Einstieg in Text Analytics für SAS Enterprise Guide Anwender: Von der Datenquelle zum Bericht - und darüber hinaus...

Johannes Lang
HMS Analytical Software
Rohrbacher Straße 26
69115 Heidelberg
johannes.lang@analytical-software.de

Zusammenfassung

Immer mehr Firmen erkennen das Potential ihrer unstrukturierten Daten in Textform, welche meist nicht in bestehende analytische Umgebungen (z.B. Data Warehouse) integriert sind. Mithilfe von Text Analytics Verfahren ist es möglich, wertvolle Informationen aus Kundenemails, Call-Center-Notizen oder Vertragsdokumenten automatisch aufzubereiten und in eine Analyseplattform zu integrieren – auf Wunsch mit Berücksichtigung von Informationen aus dem Web. Allein – wie fängt man damit an?
Um den Einstieg für Interessierte zu erleichtern, wird anhand des intuitiven Anwenderwerkzeugs SAS Enterprise Guide gezeigt, wie man unstrukturierte Daten einliest und vorverarbeitet, so dass anschließend einfache tabellarische und grafische Auswertungen möglich sind. Es wird auch gezeigt, wie fehlende Funktionalität in Form von externen Java-Bibliotheken eingebunden werden kann.

Schlüsselwörter: Text Analytics, Text Mining, SAS Enterprise Guide, Java

1 Einführung: Warum Text Analytics heute wichtiger ist als jemals zuvor

Immer mehr Firmen erkennen das Potential ihrer unstrukturierten Daten in Textform, welche meist nicht in bestehende analytische Umgebungen (z.B. Data Warehouse) integriert sind. Viel Aufwand wurde in die Erfassung und Speicherung von textueller Information investiert, in Form von E-Mails, Call-Center-Notizen oder auch Vertragsdokumenten. Darüber hinaus werden textuelle Informationen aus dem Internet (z.B. Produktbewertungen, Kundenprofile in sozialen Netzwerken) immer wichtiger für ein umfassendes Customer Relationship Management (CRM), das viele Unternehmen anstreben.
Daher stellt sich immer drängender die Frage, wie diese Informationen maschinell gesammelt und ausgewertet werden können. Während Tabellen vergleichsweise einfach mit SQL abgefragt werden können, ist dies bei Textdaten nicht so ohne weiteres möglich. Viel manueller Aufwand ist notwendig, da die wichtigen Daten alle gelesen und analysiert werden müssen. Hier kann Text Analytics helfen, um den menschlichen In-

formationsverarbeiter effektiv durch Filterung und Aufbereitung zu unterstützen, damit mehr wertvolle Analysezeit am Ende des Tages übrigbleibt.

2 Worum geht es bei Text Analytics?

2.1 Ziele und Anforderungen

Bei Text Analytics geht es darum, aus textuellen Daten mit Hilfe von Software aussagekräftige Informationen zu gewinnen, um bessere Geschäftsentscheidungen treffen zu können. Ein gutes System sollte dem Anwender breite Möglichkeiten der Informationsextraktion bieten, sowie die Integration von Textdaten in strukturierte Analyseumgebungen unterstützen. Wichtig ist auch die interaktive, intuitive Visualisierung der Textdaten, um Recherche und Analyse gleichermaßen zu fördern.
Oft wird Text Analytics synonym mit Text Mining verwendet, eine Disziplin welche besondere statistische und linguistische Methoden einsetzt, um möglichst gute Klassifikationsmodelle für textuelle Daten zu finden. In diesem Beitrag soll jedoch der Fokus auf der pragmatischen, fachlich orientierten Herangehensweise bei der Auswertung von Textdaten liegen, um interessierten Anwendern einen Einstieg in das Thema anzubieten.

2.2 Drei Ansätze zum Umgang mit unstrukturierten Daten

Viele Unternehmen haben so genannte Enterprise Content Management (ECM) Systeme eingeführt, um ihre Textdaten zu konsolidieren und für die spätere Verwendung abzuspeichern. Solche Systeme bieten oftmals einfache Suchmöglichkeiten, wie Volltextsuche mit Wildcards (z.B. „Kunde*“ findet alle Wörter die mit „Kunde“ beginnen). Tiefergehende Suchanfragen wie „Finde alle Dokumente, in denen einer unserer Kunden namentlich erwähnt wird“ werden in der Regel nicht unterstützt.
Diesem einfachen ECM-Ansatz steht der Einsatz von profunden computerlinguistischen Methoden gegenüber, die zum Ziel haben, die Bedeutung der Sprache in den Textdaten semantisch korrekt zu erfassen. Dafür ist umfassendes Kontextwissen (sprachlich und bezogen auf die jeweilige Domäne) notwendig, um die jeder Sprache inhärenten Mehrdeutigkeiten im jeweiligen Text auflösen zu können. So ist beispielsweise das Wort „Bank“ in den Nachrichten einer Wirtschaftszeitung eher als „Geldinstitut“ zu interpretieren, und nicht als „Sitzgelegenheit“, was für einen Computer im Gegensatz zum menschlichen Leser nicht ohne weiteres klar ist. Für solche und andere Aufgaben ist spezielles Expertenwissen über Methoden des so genannten Natural Language Processing (NLP) nötig, das an die jeweilige Domäne angepasst werden muss. Viele Firmen schrecken vor dieser Investition zurück, und bleiben bei der einfachen Volltextsuche in den textuellen Rohdaten.
Text Analytics bietet nun einen eigenen, dritten Ansatz, um mit den unstrukturierten Textdaten umzugehen[1]. Die Grundannahme, dass Wörter als Datenelemente (Token) betrachtet werden können, die auch ohne Kontext verstanden werden können, bietet ei-

