Technische Universität Dresden
Fakultät Informatik
Institut für Systemarchitektur
Professur für Rechnernetze
Prof. Dr. rer. nat. habil. Dr. h. c. Alexander Schill

Diplomarbeit

zur Erlangung des akademischen Grades

Diplom-Medieninformatiker

Konzeption und Umsetzung der Benutzeroberfläche eines serviceorientierten Text Mining Systems

Autor Marcus Krejpowicz

Matrikelnr. 3233721

Abgabedatum 30.06.2011

Betreuer Dipl.-Medien-Inf. Katja Seidler

Dipl.-Inf. Sandro Reichert

Selbstständigkeitserklärung

Hiermit erkläre ich, Marcus Krejpowicz, dass ich die vorliegende Diplomarbeit ohne fremde Hilfe nur unter Verwendung der angegebenen Hilfsmittel verfasst habe.

Dresden, 30.06.2011

Danksagung

An dieser Stelle möchte ich mich bei meiner Betreuerin Katja Seidler bedanken, welche mir stets mit Rat und Tat zur Seite stand. Ein weiteres Dankeschön geht an Norman Sessler, der mir mit seiner klaren und konstruktiven Kritik sehr geholfen hat. Zudem möchte ich allen Probanden für deren Zeit und zusätzliche Anregungen danken. Ein ganz besonderer Dank gilt meinen Eltern, die mich nicht nur finanziell, sondern auch moralisch immer unterstützt und mir den Rücken gestärkt haben.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Listings

1 Einleitung

1.1 Motivation dieser Arbeit

Ein immer größer werdender Anteil unternehmensrelevanter Informationen liegt in unstrukturierten Daten wie Webseiten, Word-Dokumenten oder Blogeinträgen vor [Rus07]. Die effiziente Auswertung der in den meist textuellen Daten enthaltenen Informationen wird somit zu einem wichtigen Erfolgsfaktor für Unternehmen. Aus diesem Grund sind in den letzten Jahren eine Vielzahl von Text Mining Systemen zur computergestützten Analyse unstrukturierter Datenquellen entstanden [Pia11, Got11, HNP05]. Mit Hilfe dieser Text Mining Systeme lassen sich automatisiert strukturierte Informationen extrahieren und zu neuem, vorher nicht erkanntem Wissen verknüpfen. Entsprechend der zu analysierenden Informationen werden von den Systemen verschiedene Text Mining Aufgaben wie die automatische Klassifikation von Texten, die Erkennung von Entitäten und Relationen oder die Stichwortextraktion bereitgestellt. Seit einiger Zeit ist zu beobachten, dass neben Komplettlösungen immer mehr Webdienste entstehen, welche sich in bestehende Anwendungen einbinden lassen. Beispiele solcher Dienste sind AlchemyAPI [Orc11] oder OpenCalais [Cal11b].

Trotz der großen Anzahl an existierenden Text Mining Lösungen gibt es noch eine Vielzahl von ungelösten oder nicht ausreichend behandelten Problemfeldern. Ein Problem ist beispielsweise die unzureichende Genauigkeit und Vollständigkeit der Analyseergebnisse, was sich in falsch klassifizierten Texten oder nicht erkannten Entitäten widerspiegelt. Ein zweites Problem besteht darin, dass sich jede Text Mining Lösung auf ein spezielles Anwendungsgebiet wie zum Beispiel die Biomedizin oder das Finanzwesen beschränkt. Um diese Systeme auch für andere Anwendungsgebiete einzusetzen, bedarf es zunächst einer zeitaufwendigen Anpassung.

Ausgehend von diesen Problemen wird in [SS11] ein Ansatz für ein serviceorientiertes Text Mining System beschrieben. Die Idee des Ansatzes ist es, die Stärken bestehender Text Mining Dienste und Systemen zu kombinieren, um dadurch die genannten Probleme zu reduzieren. Die Genauigkeit der Analyseergebnisse soll dabei durch die automatische Aggregation der Ergebnisse mehrerer Text Mining Dienste erhöht werden. Des Weiteren soll das System durch die Einbindung unterschiedlicher, domänenspezifischer Text Mining Dienste für verschiedene Anwendungsbereiche eingesetzt werden können.

