Stand off-Annotation für Textdokumente: Vom Konzept zur Implementierung (zur Standardisierung?)*

Manuel Burghardt & Christian Wolff
Institut für Information und Medien, Sprache und Kultur (I:IMSK)
Universität Regensburg
93040 Regensburg, Germany
{manuel.burghardt,christian.wolff}@sprachlit.uni-regensburg.de

Zusammenfassung

Stand off-Annotation beschreibt die logische Trennung von Primärdaten und Annotation. Dieses Konzept läßt sich bis in die 90er Jahre zurückverfolgen und ist seitdem auf vielfältige Weise interpretiert und implementiert worden. Der vorliegende Beitrag untersucht, wie sich die verschiedenen Umsetzungen der *stand off*-Annotation voneinander unterscheiden und versucht Vor- und Nachteile der einzelnen Ansätze herauszuarbeiten, um künftigen Standardisierungsansätzen im Bereich *stand off*-Annotation den Weg zu ebnen. Bereits bestehende Standardisierungsansätze werden abschließend diskutiert.

1 Motivation: *Stand off*-Annotation heute

Seit Mitte der 90er Jahre erstmals die Idee einer Trennung von Markup und Primärdaten durch semantische Hyperlinks (Thompson & McKelvie, 1997) formuliert wurde, hat sich *stand off*-Annotation als *de facto*-Standard für die Metadatenanreicherung im Bereich des *literary and linguistic computing* etabliert, insbesondere bei Mehrebenen- oder Zeitleistenannotationen. Trotz der Verbreitung und Anwendung des *stand off*-Konzepts ist man von einer standardisierten Architektur noch weit entfernt. Tatsächlich wird das *stand off*-Konzept sehr unterschiedlich interpretiert. Diese Studie versucht, das Spektrum vorhandener *stand off*-Implementierungen zu erfassen und sie miteinander zu vergleichen, um Vor- und Nachteile der einzelnen Ansätze aufzuzeigen.

* Erschienen in: C. Chiarcos, R. Eckart de Castilho, M. Stede (Hrsg.), *Von der Form zur Bedeutung: Texte automatisch verarbeiten / From Form to Meaning: Processing Texts Automatically*. Tübingen: Narr, 2009, S. 53–59.

2 Konzept und Entwicklung der *stand off*-Annotation

Um Texte über eine Abfrageschnittstelle quantitativ und qualitativ auswerten zu können, müssen sie zuerst annotiert werden. Annotation bezeichnet dabei die Beifügung von Metadaten zu einer definierten Annotationsbasis. Ferner unterscheidet man bei der Textauszeichnung zwischen Header-Annotation, die z. B. bibliographische Metadaten zum (Gesamt-)Text enthält, struktureller Annotation, also der Auszeichnung von physischer und logischer Textstruktur und positioneller Annotation zur inhaltlichen Auszeichnung der einzelnen Annotationseinheiten. In der Vergangenheit wurde die Textauszeichnung vor allem durch so genannte *inline*-Annotationen realisiert. Bei der *inline*-Annotation bilden Primärtext und Annotation eine Einheit und werden in ein und derselben Datei gespeichert. Dieses Vorgehen birgt jedoch Unzulänglichkeiten: So wird durch die direkte Annotation der Primärtext in gewisser Weise zerstört, zumindest aber manipuliert. Dies ist vor allem dann problematisch, wenn der Text nicht frei verfügbar ist, sondern beispielsweise nur online oder auf einem ROM-Medium (*read only memory*) vorliegt. Die Lesbarkeit des Originaltextes nimmt zunehmend ab, je intensiver das Dokument mit Metadaten z. B. im XML-Format annotiert wird. Der größte Nachteil des *inline*-Ansatzes wird jedoch in den stark begrenzten Möglichkeiten zur Annotation von Überschneidungen und konkurrierenden Hierarchien deutlich. Die parallele Annotation eines Textes auf mehreren Annotationsebenen ist mit dem *inline*-Ansatz kaum zu bewerkstelligen.