[1] Der hier vorgestellte Ansatz orientiert sich an Inmon u. Nesavic (2007).

nen viel direkteren Zugang zur fachlich motivierten Auswertung von Texten. Wenn es gelingt, die Textdaten in eine strukturierte Analyseumgebung (z.B. ein Datenbankschema) zu überführen, dann sind ähnliche Analysen wie auf Tabellen mit numerischen Kennzahlen möglich. Dabei verbaut man sich nicht den Einsatz der oben erwähnten NLP-Methoden: Diese können nachträglich immer noch nachgerüstet werden, durch Eigenentwicklung, Zukauf oder durch Einsatz eines entsprechend ausgestatteten Analysetools.

3 Wie erstelle ich eine Text Analytics Datenbasis?

3.1 Überblick über die Gesamtarchitektur eines Text Analytics Systems

Ein Text Analytics System besteht in der Regel aus mehreren Komponenten, die denen eines herkömmlichen Data Warehouse ähneln[2]: Ausgehend von mehreren Quellsystemen werden Daten mittels Datenintegrationsprozessen in einen konsolidierten Datenbestand überführt. Besonders hierbei ist, dass bei der Verwendung von Web-Datenquellen ein so genannter Crawling-Mechanismus beteiligt ist, der (in der Regel turnusmäßig) Daten aus dem Web holt und für die Offline-Verarbeitung speichert. Außerdem kommen spezielle Filter- und Kategorisierungsverfahren zum Einsatz, um die textuellen Rohdaten aufzubereiten.
Diese ETL-Schicht (Extraction-Transformation-Load) bildet den Kern des Systems. Die so integrierten Textdaten werden dann über eine verdichtete Sicht nachfolgenden Komponenten bereitgestellt, ähnlich eines OLAP-Würfels mit unterschiedlichen Dimensionen und Hierarchien. Erst auf dieser Basis werden dann Text Mining Algorithmen angewendet, um beispielsweise Klassifikationsmodelle zu erstellen und zu trainieren. Der Anwender des Systems kann mit einer grafischen Oberfläche durch die verdichteten Daten navigieren, suchen und filtern. Wichtig dabei ist, dass die Textdaten entsprechend aufbereitet werden, so dass interaktive Recherche und Analyse gleichsam möglich sind.

3.2 Auswahl der Software-Plattform

Bevor konkrete fachliche Fragestellungen anhand von Textdaten untersucht werden können, sollte eine geeignete Text Analytics Software-Plattform ausgewählt und installiert werden. Diese Entscheidung ist wichtig, da hierbei die Weichen für die späteren Anwendungsmöglichkeiten gestellt werden. Es empfiehlt sich wie beim Data Warehousing ein hybrider Top-Down-/Bottom-Up-Ansatz, bei dem ausgehend von konkreten fachlichen Anforderungen (Top-Down) und konkreten Datenquellen (Bottom-Up) eine End-to-End-Lösung konzipiert wird, die nachträgliche Erweiterungen unterstützt.

[2] Siehe Feldman u. Sanger (2006).

3.3 Bildung des Analyse-Korpus

Um eine fachliche Fragestellung mit einem Text Analytics Projekt zielführend bearbeiten zu können, muss der Rahmen der zu betrachtenden Daten klar abgesteckt werden. Unscharfe Anforderungen wie „Alle Webseiten zu gentechnisch veränderten Pflanzen überwachen“ müssen daher konkretisiert werden, z.B. zur Formulierung „Täglicher Abgleich der Datenbank mit den Webseiten x, y inklusive Unterseiten der ersten und zweiten Stufe“.
Auch bei unternehmensinternen Daten wie z.B. E-Mails oder Projektdokumenten empfiehlt sich die Bildung eines Korpus, um die Aussagekraft der späteren Ergebnisse in einen Zusammenhang einordnen zu können.

3.4 Konvertierung von Dateiformaten

Falls das gebildete Korpus Dateien unterschiedlicher Kodierung (z.B. Textdateien in ANSI, UTF-8, bzw. Dateiformate wie Word, PDF) umfasst, ist ein Konvertierungsschritt sinnvoll, der diese unterschiedlichen Formate vereinheitlicht. Dies spielt insbesondere bei mehrsprachigen Korpora eine Rolle, da sprachspezifische Sonderzeichen (z.B. deutsche Umlaute, französische Akzente) in manchen Kodierungen nicht oder nicht einheitlich repräsentiert sind. Empfehlenswert ist die UTF-8-Kodierung, da diese (zumindest für den europäischen Sprachraum) ein einheitliches, ausreichend mächtiges Format darstellt.