In der vorliegenden Arbeit soll eine einfach und intuitiv zu bedienende Benutzeroberfläche für solch ein serviceorientiertes System entwickelt und mit Hilfe eines Prototypen evaluiert werden. Die Benutzeroberfläche soll es Personen aus unterschiedlichen Fachgebieten ermöglichen, spezielle Text Mining Aufgaben wie die Erkennung von Entitäten oder die Klassifikation von Texten durchzuführen. Neben den Grundfunktionen zur Konfiguration und Durchführung einer Textanalyse sollen verschiedene Funktionen zur Bewertung und Korrektur der Ergebnisse bereitgestellt werden. Diese sollen es dem Nutzer erlauben, die automatische Auswahl und Aggregation der Dienste zu optimieren, um somit die Genauigkeit der Ergebnisse für zukünftige Textanalysen zu erhöhen.

Um eine bessere Vorstellung davon zu bekommen, für welche Aufgaben das System später eingesetzt werden kann, dient das folgende Beispielszenario.

1.2 Beispielszenario

Neben Unternehmen, die Text Mining Systeme für die Analyse von unternehmensrelvanten Daten verwenden, eignet sich das beschriebene System auch für Nachrichtenagenturen. Gerade in diesem Bereich entstehen Tag für Tag Unmengen an unstrukturierten Daten in Form von Nachrichtentexten. Mit Hilfe des serviceorientierten Systems könnten Journalisten die in den Nachrichten auftauchenden Themen automatisiert bestimmen lassen. Des Weiteren könnten die Nachrichten automatisch auf verschiedene Merkmale, wie die darin genannten Personen oder Firmen untersucht werden. Die ermittelten Informationen könnten dann genutzt werden, um relevante Zusammenhänge zwischen den einzelnen Nachrichten schneller zu erkennen. Wie an diesem Beispielszenario zu erkennen, soll das System auch von Personen bedient werden können, die sich nicht im Bereich Text Mining auskennen. Das Ziel der vorliegenden Arbeit ist es daher, sowohl für erfahrene als auch für unerfahrene Nutzer eine einfache und intuitiv zu bedienende Benutzeroberfläche zu entwerfen.

1.3 Aufbau der Arbeit

Die Arbeit ist wie folgt aufgebaut. Im ersten Teil von Kapitel 2 werden der Begriff Text Mining eingeführt und die unterschiedlichen Text Mining Aufgaben erläutert. Zusätzlich wird auf die bestehenden Herausforderungen von Textanalyseverfahren eingegangen sowie mögliche Anwendungsgebiete aufgezeigt. Der zweite Teil von Kapitel 2 stellt grundlegende Prinzipien für den Entwurf von Benutzeroberflächen vor. In Kapitel 3 erfolgt eine Betrachtung der Zielgruppe sowie der Funktionsweise des serviceorientierten Text Mining Systems. Anschließend werden die mit dem System

einhergehenden Anforderungen an die Benutzeroberfläche analysiert. Kapitel 4 zeigt anhand ausgewählter Benutzeroberflächen bestehender Text Mining Systeme verschiedene Möglichkeiten zur Umsetzung der Anforderungen. Ausgehend von den in Kapitel 3 erhobenen Anforderungen erfolgt in Kapitel 5 der Entwurf der Benutzeroberfläche. Hierfür werden auf Basis des grundlegenden Arbeitsablaufs des Nutzers alle für die Bedienung notwendigen Ansichten hergeleitet. Ausgehend von dem Entwurf wurde ein Prototyp zur Evaluation umgesetzt. Kapitel 6 geht auf wichtige Kernpunkte der Implementierung des umgesetzten Prototyps ein. Hierzu zählen beispielsweise die verwendeten Technologien sowie die Architektur. Die mit Hilfe des Prototyps durchgeführte Evaluation ist Thema von Kapitel 7. Anhand der Evaluationsergebnisse werden sowohl Stärken als auch Schwächen aufgezeigt. Abschließend fasst Kapitel 8 noch einmal die Ergebnisse der Arbeit zusammen und gibt einen Ausblick für zukünftige Arbeiten.

2 Grundlagen und Begriffe

Im ersten Abschnitt dieses Kapitels werden der Begriff Text Mining und die dazugehörigen Aufgaben erklärt. Des Weiteren werden die zu berücksichtigenden Herausforderungen erläutert und mögliche Anwendungsgebiete aufgezeigt. Da die Hauptaufgabe dieser Arbeit der Entwurf einer intuitiven Benutzerschnittstelle ist, widmet sich Abschnitt 2.2 grundlegenden Aspekten, die beim Entwurf beachtet werden sollten.