Mit dem *stand off*-Konzept wird eine strikte logische Trennung von Primär- und Sekundärdaten gefordert (Thompson & McKelvie, 1997; Dipper, 2005; Rodríguez *et al.*, 2007). Diese Trennung beseitigt die oben genannten Einschränkungen der *inline*-Annotation weitestgehend. Der Originaltext bleibt unverändert, da er noch vor der eigentlichen Annotation auf Zeichen- oder Wortebene indexiert wird, um so über externe Annotationsdateien referenzierbar zu sein (Dybkjær & Bernsen, 2000). Überschneidungen und Mehrfachannotationen können über diesen Referenzierungsmechanismus ebenso realisiert werden, wie die nachträgliche Hinzufügung oder Löschung von Annotationsebenen. Ein „Hund" kann gleichzeitig als Substantiv und als Säugetier annotiert werden. Die zwei entsprechenden Annotationsebenen könnten „Wortart" und „Ontologie" lauten. Für jede neue Ebene wird eine weitere Annotationsdatei angelegt. Im Falle einer Löschung wird einfach die Datei mitsamt der Referenzierungen gelöscht. Probleme ergeben sich bei dieser Art der Annotation lediglich, wenn der Originaltext nachträglich geändert wird, da auf diese Weise der Index durcheinander gerät. Nach derartigen Änderungen muss der Primärtext in jedem Fall mit bereits erstellten Annotationen synchronisiert werden.

3 XML und stand off-Annotation

Obwohl sich die *stand off*-Formate verschiedener Projekte teilweise stark voneinander unterscheiden, wird in fast allen Fällen die Ebene der physischen Datenstruktur (Dipper *et al.*, 2007) mit Hilfe der *eXtensible Markup Language* (XML) realisiert. Die Idee hinter XML ist es, die implizite Struktur eines Textdokuments durch das Hinzufügen von strukturell-beschreibenden Markuptags zu explizieren. So ist es möglich, die Struktur eines Textes unabhängig von seiner physischen Erscheinung, wie etwa Fettschrift oder Kursivschrift, zu definieren (Bryan, 1992). Diese Markuptags werden bei der *inline*-Annotation, die auch heute noch vor allem für die Repräsentation hierarchischer Baumstrukturen eingesetzt wird, verwendet, um beliebige Informationen wie etwa *Parts of Speech* (PoS) oder Satzarten direkt in den Primärtext zu annotieren. Beim *stand off*-Ansatz werden meist die Möglichkeiten der XML-Techologien *XPointer* und *XLink* (DeRose *et al.*, 2002, 2001) genutzt, um die Annotationen über Referenzen vom Originaltext zu trennen. Mit *XLink* ist es möglich,

über die Attribute eines Elements uni- und multidirektionale Links in XML-Dokumenten definieren. Mit der Anfragesprache *XML Pointer Language* können darüber hinaus bestimmte Teile eines XML-Dokuments referenziert werden, indem die entsprechenden Knoten in der XML-Baumstruktur adressiert werden.

Die *TEI Standoff Markup Workgroup*[1] empfiehlt zur Erstellung von *stand off*-Inhalten *XML Includes*, eine weitere XML-Technologie, die es ermöglicht, innerhalb von XML-Dokumenten auf Teile anderer Dokumente zu verweisen. Aufgrund des hohen Verbreitungsgrades des XML-Standards als Instrument zur Modellierung von strukturierter Information (Lobin, 1998), und aufgrund der Verfügbarkeit von Mechanismen wie *XLink*, *XPointer* und *XML Include* sollte XML auch als Grundlage für jegliche Standardisierungsbestrebungen im Bereich des *stand off*-Markup herangezogen werden. XML stellt deshalb den kleinsten gemeinsamen Nenner der untersuchten *stand off*-Formate dar.

4 Implementierungsansätze des stand off-Konzepts

Nachfolgend vergleichen wir unterschiedliche Implementierungen des *stand off*-Konzepts auf Basis von XML. Hierfür werden exemplarisch die *stand off*-Formate ausgewählter Textannotationswerkzeuge analysiert, die alle die parallele Annotation auf mehreren Ebenen unterstützen. In einer projektbezogenen Vorstudie zur Eignung von Annotationswerkzeugen für diachrone Korpora wurde unter anderem das Kriterium „Unterstützung von *stand off*-Annotation" untersucht. Die vier Tools *Callisto*[2], *GATE*[3], *MMAX2*[4] und *UAMCorpusTool*[5] fielen bei der Evaluation durch ihre teilweise stark divergierenden Umsetzungen des *stand off*-Konzepts auf. In diesem Abschnitt sollen deshalb die *stand off*-Annotationsformate der vier oben genannten Tools anhand folgender Parameter verglichen werden:

(a) Speicherung und Konservierung des Originaltextes

(b) Synchronisierungsmechanismen bei Änderungen im Originaltext

(c) Indexierung und Tokenisierung des Originaltexts

(d) Realisierung der logischen Trennung von Originaltext und Annotation

4.1 Speicherung und Konservierung des Originaltextes

Nur beim *UAMCorpusTool* wird der Originaltext ohne jegliche Manipulation in einem separaten Ordner gespeichert und so für spätere Anwendungen konserviert. *Callisto* speichert den Originaltext als Base64-kodierten Signalstrom ab. Base64 beschreibt ein Verfahren bei dem Daten als ASCII-Zeichenstrom kodiert werden (Josefsson, 2006). Die Entwickler greifen auf diesen Mechanismus zurück, um unerwünschte Zeilenumbrüche, welche bei der Portierung von Texten zwischen UNIX- und PC-Systemen entstehen können, zu umgehen, da dies die eindeutige Referenzierung des Originaltextes unmöglich macht. Durch die Base64-Verschlüsselung kann der Zeichenstrom auf beiden Systemen konsistent dargestellt werden. Nachteile dieser Lösung sind eine Zunahme der Dateigröße, sowie der völlige Verlust der Lesbarkeit des Originaltextes ohne entsprechende Decodersoftware.

[1] `http://www.tei-c.org/Activities/Workgroups/SO/`, Zugriff Juli 2009
[2] `http://callisto.mitre.org/`, Zugriff Juli 2009 .
[3] `http://gate.ac.uk/index.html`, Zugriff Juli 2009
[4] `http://www.eml-research.de/english/research/nlp/download/mmax.php`, Zugriff Juli 2009
[5] `http://wagsoft.com/CorpusTool/index.html`, Zugriff Juli 2009

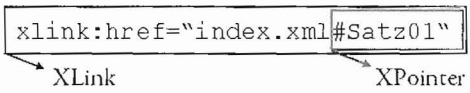

Abbildung 1: Zusammenspiel von XLink und XPointer

Bei den *stand off*-Formaten von *GATE* und *MMAX2* (Müller, 2005) wird der Originaltext insofern manipuliert, als die Indizes direkt in die Primärdaten geschrieben werden. Dies beeinträchtigt nicht nur die Lesbarkeit des Textes, sondern verstößt auch gegen die grundlegende Forderung nach Unversehrtheit der Originaldaten.

4.2 Synchronisierungsmechanismen bei Änderungen im Originaltext

Bei zwei von vier Formaten beinhaltet das Annotationswerkzeug Synchronisierungsmechanismen, die auch nachträgliche Änderungen am Originaltext erlauben. *GATE* und *MMAX2* ermöglichen die Korrektur orthografischer Fehler im Originaltext während des laufenden Annotationsprozesses. Der geänderte Index wird über das Annotationstool mit den bisherigen Annotationen synchronisiert. Die Software von *Callisto* und *UAMCorpusTool* unterstützt eine solche Synchronisierung nicht. Änderungen im Originaltext machen alle vorherigen Annotationen zu diesem Text unbrauchbar.

4.3 Indexierung und Tokenisierung des Originaltexts

Bis auf das *MMAX2*-Tool wird der Primärtext bei allen anderen Werkzeugen zeichenweise zerlegt. Die Annotationseinheiten werden dann durch zwei Zahlen beschrieben, welche die Start- und Endpunkte der jeweiligen Einheit im laufenden Zeichenstrom beschreiben. Beim Format von *MMAX2* kann über eine grafische Oberfläche ein Tokenisierer konfiguriert werden, über welchen sich die Grenzen zwischen den Annotationseinheiten beliebig feinkörnig bestimmen lassen. Will man einen Text beispielsweise nur hinsichtlich seiner Wortarten annotieren, so kann man über *white space* und Satzzeichen den Tokenizer anweisen, den Text wortweise zu zerlegen. Dies hat zwar den Vorteil, dass während des Annotationsprozesses die gewünschten Annotationseinheiten mit einem Klick selektiert werden können, birgt aber Probleme wenn man beispielsweise nachträglich den Text noch auf Morphemebene annotieren möchte. In diesem Fall muss der Text nochmals neu tokenisiert und mit bereits vorhandenen Annotationen synchronisiert werden.