3.5 Das Problem der Worthäufigkeiten

Die wahrscheinlich einfachste Auswertung auf einem Textkorpus ist die Auszählung der Worthäufigkeiten. Dazu müssen die Textdaten aber erst in einzelne Wörter (Token) aufgeteilt werden, man spricht hierbei von Tokenisierung. Dies kann mit den Mitteln einer Programmiersprache durchgeführt werden (z.B. SAS Base, Perl, Java), oder mit den Bordmitteln eines eingesetzten Text Analytics Tools (hierzu kann auch ein Texteditor wie Notepad++ für Windows zählen[3]). Gängige Trennzeichen (Komma, Punkt, Anführungszeichen u.a.) sollten in einer separaten Liste gepflegt werden, damit sie nachträglich anpassbar sind. Die Trennzeichen sollten als eigene Token gezählt werden, damit sie später bei Bedarf einfach ausgefiltert werden können.
Wird nun auf den tokenisierten Textdaten eine Häufigkeitszählung durchgeführt, so ist zu beachten, dass je nach Sprache und Domäne zwischen relevanten und signifikanten Token unterschieden werden muss. Ein naives Top-5-Token-Ranking für ein deutschsprachiges Korpus könnte sonst folgendes wenig aussagekräftiges Ergebnis liefern:

1. die
2. das
3. in
4. den
5. Das

[3] Siehe http://notepad-plus-plus.org/.

Damit solche (zumindest für die meisten Fragestellungen) irrelevanten Token nicht irrtümlich als relevant bewertet werden, sollten so genannte Stopwort-Listen eingesetzt werden, wie im folgenden Abschnitt beschrieben wird.
Nicht betrachtet wird an dieser Stelle die Frage, inwieweit der Inhalt eines Dokumentes eigentlich durch Worthäufigkeiten repräsentiert werden kann. Es soll hier lediglich gezeigt werden, wie maschinelle Analysemethoden auf Textdaten angewendet werden können.

3.6 Filterung mit Stop- und Startwortlisten

Um das zuvor beschriebene Problem der irrelevanten Token zu lösen, werden Listen von so genannten Stopwörtern eingesetzt, die solche Token enthalten, die vor der Häufigkeitszählung ausgefiltert werden sollen. Für viele Sprachen sind bereits vorgefertigte Listen verfügbar, die einfach eingelesen werden können. Auch hier ist es wichtig, dass die Liste nachträglich noch angepasst werden kann. Falls aus fachlicher Sicht nichts dagegen spricht, sollten alle eingelesenen Token (und auch die Token in den gepflegten Listen) in Kleinbuchstaben (Lowercase) konvertiert werden, damit Token unterschiedlicher Groß-/Kleinschreibung automatisch zusammengefasst werden können.
Der umgekehrte Ansatz so genannter Startwortlisten (oder auch Schlüsselwortlisten) ist ebenfalls üblich, wenn die fachliche Fragestellung so konkret ist, dass eine Positivliste von relevanten Token erstellt werden kann. Alle übrigen Token werden dann ausgefiltert bzw. nicht betrachtet. Ein Beispiel wäre die gezielte Suche nach den firmeneigenen Produktbezeichnungen im Korpus: Diese könnten dann (mit ihren unterschiedlichen Schreibweisen!) in einer solchen Startwortliste hinterlegt werden.

3.7 Einsatz eines Stemmers zur Wortstammreduktion

Oft ist es zusätzlich zur Filterung mittels einer Stopwortliste sinnvoll, mehrere ähnliche Token zusammenzufassen, da sie sich vom selben Wortstamm ableiten. Besonders in der deutschen Sprache gibt es z.B. bei Substantiven viele Beugungsformen, je nachdem in welcher grammatikalischen Funktion das Wort im Satz vorkommt. Gegeben sei folgende Token-Liste:

1. Ärzte
2. Arzt
3. Ärzten
4. Doktor
5. Doktors

Für die meisten menschlichen Analysten dürfte klar sein, dass die Token 1,2 und 3 auf dasselbe Grundwort „Arzt“ zurückzuführen sind, ebenso wie die Token 4 und 5 auf „Doktor“. Mit Hilfe eines Softwareprogramms zur Wortstammreduktion (sog. Stemmer) können diese drei Token auch tatsächlich auf ein gemeinsames „Einheitstoken“ abgebildet werden, so dass sie nicht als unterschiedliche Token gezählt werden. Es gibt unterschiedliche Stemming-Algorithmen, deren Implementierungen oft frei verfügbar sind

(Beispiel: Porter-Stemming-Algorithmus[4], Implementierung in SAS Base verfügbar). Solche Stemmer sind in der Regel sprachabhängig, und sollten daher erst eingesetzt werden wenn die Textsprache festgelegt ist. Zudem liefern Stemmer nicht unbedingt immer das Grundwort eines Token, sondern eben nur den Wortstamm (Beispiel: „Umsätzen“ wird zu „Umsätz“). Linguistisch aufwändiger ist die Ermittlung des Grundwortes, hierfür wird ein so genannter Lemmatisierer benötigt (dieser würde für voriges Beispiel das Grundwort „Umsatz“ liefern). Oftmals genügt aber der Einsatz eines Stemmers, da der Wortstamm aussagekräftig genug ist.