2.1 Text Mining

Der Begriff Text Mining (TM) stammt ursprünglich von dem Begriff Data Mining (DM) ab. Das Ziel des DMs ist es, relevante Informationen oder Muster in einer großen Menge von strukturierten Daten zu entdecken. Im Gegensatz zum DM bilden beim TM unstrukturierte Textdaten, wie beispielsweise Textdokumente, Blogeinträge, E-Mails oder Berichte die Datenquelle. TM wird daher auch als *Data Mining auf Textdaten* [RB97], *Text Data Mining* [Hea99] oder *Textual Data Mining* [LOK00] bezeichnet. Über die konkreten Aufgaben, die TM zugewiesen werden, ist sich die Literatur nicht ganz einig. Beispielsweise setzt Sebastiani TM weitestgehend mit Informationsextraktion (IE) und Textklassifkation (TK) gleich [Seb02], während Kosala, Blockeel und Dörre diese zwei Aufgaben nur als Teilgebiete sehen und zusätzlich die Wissensentdeckung bzw. die Verknüpfung der extrahierten Informationen mit einbeziehen [DGS99, KB00]. Heast im Gegensatz zählt die IE explizit nicht zu dem Bereich des TMs, sondern betont die Metapher des „Minings“ (Goldschürfen), wonach TM nur die Aufgabe hat, neues vorher nicht erkanntes Wissen zu entdecken [Hea99]. Der Begriff TM unterliegt also unterschiedlichen Interpretationen. In dieser Arbeit wird die Definition von [DGS99] verwendet. Danach umfasst TM:

> „[...] sowohl die Extraktion von Informationen und Merkmalen aus einzelnen Dokumenten als auch die Analyse dieser [...] über komplette Dokumentensammlungen zur Entdeckung interessanter Phänomene, Muster oder Trends.“[DGS99]

Nach dieser Definition kann TM wie in [Tan99] als zweiphasiger Prozess verstanden werden, bei dem es in der ersten Phase darum geht, relevante Informationen in strukturierter Form zu ermitteln, die dann in der zweiten Phase entsprechend verknüpft

und auf einen konkreten Kontext hin untersucht werden. Abbildung 2.1 stellt diesen Prozess graphisch dar.

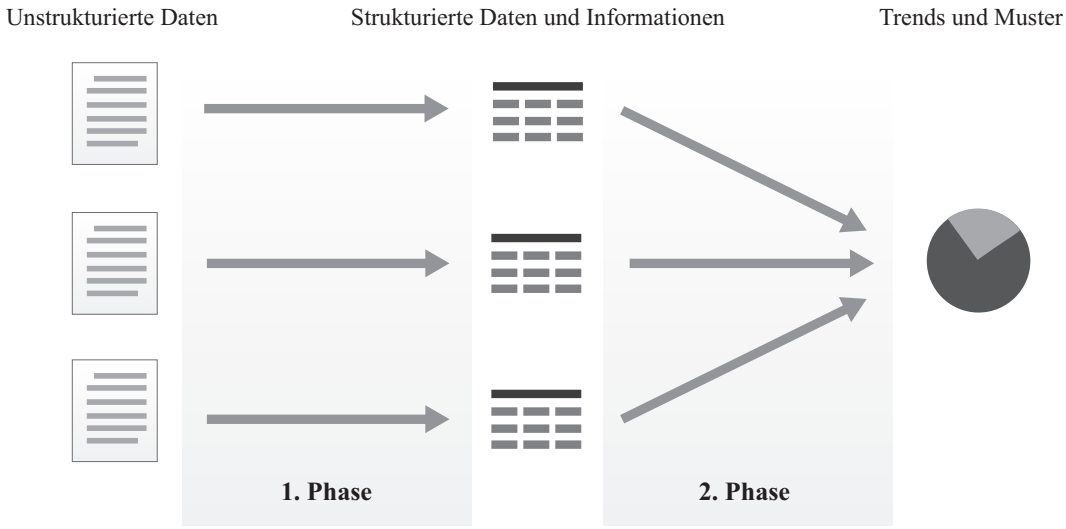


Abbildung 2.1: Prozess des Text Minings

Der Ansatz aus [SS11], welcher die Grundlage für die zu entwerfende Benutzeroberfläche bildet, fokusiert sich auf die Aufgaben der ersten Phase. Diese sollen im Folgenden genauer betrachtet werden.