4.4 Realisierung der logischen Trennung von Originaltext und Annotation

Große Unterschiede zeigen sich bei der Implementierung der Trennung von Originaltext und Annotation. *Callisto* und *GATE* trennen Primär- und Sekundärdaten zwar logisch voneinander, speichern die Daten allerdings in ein und derselben Datei. Dabei stehen die Originaldaten in einer Art Header-Bereich, die Annotationen im Body-Bereich. Bei *MMAX2* und *UAMCorpusTool* werden Originaltext und Annotation auch physisch rigoros getrennt und in unterschiedlichen Dateien gespeichert. Der Vorteil dieser konsequenten Trennung liegt in der Konservierung des Originaltextes, welche bei *MMAX2* wegen der *Inline*-Annotation der Indizes trotzdem verletzt wird. Der Nachteil liegt im Verwaltungsaufwand der einzelnen Dateien. Mit jeder neuen Annotationsebene kommt eine weitere Datei hinzu. Allerdings fördert die dateiweise Trennung der Annotation alles in allem die Lesbarkeit.

Tabelle 1: Indexierung und Referenzierung beim UAMCorpusTool

Das also war des Pudels...																							
D	a	s		a	l	s	o		w	a	r		d	e	s		P	u	d	e	l	s
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
start: 1 end: 3				start: 5 end: 8					start: 10 end: 12				start: 14 end: 16				start: 18 end: 23					
<segment id='1' start='1' end='3' features='POS;ART' state='active'/>																							

5 Empfehlungen für ein standardisiertes stand off-Format

Die Untersuchung der Realisierungsformen von *stand off*-Formaten bei Annotationswerkzeugen macht deutlich, dass das Konzept einer *stand off*-Annotation auf unterschiedlichste Weise interpretiert und umgesetzt wird. Dabei scheinen einige Implementierungen Vorteile gegenüber anderen Ansätzen zu haben. In diesem Teil der Studie werden die besten Implementierungen der einzelnen Untersuchungsparameter zu einer kurzen Empfehlungsliste für künftige Standardisierungsansätze auf dem Gebiet der *stand off*-Annotation zusammengefasst:

(a) Der Originaltext sollte in seinem ursprünglichen Zustand im Dateisystem der Annotationsdatei gesichert werden, um die Lesbarkeit und Wiederverwendbarkeit zu gewährleisten.

(b) Die Indexierung des Originaltextes sollte in einer gesonderten Datei gespeichert werden.

(c) Die Annotationssoftware, die das stand off-Format generiert, sollte Synchronisierungsmechanismen enthalten, die es erlauben den Originaltext auch während des laufenden Annotationsprozesses zu ändern.

(d) Die Software sollte Versionskontrolle und Änderungshistorie der Primärdaten unterstützen.

(e) Bei der Indexierung sollte der Text am besten zeichenweise erfasst werden, da so später beliebig feinkörnige Annotationen hinzugefügt werden können.

(f) Die Speicherung von Originaltext und Annotation in unterschiedlichen Dateien erhöht die Lesbarkeit und ermöglicht die Konservierung der Primärdaten.

6 Von der Implementierung zur Standardisierung?

Bei den untersuchten *stand off*-Formaten handelt es sich durchweg um konkrete Implementierungen innerhalb eines Annotationswerkzeugs. Ein standardisiertes *stand off*-Format sollte jedoch als Meta-Format konzipiert werden, welches innerhalb eines definierten Rahmens und unter Berücksichtigung der oben formulierten Empfehlungen verschiedene Implementierungen erlaubt (Dipper *et al.*, 2007). Eine grundlegende Forderung in Hinblick auf ein standardisiertes Format ist somit die Entflechtung von Annotationswerkzeugen und *stand off*-Formaten. In diesem Bereich wurden mit dem *stand off*-Format PAULA[6] (*Potsdam Interchange Format for Linguistic Annotation*), einem generisches Format, das *stand off*-Annotationen verschiedener Annotationswerkzeuge vereinheitlichen kann und