3.8 Erstellung von Synonym- und Homographlisten

Eine weitere Methode, um Token zusammenzufassen, ist die Bildung einer Synonymliste. Hierbei werden Token mit gleicher oder sehr ähnlicher Bedeutung zusammengefasst, so dass sie für die Analyse als ein Token behandelt werden können. Eine solche Liste – besser: Tabelle – kann entweder manuell erstellt, von einer externen Quelle eingebunden, oder bezogen auf das Korpus eigens erzeugt werden. Wird die Liste manuell erstellt, so kann folgende Struktur verwendet werden:

Tabelle 1: Beispiel für die Struktur einer manuell gepflegten Synonymliste

Token	**Synonym1**	**Synonym2**	**Synonym3**
arzt	doktor	facharzt	chirurg
…	…	…	…

Eine frei verfügbare Synonymliste für die deutsche Sprache findet sich im Internet (http://www.openthesaurus.de). Manche Text Analytics Tools (z.B. SAS Text Miner) können eine solche Liste dynamisch erzeugen.
Neben den Synonymen gibt es noch so genannte Homographen, dies sind Wörter mit gleicher Schreibweise, aber unterschiedlicher Bedeutung (z.B. „Bank“(Sitzgelegenheit) vs. „Bank“(Geldinstitut)). Diese unterschiedlichen Bedeutungen können mit Hilfe der jeweiligen Synonyme (Sitzgelegenheit bzw. Geldinstitut) in einer separaten Liste gepflegt werden, zusammen mit weiteren Wörtern welche im Zusammenhang der jeweiligen Bedeutung stehen (Park, öffentlich bzw. Finanzen, Krise). Dadurch kann eine Komponente im Text Analytics Prozess die wahrscheinlichste Bedeutung eines Wortes in den Textdaten ermitteln, man spricht dabei von Disambiguierung.

3.9 Kategorisierung der Daten

Dieser Schritt ist wichtig, um spätere Analysen gezielter durchführen zu können. Manche Unternehmen haben bereits ein Kategoriensystem entwickelt, welches dann an dieser Stelle eingebunden wird. Ebenso können aber neue Kategorien definiert werden, die den Textdaten zugeordnet werden sollen. Werden die Daten ohne vordefinierte Kategorien mittels eines Ähnlichkeitsmaßes eingeteilt, spricht man von Clusterbildung (Clustering). Es empfiehlt sich, aussagekräftige, spezifische Kategorien- bzw. Clusterbezeichnungen zu wählen, um Überschneidungen zu vermeiden.

[4] Siehe http://tartarus.org/~martin/PorterStemmer/.

3.10 Ermittlung und Anwendung von Relevanzkriterien

Sind die Textdaten kategorisiert, dann können erste fachliche Relevanzkriterien definiert bzw. angewendet werden, um die weitere Analyse weiter zu fokussieren. Umfasst das Korpus z.B. die Kategorien „Mitarbeiter“, „Produkte“, „Marktentwicklung“, und es soll eine Wettbewerbsanalyse durchgeführt werden, so kann die Kategorie „Mitarbeiter“ eventuell vernachlässigt werden, da man sich auf Produkte und Märkte konzentrieren möchte.
Bei dieser Relevanzbestimmung kommt es letztlich nicht so sehr auf die eingesetzte Technologie an, sondern vielmehr auf die möglichst klare Definition des fachlichen Anwendungsfalls, um Analysen auf einer möglichst homogenen und qualitativ hochwertigen Datenbasis durchzuführen.

3.11 Entwurf einer Datenbank für integrierte Textdaten

Um die aufbereiteten Textdaten für weitere Analysen zu speichern, sollte eine eigene Datenbank aufgesetzt werden, welche speziell für die integrierten Textdaten konzipiert ist. Im einfachsten Fall werden die Dokumente im Volltext gespeichert, plus die einzelnen Token in separaten Tabellen mit Fremdschlüsselbeziehungen auf die jeweiligen Volltext-Tabellen. So kann später leicht abgefragt werden, welches Token in welchem Dokument vorkommt.
Liegen die Textdaten bereits semi-strukturiert (z.B. in XML) vor, dann können statt einzelner Token auch größere Einheiten (so genannte Entitäten, wie z.B. <Person>Dr. Klaus Müller</Person>) in einer Tabelle abgelegt und mit dem Volltext verknüpft werden. Auch die gepflegten Tokenlisten (Stop- bzw. Startwörter, Synonyme, Homographe) werden verlinkt und in der Datenbank gespeichert.