2.1.1 Text Mining Aufgaben

Entsprechend der Informationen, die aus einem Dokument ermittelt werden, lassen sich die Analyseverfahren verschiedenen TM-Aufgaben zuordnen. In den folgenden Abschnitte werden die unterschiedlichen TM-Aufgaben vorgestellt.

Erkennung von Entitäten (NER)

Das ursprüngliche Ziel dieser TM-Aufgabe bestand darin, Entitäten wie Personen, Firmen und Orte in Texten zu erkennen. Das Verfahren wird daher auch als *Named Entity Recognition* (NER) bezeichnet. Wenig später kamen noch numerische und zeitliche Angaben hinzu. Aufbauend auf diesen drei Kategorien (Name, Zeit und numerischen Angaben) haben sich seit der Einführung der NER verschiedene Typen von Entitäten wie spezielle Personen- oder Organisationstypen herausgebildet. Trotz bestehender Typhierarchien wie der von [Sek10], verwenden Textanalysedienste ihre eigenen Taxonomien [Orc11, Cal11b].

Bei vielen Verfahren wird heutzutage auch eine Koreferenzanalyse durchgeführt. Das Ziel dieser Analyse ist es, sprachliche Ausdrücke zu identifizieren, welche sich auf ein und die selbe Entität beziehen. In den zwei Sätzen „Herr Müller fährt einen Golf. Er

ist sehr zufrieden damit.“ besteht zum Beispiel eine Koreferenz zwischen „Er“ und „Herr Müller“ sowie zwischen „Golf“ und „damit“. Mit Hilfe dieser Methode können alle Instanzen einer Entität ermittelt werden [NS07].

Erkennung von Relationen zwischen Entitäten (ERD)

Eine weitere wichtige TM-Aufgabe ist das Finden von Relationen zwischen zwei oder mehreren Entitäten (ERD[1]) . Beispiele solcher Relationen sind „ist Chef von“ zwischen einer Person und einer Firma, „ist Konkurrent von“ zwischen zwei Firmen oder „ist ausgebrochen in“ zwischen einer Krankheit und einem Ort. Ähnlich wie bei der NER lassen sich auch die Relationen unterschiedlichen Typen zuordnen [Sar07]. Die Extraktions- bzw. TM-Dienst verwenden dabei ihre eigenen Taxonomien [Orc11, Cal11b].

Stimmungs- und Meinungsanalyse (SA und OM)

Durch den zunehmenden digitalen Austausch von Meinungen, Kritiken und Bewertungen über Online-Platformen oder andere Kommunikationskanäle stieg das Interesse, neben reinen Fakten auch die in Texten zum Ausdruck gebrachte Subjektivität zu extrahieren. Bei den Verfahren, die dafür entwickelt wurden, unterscheidet man zwischen Opinion Mining (OM) und Sentiment Analyse (SA). In Anlehnung an die NER besteht das Ziel des OMs darin, konkrete Meinungen zu bestimmten Produkten oder Themen zu ermitteln und diese in strukturierter Form aufzubereiten. Bei der SA wird versucht die im Text zum Ausdruck gebrachten Gefühle wie Freude oder Wut zu ermitteln. Einige Verfahren vereinfachen dies soweit, dass nur die Polarität bestimmt wird, also ob ein Text positiv oder negativ formuliert ist [Sar07].

Stichwortextraktion und Konzeptzuweisung (SE und KoZ)

Stichwörter eignen sich hervorragend um den Inhalt eines Textes grob wiederzugeben. Obwohl sie inhaltlich keinen Ersatz für Zusammenfassungen bieten, besitzen sie den Vorteil, dass sie in strukturierter Form vorliegen. Da die manuelle Zuweisung von Stichwörtern ein sehr zeitaufwendiger Prozess ist, wurden Verfahren zur automatischen Ermittlung von inhaltlich passenden Stichwörtern entwickelt. Unterschieden wird dabei zwischen Verfahren der Stichwortextraktion (SE) und der Stichwortgenerierung bzw. Konzeptzuweisung (KoZ). Die Verfahren der SE suchen relevante Textphrasen innerhalb des Textes, wohingegen die Verfahren der KoZ auf Basis von inhaltlichen Analysen und Wissensdatenbanken passende Begriffe generieren [EC07].