[6] `http://www.sfb632.uni-potsdam.de/~d1/paula/doc/index.html`, Zugriff Juli 2009

für eine große Datenbank namens ANNIS (*ANNotation of Information Structure*) verfügbar macht, erste wichtige Schritte auf dem Weg zu einer weiterreichenden *stand off*-Standardisierung unternommen. Das Hauptproblem bei der Entwicklung eines *stand-off*-Annotationsstandards dürfte eher die Schnittstelle zu bestehenden Datenbanken und Korpora sowie die Vielzahl an unterschiedlichen Annotationsformaten darstellen. Wenn man also ein Format schaffen möchte – egal ob nun *stand off* oder *inline* – so muss dieses Format nicht nur verschiedene Annotationswerkzeuge unterstützen, sondern auch eine Schnittstelle zu bereits bestehenden Annotationen schaffen. Mit GrAF (Ide & Suderman, 2007) wird ein solch generisches Austauschformat auf der Basis von Graphen beschrieben. GrAF ensteht im Rahmen des *Linguistic Annotation Framework* (LAF), einem großangelegten Standardisierungsprojekt, welches durch die Zusammenführung einzelner Teilstandards (TEI, CES, XCES, etc.) einen internationalen Standard für die Erstellung, Annotation und Manipulation von linguistischen Daten definiert. Aufgrund des äußerst heterogenen Feldes an Annotationswerkzeugen und Formaten scheint ein standardisiertes Format in naher Zukunft wenig realistisch. Allerdings könnte eine großangelegtes Projekt wie das LAF, welches der Forschungsgruppe ISO/TC37/SC4 (Normierung von Sprachressourcen) angehört, wohl am ehesten den Annotationsbereich homogenisieren. Um vorläufig zumindest ein Mindestmaß an Benutzbarkeit, Qualität und Konsistenz konkreter *stand off*-Implementierungen zu gewährleisten, sollten bei der Konzeption neuer Formate die unter Punkt 5 formulierten Empfehlungen berücksichtigt werden.

Literatur

Bryan, M. (1992). *SGML. An Author's Guide to the Standard Generalized Markup Language.* Addison-Wesley, Bonn.

DeRose, S., Maler, E., & Orchard, D. (2001). XML Linking Language (XLink) Version 1.0. W3C Recommendation, June 27, 2001.

DeRose, S., Jr., R. D., Grosso, P., Maler, E., Marsh, J., & Walsh, N. (2002). XML Pointer Language (XPointer). W3C Working Draft, August 16, 2002.

Dipper, S. (2005). XML-based Stand off-Representation and Exploitation of Multi-Level Linguistic Annotation. In *Proceedings of Berliner XML Tage 2005 (BXML 2005)*, pages 39–50.

Dipper, S., Götze, M., Küssner, U., & Stede, M. (2007). Representing and Querying Standoff XML. In G. Rehm, A. Witt, & L. Lemnitzer, editors, *Data structures for linguistic resources and applications. Proceedings of the Biennial GLDV Conference 2007*, pages 337–346, Tübingen. Narr.

Dybkjær, L. & Bernsen, N. O. (2000). The MATE markup framework. In *Annual meeting of the ACL, Proceedings of the 1st SIGdial Workshop on Discourse and Dialogue*, pages 19 –28, Morristown/NJ. Association for Computational Linguistics, Association for Computational Linguistics.

Ide, N. & Suderman, K. (2007). GrAF: A Graph-based Format for Linguistic Annotations. In *Proceedings Linguistic Annotation Workshop held in conjunction with ACL 2007*, pages 1–8, Morristown/NJ. Association for Computational Linguistics, Association for Computational Linguistics.

Josefsson, S. (2006). RFC4648. The Base16, Base32, and Base64 Data Encodings (proposed standard). `http://tools.ietf.org/html/rfc4648`, `http://www.rfc-editor.org/rfc/rfc4648.txt`. Internet Engineering Task Force (IETF), RFC4648, Fremont/CA, Oktober 2006.

Lobin, H. (1998). *Informationsmodellierung in XML und SGML.* Springer, Heidelberg / Berlin.

Müller, C. (2005). A flexible stand off-data model with query language for multi-level annotation. In *Proceedings of the ACL Interactive Poster and Demonstration Sessions*, pages 109–112, Morristown/NJ. Association for Computational Linguistics.

Rodríguez, K. J., Dipper, S., Götze, M., Poesio, M., Riccardi, G., Raymond, C., & Rabiega-Wiśniewska, J. (2007). Standoff Coordination for Multi-Tool Annotation in a Dialogue Corpus. In *Proceedings of the ACL Linguistic Annotation Workshop*, pages 148–155, Morristown/NJ. Association for Computational Linguistics.

Thompson, H. S. & McKelvie, D. (1997). Hyperlink semantics for standoff markup of read-only documents. In *Proceedings SGML Europe 1997*.