3.12 Visualisierung von Textdaten

Um die integrierten Textdaten dem Anwender zu präsentieren, gibt es unterschiedliche Möglichkeiten. Neben der rein tabellarischen Darstellung von Token in ihrem Kontext (Keywords-In-Context, KWIC) sind für ein fachliches Analysetool weitere grafische Darstellungen wichtig. Neben einfachen Balkengraphiken gibt es viele weitere Möglichkeiten: Wort- und Konzeptgraphen können z.B. die Vernetzung von Begriffen untereinander anzeigen, werden aber schnell unübersichtlich. Tag Clouds („Begriffswolken“) helfen bei der Trenderkennung, indem häufig vorkommende Begriffe größer dargestellt sind. Eine kartographische Darstellung der Textdaten eignet sich gut zur explorativen Analyse, da die Relevanz und der Zusammenhang von Begriffen durch Farbe, Größe und Distanz dargestellt werden kann[5].

[5] Ein Beispiel hierfür sind selbstorganisierende Karten (self-organizing maps, SOMs). Hierzu und zu anderen Visualisierungsmöglichkeiten siehe z.B. Risch et al. (2008).

3.13 Iterative Systemanpassung

Wenn das System soweit ist, dass die integrierten Textdaten visualisiert werden können, dann beginnt die eigentliche Arbeit: Begriffe müssen korrigiert oder passend zugeordnet werden, und bei den Filterlisten und Relevanzkriterien besteht noch Anpassungsbedarf. Nach jeder Anpassung sollte der Datenintegrations- und Visualisierungsprozess erneut ausgeführt werden, um das Resultat beurteilen zu können.
Wie viele Iterationen benötigt werden, bis ein brauchbares Ergebnis vorliegt, hängt stark vom jeweiligen Anwendungsfall und von der Heterogenität der Daten ab.

4 Erweiterte Abfrage- und Integrationsmethoden

Wenn die Textdaten wie im vorigen Abschnitt beschrieben integriert wurden, sind erweiterte Abfragen bzw. Integrationsverfahren möglich, die hier nur kurz aufgeführt werden sollen.

4.1 Erkennung von benannten Entitäten (Named Entity Recognition)

Hierbei können beispielsweise zusammengesetzte Namen von Personen, Firmen oder Produkten als Einheit erkannt werden, indem zuvor gepflegte Look-Up-Listen mit den aufbereiteten Daten abgeglichen werden. Eine Verknüpfung mit bestehenden Kontaktdaten ist beispielsweise möglich, oder auch die Auswertung der Textdaten nach Vorkommen von Firmenkunden oder fremden Produkten.

4.2 Erkennung von Konzepten und Themen

Durch den beschriebenen Einsatz von Startwort- und Synonymlisten können automatisiert Schlagwörter für Texte vorgeschlagen werden, die den thematischen Kontext beschreiben. Dies ist essentiell für die Kategorisierung der Textdaten und für spätere Suchanfragen. Weiterhin können mehrere Token oder benannte Entitäten zu einem Konzept gruppiert werden (z.B. ‚FC Bayern München' und ‚Dynamo Dresden' zu ‚Fussballverein'), welches dann gezielt in den Daten nachverfolgt werden kann.

4.3 Erkennung von Satzbestandteilen (Part-of-Speech Tagging, Parsing)

Für Fragestellungen, bei denen mehr Information über die innere Struktur der Textdaten benötigt wird, kann der Einsatz von linguistischen Analyseverfahren weiterhelfen. Diese können beispielsweise zu jedem Token die Wortart (Substantiv, Adjektiv etc.) ermitteln, was als Part-of-Speech-Tagging bezeichnet wird. Wird darüber hinaus auch die Rolle im Satz zugeordnet (Subjekt, Prädikat, Objekt), spricht man von Syntaxanalyse oder Parsing.

5 Anwendungsfälle für Text Analytics

5.1 Analyse der Tonalität von Texten (Sentiment Analysis)

Sentiment Analysis ist ein relativ neuer Anwendungsfall für Text Analytics, der durch die zunehmende Nutzung von Produktbewertungsportalen im Web an Bedeutung gewinnt. Ziel ist es, die Tonalität (positiv, negativ, neutral) eines Textes (E-Mail, Forumseintrag) bezüglich eines Produktes oder einer Marke automatisch zu erkennen. Damit kann eine Organisation ihr Customer Relationship Management verbessern, da negative Bewertungen schneller erkannt und gezielt analysiert werden können.

5.2 Automatische Dokument-Kategorisierung

Wie bereits zuvor erwähnt, können mit Text Analytics Themen und Konzepte identifiziert werden, die in Textdaten vorkommen. Diese können für die Erstellung eines kontrollierten Kategoriensystems genutzt werden, welches anschließend automatisch auf weitere Textdaten angewendet wird. Dadurch können Arbeitsabläufe beschleunigt und Suchanfragen erleichtert werden.