[1] Ausgehend von der englischen Bezeichnung *Entity Relationship Detection*

Textklassifikation (TK)

Das Ziel bei den Verfahren der automatischen Textklassifikation (TK) oder Textkategorisierung ist die Zuordnung eines Dokuments zu einem oder mehreren vorgegebenen Themengebieten bzw. Kategorien. Mit Hilfe dieser Verfahren kann beispielsweise ermittelt werden, von welchen Themen ein Text handelt [Seb02].

2.1.2 Herausforderungen

Beim Einsatz der genannten TM-Verfahren müssen verschiedene Aspekte hinsichtlich der Genauigkeit, Laufzeit und Einsatzgebiete berücksichtigt werden. Da sich diese Aspekte auch auf die Verwendung der TM-Dienste auswirken, werden sie im Folgenden erläutert.

Genauigkeit

Generell sind die Verfahren so konzipiert, dass die Texte abhängig von der zu ermittelnden Information auf eine bestimmte Menge von Hinweisen untersucht werden. Die Genauigkeit eines Verfahrens hängt damit von der Menge der zu untersuchenden Hinweise ab. Umso mehr Hinweise dabei gefunden werden, um so höher ist die Konfidenz bzw. die Sicherheit, dass das ermittelte Ergebnis stimmt. Ausgedrückt wird dies durch den Konfidenzwert [Tan99, Sar07, App99].

Laufzeit

Neben einer hohen Genauigkeit wird auch eine kurze Laufzeit angestrebt. Diese hängt jedoch von verschiedenen Faktoren wie der Komplexität des Verfahrens, der zu erledigenden Aufgabe, der Anzahl der zu betrachtenden Dokumente sowie der zur Verfügung stehenden Ressourcen ab. Es muss daher berücksichtigt werden, dass jede TM-Aufgabe eine gewisse Zeit in Anspruch nimmt [Sar07].

Domänen- und Sprachabhängigkeit

Wie bei der Genauigkeit bereits erläutert, müssen zur Extraktion bzw. Ermittlung der Informationen eine Menge spezifischer Hinweise berücksichtigt werden. Aufgrund der syntaktischen und semantischen Vielfältig von unstrukturierten Daten beschränken sich die meisten Verfahren daher auf bestimmte Domänen und Sprachen, um somit eine höhere Genauigkeit bei den Ergebnissen zu erzielen [Tan99, Sar07].

2.1.3 Anwendungsgebiete

Obwohl TM-Verfahren vorrangig im Bereich der Business Intelligence eingesetzt werden, um neue Erkenntnisse aus unstrukturierten Datenquellen zu ermitteln, lassen

sich die extrahierten Informationen noch für weitere Anwendungsgebiete einsetzen [DGS99, CLRR10]. Diese werden im Folgenden erläutert.

Semantische Suche

Obwohl das Finden von Dokumenten keine Aufgabe des TMs ist, können die Verfahren helfen, die Suche zu verbessern, indem die Dokumente entsprechend der extrahierten Informationen kategorisiert werden [DGS99]. Zudem können die NER oder ERD genutzt werden, um die Semantik von Begriffen und Textstellen zu klären und bei der Suche zu berücksichtigen [CLRR10].

Datengetriebene Mashups

Mit Hilfe der extrahierten Informationen lassen sich unstrukturierte Quellen leichter mit anderen unstrukturierten oder strukturierten Daten verknüpfen. Somit können automatisch datengetriebene Mashups aus unterschiedlichen Quellen erstellt werden [CLRR10]. In [Sar07] wird als Beispiel ein Video-Generator für Nachrichten genannt, welcher zu aktuellen Ereignissen passende Videos mit Bildern von Personen und Orten generiert.

Daten als Dienste

Die extrahierten Daten können in strukturierter Form für andere Anwendungen angeboten werden. Sarawagi nennt in ihrer Untersuchung verschiedene Beispiele wie die automatische Erstellung von Literaturdatenbanken oder die Extraktion von Produktdaten für Shopping-Vergleichsseiten [Sar07, CLRR10].

2.2 Gestaltung von Benutzeroberflächen

Die Akzeptanz einer Anwendung hängt maßgeblich von ihrer Usability ab [VW01]. Nach ISO 9241-11 ist der Begriff Usability definiert als,

> „[das] Ausmaß, in dem ein Produkt durch bestimmte Benutzer in einem bestimmten Nutzungskontext genutzt werden kann, um bestimmte Ziele effektiv, effizient und zufriedenstellend zu erreichen.“ [VW01]

Dies bedeutet, dass die Brauchbarkeit eines Produkts bzw. eines Systems nicht nur von der Benutzerschnittstelle, sondern auch von den verfügbaren Funktionen abhängt. Im Fall von Softwareanwendungen ist die Benutzeroberfläche der für den Nutzer sichtbare Teil des Systems. Sie trägt damit einen großen Anteil.

