und die Token der Dokumente als Subgraphen der Artikelknoten gespeichert. Nachfolgend werden die Hyperlinks eingelesen und auf Kanten abgebildet. Abbildung 2 zeigt die Entwicklung der Einfügezeiten pro Dokument im Laufe des Imports. Da die Anzahl der Token n pro Dokument für die Laufzeit t ausschlaggebend ist, geben wir das Verhältnis t/n an. Bei der betrachteten Artikelmenge ist kein signifikanter Anstieg der Laufzeit beobachtbar. Weitere Experimente müssen zeigen, ob dieses Ergebnis fortbesteht. Im zweiten Teil des Experiments werden die Artikel aus der Datenbank in eine GXL-Datei exportiert. In der Datenbankrepräsentation werden 1177,75 MB für die Nutzdaten, sowie 510,85 MB für den Token-Index benötigt. Die nach einer Laufzeit von 526,769s erstellte GXL-Datei umfasst 963,86 MB. Der Speicherbedarf der Linearisierung liegt also ca. 20% unter dem der Datenbank.

4 Ausblick

In diesem Beitrag wurde ein Ansatz zur Verwaltung großer Dokumentkorpora vorgestellt. Die bisherigen Evaluationen weisen in eine vielversprechende Richtung, zumal der Ansatz Datenbankfunktionalitäten integriert. Anstehende Weiterentwicklungen zielen auf die komplette Verfügbarmachung von Korpora aus dem Bereich des Web 2.0 wie auch auf die Erfassung multimodaler Korpora.

Literaturverzeichnis

Bird, S. and Liberman, M. (1999). A formal framework for linguistic annotation. Technical Report MS-CIS-99-01, Department of Computer and Information Science.

Brandes, U., Eiglsperger, M., Herman, I., Himsolt, M., and Marshall, M. (2002). GraphML progress report: Structural layer proposal. In *Proc. 9th Intl. Symp. Graph Drawing (GD 2001)*, pages 501–512. Springer-Verlag.

Capocci, A., Servedio, V. D. P., Colaiori, F., Buriol, L. S., Donato, D., Leonardi, S., and Caldarelli, G. (2006). Preferential attachment in the growth of social networks: the case of wikipedia. `http://www.citebase.org/cgi-bin/citations?id=oai:arXiv.org:physics/0602%026`.

Gleim, R., Mehler, A., and Dehmer, M. (2006a). Web corpus mining by instance of wikipedia. In Kilgarriff, A. and Baroni, M., editors, *Proceedings of the EACL Workshop on Web as Corpus, Trento, Italy, April 3-7*.

Gleim, R., Mehler, A., Dehmer, M., and Pustylnikov, O. (2006b). Aisles through the category forest — utilising the wikipedia category system for corpus building in machine learning. 3rd International Conference on Web Information Systems and Technologies (WEBIST '07), March 3-6, 2007, Barcelona.

Herman, I. and Marshall, M. S. (2000). GraphXML — an XML-based graph description format. In *Graph Drawing*, pages 52–62.

Holloway, T., Božičević, M., and Börner, K. (2005). Analyzing and visualizing the semantic coverage of wikipedia and its authors. `http://tw.arxiv.org/abs/cs.IR/0512085`.

Holt, R. C., Schürr, A., Elliott Sim, S., and Winter, A. (2006). GXL: A graph-based standard exchange format for reengineering. *Science of Computer Programming*, 60(2):149–170.

Kranstedt, A., Lücking, A., Mehler, A., Pfeiffer, T., and Rieser, H. (2007). A multimodal corpus for speech and pointing gestures. Submitted.

Kuĉera, H. and Francis, W. N. (1967). *Computational Analysis of Present-Day American English*. Brown University Press, Providence, RI, USA.

Mehler, A. (2006). Text linkage in the wiki medium – a comparative study. In Karlgren, J., editor, *Proceedings of the EACL Workshop on New Text – Wikis and blogs and other dynamic text sources, April 3-7, 2006, Trento, Italy*, pages 1–8.

Mehler, A. (2007). Structure formation in the web. A graph-theoretical model of hypertext types. In Witt, A. and Metzing, D., editors, *Linguistic Modeling of Information and Markup Languages. Contributions to Language Technology. Series: Text, Speech and Language Technology*. Springer, Dordrecht.

Milde, J.-T. and Gut, U. (2004). TASX — eine XML-basierte Umgebung für die Erstellung und Auswertung sprachlicher Korpora. In Mehler, A. and Lobin, H., editors, *Automatische Textanalyse: Systeme und Methoden zur Annotation und Analyse natürlichsprachlicher Texte*, pages 249–264. Verlag für Sozialwissenschaften.

Pickering, M. J. and Garrod, S. (2004). Toward a mechanistic psychology of dialogue. *Behavioral and Brain Sciences*, 27:169–226.

Rohlfing, K., Loehr, D., Duncan, S., Brown, A., Franklin, A., Kimbara, I., Milde, J.-T., Parrill, F., Rose, T., Schmidt, T., Sloetjes, H., Thies, A., and Wellinghoff, S. (2006). Comparison of multimodal annotation tools. *Gesprächsforschung*, 7:99–123.

Schmidt, T., Chiarcos, C., Lehmberg, T., Rehm, G., Witt, A., and Hinrichs, E. (2006). Avoiding data graveyards: From heterogeneous data collected in multiple research projects to sustainable linguistic resources. *Proceedings of the E-MELD workshop*.

Stührenberg, M., Witt, A., Goecke, D., Metzing, D., and Schonefeld, O. (2006). Multidimensional markup and heterogeneous linguistic resources. In *Proceedings of the 5th Workshop on NLP and XML (NLPXML-2006): Multi-Dimensional Markup in Natural Language Processing*, pages 85–88.

Zlatic, V., Bozicevic, M., Stefancic, H., and Domazet, M. (2006). Wikipedias: Collaborative web-based encyclopedias as complex networks. `http://www.citebase.org/cgi-bin/citations?id=oai:arXiv.org:physics/0602%149`.