5.3 Branchenspezifische Anwendungsfälle

Stellvertretend werden einige Anwendungsfälle aus den Branchen Life Science und Versicherungen genannt, da hier Schwerpunkte unserer Projektarbeit liegen.
Kliniken und Pharmaunternehmen können mit Text Analytics beispielsweise Arztberichte zu Therapieverläufen mit unterschiedlichen Medikamenten auswerten, um die Effektivität von Behandlungsmethoden besser beurteilen und vergleichen zu können. Ein weiteres Einsatzszenario ist der Aufbau einer durchsuchbaren Datenbank über Chemikalien und ihre Verwendung zur Arzneimittelherstellung, ausgehend von textuellen Einzelreports die im Rahmen unterschiedlicher Projekte erstellt wurden.
In der Krebsforschung werden heute bereits Text Analytics Verfahren eingesetzt, um Publikationen automatisiert nach genetischen Assoziationen zu durchsuchen.
In der Versicherungsbranche kann Text Analytics ebenfalls sinnvoll eingesetzt werden: Beispielsweise können Vertragsdokumente in eine Datenbank überführt werden, die nach Querbeziehungen durchsucht werden kann. Dadurch können bei Firmenfusionen die betroffenen Verträge schneller identifiziert werden, wodurch Recherchezeit gespart wird. Ebenso ist die Auswertung von internen Schadensberichten möglich, um wiederkehrende Ursachen für Zahlungsfälle sichtbar zu machen. Diese können dann bei der Gewichtung von Kostenfaktoren berücksichtigt werden.

6 Text Analytics mit SAS

6.1 Überblick über das Angebot

Das eigentliche Text Analytics Frontend von SAS ist der SAS Text Miner, der seit der Version 4.2 nicht mehr als eigenes Tool existiert, sondern in SAS Enterprise Miner integriert ist[6]. Er fügt diesem spezielle Prozessknoten und Funktionen für Text Mining hinzu, wie z.B. Text Parsing, Synonymerkennung oder Stemming. Er ist für Anwender mit starkem statistisch-mathematischen Hintergrund zugeschnitten und daher für Einsteiger in diesem Bereich nur bedingt geeignet.
Neben diesem schon länger existierenden Softwarepaket bietet SAS mehrere zusätzliche Frontends an, welche auf den Technologien der 2008 akquirierten Firma Teragram basieren[7]. Das SAS Enterprise Categorization Studio ist eine grafische Oberfläche, die es Experten ermöglicht, Metadaten wie Kategorien, Taxonomien oder Geschäftsregeln zu definieren, die dann auf Textdaten angewendet werden können, um diese z.B. gemäß der Regeln in die Kategorien einzuordnen. Im Gegensatz zum SAS Text Miner, der Metadaten auf Basis von Textdaten erzeugt (Bottom-Up), unterstützt dieses Tool den Ansatz, Metadaten manuell von Experten zu erzeugen, um dann Analysetools darauf anzuwenden (Top-Down).
Daneben gibt es SAS Sentiment Analysis Studio, das speziell auf die Analyse der Tonalität von Texten wie Kundenbewertungen von Produkten zugeschnitten ist. Das zusätzliche Frontend SAS Ontology Management Studio wird als fortgeschrittene Variante des SAS Enterprise Categorization Studio (welches nur Taxonomien, aber keine Ontologien unterstützt) positioniert. Ebenfalls neu ist das SAS Information Retrieval Studio, welches als webbasierte Oberfläche zur Suche in Textdatenbeständen dargestellt wird[8].

6.2 Motivation für den Einsatz von SAS Enterprise Guide

Bei dem beschriebenen Angebot von unterschiedlichen SAS Text Analytics Frontends, wie kommt man da auf die Idee, Text Analytics mit SAS Enterprise Guide zu versuchen?
Dieses Analysewerkzeug ist speziell für Anwender aus Fachabteilungen entworfen, die ohne tiefe SAS-Programmierkenntnisse Berichte mit Tabellen und Grafiken anhand unterschiedlicher Datenquellen erstellen möchten. Dabei sind auch einfache Datenintegrationsaufgaben ohne weiteres möglich (z.B. Tabellen verbinden und transponieren). Unsere Kunden aus unterschiedlichen Branchen, bei denen wir SAS Enterprise Guide bereits vorgestellt und eingesetzt haben, geben uns immer wieder die Rückmeldung, dass es sich hierbei um ein leicht zu bedienendes, aber mächtiges Analysewerkzeug handle.

[6] Vgl. http://www.sas.com/resources/factsheet/text-miner-factsheet1.pdf
[7] Siehe SAS Pressemitteilung von 2010: http://www.sas.com/news/preleases/text-analytics.html
[8] Vgl. SAS Produktbeschreibung unter http://www.sas.com/text-analytics/enterprise-content-categorization/add-ons/index.html#section=4

Wenn man also Fachanwender in die Lage versetzen möchte, aus Textdaten mit Hilfe von Software aussagekräftige Informationen zu gewinnen, warum sollte man dies dann nicht mit dieser Anwendung versuchen?
Während z.B. der SAS Enterprise Miner sehr viel spezielleres Datenanalyse-Know-how voraussetzt, kann mit etwas Hilfestellung eine einfache Text Analytics Auswertung auch für Nichtexperten in Form eines Enterprise Guide Projektes erstellt werden. Eine solche Hilfestellung will dieser Beitrag bieten.