Um eine möglichst intuitive Benutzeroberfläche zu entwickeln, sollten nach [WGL04] drei grundlegende Aspekte berücksichtigt werden. Dies betrifft erstens die Verwendung von Prinzipien, Richtlinien und Entwurfsmustern, zweitens den Einbezug des

Nutzers während des Entwurfs und drittens das frühzeitige, iterative Testen. Im Folgenden werden diese Aspekte genauer erläutert.

2.2.1 Richtlinien, Normen und Prinzipien

Zu Richtlinien zählen hauptsächlich Styleguides oder Guidelines. Sie geben plattform- oder produktspezifische Regeln für das Aussehen und die Verwendung von einzelnen Interaktionskomponenten wie Buttons oder Menüs vor. Hierzu zählen beispielsweise der SAP R3/Styleguide [SAP11] oder die Apple HCI Guidelines [App11].

Neben Richtlinien gibt es eine Vielzahl von prozess- oder anwendungsorientierten Normen und Standards zur Gestaltung von Benutzeroberflächen. In der Regel sind diese nicht kostenfrei und werden daher in dieser Arbeit nicht weiter betrachtet [RF10].

Beim Entwurf einer Benutzeroberfläche sollten jedoch verschiedene Prinzipien berücksichtigt werden. Eine sehr bekannte Sammlung solcher Prinzipien sind die „8 Goldenen Regeln“ von Shneiderman [SP04]. Diese werden in den folgenden Abschnitten vorgestellt.

1. Strebe Konsistenz an

Das System sollte eine gewisse Konsistenz hinsichtlich der verwendeten Terminologie, des Aufbaus sowie der verwendeten Icons und Farben aufweisen. Hierzu zählen auch allgemeine Erwartungen an die Interaktion, wie zum Beispiel das Bestätigen beim Löschen eines Elements oder das Verbergen von Passwörtern.

2. Universelle Benutzerfreundlichkeit

Um eine universelle Benutzerfreundlichkeit zu erreichen, sollten nicht nur Alter, Wissen, Fertigkeiten und Einschränkungen der Nutzer, sondern auch die bereits gemachten Erfahrungen mit der Anwendung berücksichtigt werden. So sollten unerfahrene Nutzer durch Hilfsfunktionen und Anleitungen unterstützt werden, wohingegen erfahrene Nutzer Zugriff auf Zusatzfunktionen oder Abkürzungen erhalten.

3. Informative Rückmeldungen

Damit jederzeit ersichtlich wird, was das System gerade macht oder gemacht hat, sollte der Nutzer für jede Aktion eine entsprechende Rückmeldung erhalten. Hierzu zählen beispielsweise das Kennzeichnen von manipulierten Objekten, Erfolgsrückmeldungen für abgeschlossene Dialoge oder das Anzeigen von noch laufenden Prozessen.

4. Abgeschlossene Dialoge

Jeder Dialog sollte einen Anfang, eine Mitte und ein Ende haben. Die Übergänge zwischen den einzelnen Stellen sollten dabei entsprechend signalisiert werden, sodass

der Nutzer immer sieht, wie weit er fortgeschritten ist. Zudem sollte der Nutzer nach Beendigung eine Rückmeldung über den Erfolg des Dialogs erhalten.

5. Fehlervermeidung

Weil davon ausgegangen werden muss, dass ein Nutzer Fehler bei der Bedienung macht, sollte ihm das System dabei helfen diese Fehler zu vermeiden. So sollte das System zum Beispiel nachfragen, bevor es eine Datei überschreibt oder bevor der Nutzer ein ungesichertes Dokument schließt. Auch die Eingaben des Nutzers sollten stets kontrolliert werden, sodass keine ernsthaften Schäden am System selbst entstehen können.

6. Erlaube das Rückgängigmachen von Aktionen

Alle vom Nutzer durchgeführten Aktionen sollten auch rückgängig gemacht werden können. Dies nimmt dem Nutzer die Angst etwas falsch zu machen und fördert das Erforschen von noch unbekannten Funktionen.

7. Vermeide Überraschungen und Ablenkungen

Ein Benutzer sollte stets das Gefühl haben, dass er die Kontrolle über das System hat und die Aktionen von ihm ausgehen. Unvorhersehbare oder schwer erreichbare Aktionen und Informationen sollten daher vermieden werden.

8. Schone das Kurzzeitgedächtnis

Die beschränkte Merkfähigkeit des menschlichen Kurzzeitgedächtnis verlangt, dass die Benutzeroberfläche so einfach wie möglich gehalten werden sollte. Dazu sollten zusammengehörende Informationen auch weitestgehend zusammen dargestellt, das Wechseln der Ansichten reduziert und genügend Trainingszeit zum Erlernen von komplexen Aktionen bereitgestellt werden.

2.2.2 Allgemeine Entwurfsmuster

Richtlinien, wie die „8 Goldenen Regeln“ von Shneidermann, geben wertvolle Hinweise zur Gestaltung von Benutzeroberflächen, jedoch sind sie sehr allgemein und nicht problembezogen. Eine genauere Hilfe zur Gestaltung von Benutzerschnittstellen bieten Entwurfsmuster.

Die Idee des Entwurfsmuster wurde das erste Mal 1978 vom Architekten Christopher Alexander beschrieben. Er definiert diesen Begriff wie folgt:

> „Jedes Entwurfsmuster ist eine dreiteilige Regel, welche eine Verbindung zwischen einem bestimmten Kontext, einem Problem und einer Lösung ausdrückt“ [ASI78].

Eine Entwurfsmuster beschreibt also immer eine konkrete Lösung für ein bestimmtes Problem. Für die Gestaltung von Benutzeroberflächen existieren heutzutage viele Bibliotheken mit Entwurfsmustern [Tox11, Tec11, Tid11, SN09, VW11]. Im Folgenden werden einige grundlegende Entwurfsmuster vorgestellt. Die Auswahl orientiert sich dabei an den „12 Standard Screen Pattern“ von [Nei09]. Abbildung 2.2 zeigt grafische Beispiele der betrachteten Entwurfsmuster.

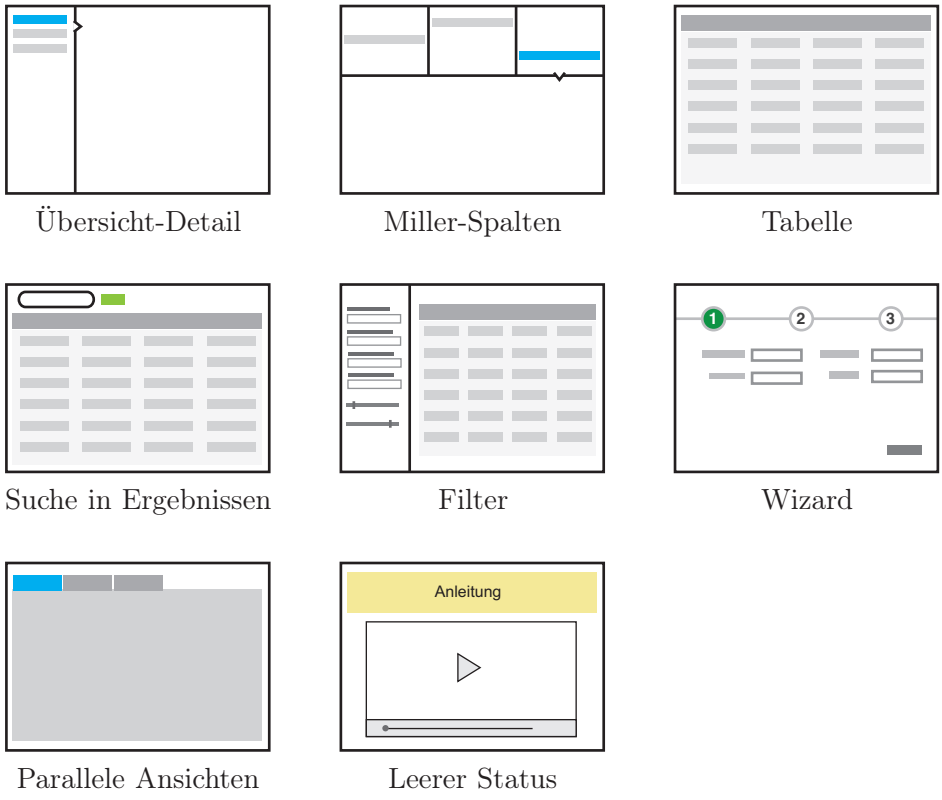


Abbildung 2.2: Übersicht der allgemeinen Entwurfsmuster in Anlehnung an [Nei09]