7 Beispiel: Erstellung einer Text Analytics Auswertung mit SAS Enterprise Guide

7.1 Automatische Web-Korpuserstellung mit %TMFILTER

Seit der Version 9.2 liefert SAS – sofern der SAS Text Miner lizensiert ist – mit einer Foundation-Installation einige Makros und Hilfslisten für Text Analytics mit, die sich standardmäßig im Installationsordner unterhalb von SASFoundation\9.2\tmine befinden.
Mit dem Makro %tmfilter kann eine komplette Webseite inklusive aller verlinkten Dokumente über mehrere Unterebenen hinweg heruntergeladen und automatisch aufbereitet werden. Anbei ist ein Beispielaufruf dieses Makros aufgeführt, der ein Korpus von Texten über unterschiedliche Arzneimittel anhand der Web-Datenbank dbpedia.org erstellt:

```
%tmfilter (  /* URL der zu crawlenden Webseite */
             url        = http://dbpedia.org/ontology/Drug
             /* Tiefe bis zu der Links verfolgt werden */
            ,depth      = 1
           /* Ausgabeverzeichnis für gecrawlte Dateien */
            ,dir        = C:\temp\tmfilter\dir
         /* Ausgabeverzeichnis für extrahierte Textdaten */
            ,destdir    = C:\temp\tmfilter\destdir
  /* Gibt an ob nur domäneninterne Links verfolgt werden */
            ,norestrict = 1
            /* Pfad zum erzeugten Inhaltsverzeichnis */
            ,dataset    = ksfe.drugwebcrawl
         );
```

Das Ergebnis ist ein kleines Korpus aus 318 HTML-Dokumenten aus dem Internet, von denen 309 bereits automatisch in reine Textdateien ohne HTML-Formatierungen konvertiert wurden.

7.2 Einbindung einer sprachspezifischen Stopwortliste

Im Beispiel-Projekt wurde eine selbsterstellte Textdatei als Stopwortliste verwendet, die mit der Anwendungsroutine „Daten importieren“ in eine SAS-Tabelle eingelesen wurde.

Es kann jedoch auch stattdessen eine von SAS mitgelieferte Liste verwendet werden, die standardmäßig im Ordner SASFoundation\<Version>\tmine\sashelp liegt.

7.3 Anlegen einer Start- bzw. Schlüsselwortliste

Der Einfachheit halber wurde eine domänenspezifische Startwortliste (ebenfalls als Textdatei) erstellt, die gleichzeitig als Synonymliste fungiert, indem zu jedem Wort bis zu zwei Alias-Spalten gefüllt werden können. Auch diese Liste wird wie die Stopwortliste in eine SAS Tabelle eingelesen.

7.4 Anbindung eines externen Java-Tokenizers im Data-Schritt

Für die Tokenisierung wurde auf die Programmiersprache Java zurückgegriffen, da es darin standardmäßig bereits einen so genannten StringTokenizer gibt, der diese Aufgabe recht einfach löst. Der Tokenizer wurde gleich erweitert, so dass er die Tokens mit ihrem rechten und linken Kontext in eine neue Textdatei ausgibt (Keywords-In-Context-Index). Zwar könnte ein Tokenizer auch mit SAS Base implementiert werden, aber etwas weniger komfortabel. Außerdem kann hierbei das generelle Vorgehen gezeigt werden, um fehlende Funktionalität im Enterprise Guide in Form von Java-Bibliotheken nachzurüsten.
Der Java-Tokenizer wurde separat programmiert und als JAR-Bibliothek kwicgenerator.jar bereitgestellt. Um den Java-Klassenpfad so zu erweitern, dass diese JAR-Bibliothek vom SAS-Programm aus genutzt werden kann, wurden drei Makros %addtoclasspath, %initclasspathupdate, %resetclasspath im Autocall-Pfad der SAS Sitzung hinterlegt[9]:
Folgendes Makro %generateKWICTxtFile inklusive Aufruf desselben wurde direkt im Enterprise Guide erstellt und im Projekt als SAS Code eingebettet:

```
%GLOBAL
   g_sFullFilePath
   g_nLeftContextLength
   g_nRightContextLength
;

%MACRO generateKWICTxtFile ( sFullFilePath =
                            ,nLeftContextLength =
                            ,nRightContextLength =
);

%InitClasspathUpdate;

%AddToClasspath(&g_sProjectRoot./Java/kwicgenerator.jar);

DATA _NULL_;
```

[9] Der Quellcode dieser SAS-Makros ist in folgender SAS-Note enthalten:
http://support.sas.com/kb/38/518.html.
Anwendungsbeispiele für dieses Vorgehen finden sich in Mengelbier u. Skowronski (2008).

```
   ATTRIB
      sFullFilePath  LENGTH = $ 256
   ;
   sFullFilePath = "&sFullFilePath.";

   declare javaobj oKwicGenerator
       ("de/hms/kwicgenerator/controller/CKWICGenerator"
       , &nLeftContextLength.
       , &nRightContextLength.);

   oKwicGenerator.callVoidMethod
                  ("initTokenCollectionFromText"
                  , sFullFilePath);

   oKwicGenerator.callVoidMethod
                  ("writeTokenCollectionToText");

RUN;

%ResetClasspath;

%MEND generateKWICTxtFile;

%generateKWICTxtFile ( sFullFilePath = &g_sFullFilePath.
              ,nLeftContextLength = &g_nLeftContextLength.
             ,nRightContextLength = &g_nRightContextLength.
                     );
```

Mit der Data-Schritt-Anweisung declare javaobj <Variablenname> wird ein Objekt der Java-Klasse CKWICGenerator instantiiert, mit den von außen per Makrovariable übergebenen Werten für die Anzahl Zeichen des linken bzw. rechten Token-Kontextes.
Anschließend wird die Methode initTokenCollectionFromText des Objektes aufgerufen, wobei der Pfad zur einzulesenden Textdatei ebenfalls als Makrovariable übergeben wird. Durch den Aufruf der Methode writeTokenCollectionToText wird der Keywords-In-Context-Index in eine vordefinierte Textdatei geschrieben.
Die Signaturen der aufgerufenen Java-Methoden müssen bekannt sein und im SAS-Code passend aufgerufen werden, damit dieses Vorgehen funktioniert.

7.5 Erzeugung eines Keywords-In-Context (KWIC) Berichtes

Die vom Java-Tokenizer erzeugte Textdatei mit dem KWIC-Index wurde wieder in eine SAS-Tabelle eingelesen. Dann wurde mit der Anwendungsroutine „Filter und Abfrage“ eine Konvertierung aller Token in Kleinbuchstaben durchgeführt, und anschließend alle Stopwörter aus dem KWIC-Index aussortiert. Anschließend wurde die Anwendungsroutine „Listenbericht“ genutzt, um den KWIC-Index als HTML-Bericht auszugeben.

7.6 Erstellung eines Top-50-Keywords-Diagramms

Mit der Anwendungsroutine „Einfache Häufigkeiten“ wurden zusätzlich die Token-Häufigkeiten ausgezählt und absteigend sortiert in eine neue SAS-Tabelle ausgegeben. Diese Tokenliste wurde mittels der Anwendungsroutine „Filter und Abfrage“ auf die zuvor erstellte Startwortliste eingeschränkt, um nur als relevant definierte Token auszugeben. Diese Liste wurde mit der Anwendungsroutine „Balkendiagramm“ als vertikales Top-50-Keywords Diagramm dargestellt.

7.7 Export der Ergebnisse

Um die erzeugten Ergebnisse außerhalb von SAS bereitzustellen, bietet der SAS Enterprise Guide mehrere Möglichkeiten. Ein davon ist die Erstellung eines neuen HTML-Dokumentes (im Menü „Extras“ aufrufbar). Ein geführter Dialog bietet die bisher erzeugten Berichtsteile (Listenberichte, Grafiken, Häufigkeitsberichte) zur Auswahl an, die dem neuen Dokument hinzugefügt werden können. Das Ergebnis wird in einem eigenen Prozessfluss „Benutzerdefinierte Berichte“ als eigener Knoten repräsentiert. Mit der Funktion „Dokument als Schritt im Projekt exportieren“ (im Kontextmenü auf dem neuen Knoten aufrufbar) kann ein Speicherort angegeben werden, wohin die Text Analytics Auswertung gespeichert werden soll.
Damit kann die komplette Korpusanalyse automatisiert durchgeführt und exportiert werden. Dies ist sinnvoll, um bei iterativer Anpassung nicht jeden Projektschritt nochmals manuell ausführen zu müssen.

8 Zusammenfassung

In diesem Beitrag wird erklärt, wie man als Einsteiger an die maschinelle Textanalyse herangeht und wozu man sie heute verwenden kann. Wie gezeigt wurde, ist es auch mit wenigen Statistikkenntnissen möglich, eine strukturierte Datenbasis aus textuellen Rohdaten aufzubauen und einfache Analysen wie eine Top-50-Keywords Auswertung durchzuführen. Weiterführende Auswertungen wurden kurz vorgestellt, ebenso wie technische und fachliche Anwendungsfälle in unterschiedlichen Branchen.
SAS bietet mehrere unterschiedliche Tools für die Analyse von Textdaten an, deren Einsatz von der jeweiligen Vorkenntnis und Fragestellung abhängt. Als Beispiel wurde der SAS Enterprise Guide ausgewählt, da er die niedrigste Einstiegshürde bietet, um fachbezogene Auswertungen mit einer intuitiven Oberfläche zu erstellen.
Damit sollen Fachanwender bei Ihrer Arbeit mit Textdaten unterstützt werden, indem der manuelle Aufwand für Filterung und Aufbereitung verringert wird, und somit mehr wertvolle Analysezeit am Ende des Tages übrigbleibt.

Literatur

[1] Feldman, R. / Sanger, J.: The Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data, Cambridge University Press, 2006.

[2] Inmon, W. / Nesavich, A.: Tapping into Unstructured Data: Integrating Unstructured Data and Textual Analytics into Business Intelligence. Prentice Hall, Boston, 2007.

[3] Mengelbier, M. / Skowronski, J.: SAS Talking via the Java Object Interface, in: Proceedings of the SAS Global Forum 2008 Conference, Paper 027-2008, SAS Institute Inc., Cary, NC, 2008.

[4] Risch, J. et al.: Text Visualization for Visual Text Analytics, in: Lecture Notes in Computer Science, Volume 4404/2008, Springer Verlag, Berlin/Heidelberg, 2008.