Übersicht-Detail

Bei diesem Entwurfmuster wird die gesamte Ansicht in einen Übersichts- und Detailbereich eingeteilt. Der Übersichtsbereich dient zur Auflistung einer Menge von Objekten. Wählt der Nutzer eines dieser Objekte aus, erscheint im Detailbereich die entsprechende Ansicht für das Objekt. Dieses Muster ist weit verbreitet und kommt beispielsweise in vielen E-Mail-Programmen oder dem Windows-Explorer zum Einsatz. Die Detailansicht kann dabei entweder unterhalb oder neben der Liste angeordnet sein. Der Vorteil dieses Entwurfsmusters besteht darin, dass der Nutzer stets die Übersicht über die zur Verfügung stehenden Objekte erhält und auf effiziente Weise darauf zugreifen kann.

Miller-Spalten

Das Miller-Spalten Entwurfsmuster ermöglicht es dem Nutzer, ausgehend von verschiedenen Anfangspunkten, durch eine Elementhierarchie zu navigieren. Jede Hierarchieebene wird dabei innerhalb einer Spalte dargestellt. Klickt der Nutzer ein Element einer Spalte an, werden in einer weiteren Spalte die Unterelemente dargestellt. Der Vorteil dieses Entwurfsmuster besteht darin, dass der Nutzer immer sieht an welcher Stelle der Hierarchie er sich gerade befindet.

Tabellen

Dieses Entwurfsmuster erlaubt es eine Liste von Daten auf effiziente Weise zu überfliegen, zu bearbeiten oder auch einzugeben. [Nei09] empfiehlt, dass eine Tabelle folgende Funktionen anbieten sollte: Sortieren, Ein- und Ausblenden von Spalten, Umsortieren von Spalten, Gruppierungen, globales Rückgängigmachen, Hinzufügen, Einfügen und Löschen von Zeilen sowie Import- und Exportfunktionen.

Suche in Ergebnissen

In datenverarbeitenden Anwendungen steht der Nutzer meist einer Vielzahl von Objekten gegenüber. Eine Stichwortsuche eignet sich, um das Finden von relevanten Objekten zu vereinfachen. Durch die Eingabe von Suchbegriffen gelangt der Nutzer so schnell zu den gewünschten Objekten.

Filter

Eine weitere Möglichkeit zur Eingrenzung von Objekten sind Filter. Im Gegensatz zur Stichwortsuche sind Filter speziell auf die Eigenschaften von Objekten zugeschnitten und ermöglichen somit eine noch genauere Einschränkung der Ergebnisse. Dieses Entwurfsmuster wird zum Beispiel in Online-Shops zur Produktsuche verwendet.

Wizards

Sehr lange oder umfangreiche Dialoge sollten in mehrere kleine Schritte aufgeteilt werden, sodass der Nutzer nicht mit zu vielen Ein- und Ausgaben auf einmal konfrontiert wird. Dabei ist es wichtig stets auf die aktuelle Position innerhalb des Dialogs hinzuweisen. Zu finden ist dieses Entwurfsmuster zum Beispiel in Installationsdialogen.

Parallele Ansichten

Dieses Entwurfsmuster wird verwendet, um unterschiedliche Facetten eines Objekts getrennt voneinander darzustellen. Häufig eingesetzt wird es beispielsweise bei Nutzerkonten zur separaten Darstellung von personenbezogenen und anderen, das Konto betreffenden Daten. Die Ansicht kann dabei vertikal, wie bei einem Akkordeon, oder horizontal, wie bei Tabs, geteilt sein.

Leerer Status

Wenn die Anwendung das erste Mal geöffnet wird und noch keine Eingaben getätigt wurden, steht sehr viel wertvoller Platz zur Verfügung. Dieser sollte genutzt werden, um dem Nutzer durch ein kurzes Einführungsvideo oder eine Anleitung die Funktionen der Anwendung näher zu bringen.

2.2.3 Evaluation von Benutzeroberflächen

Die Verwendung von Richtlinien und Entwurfsmustern sind eine große Hilfe bei der Gestaltung von Benutzeroberflächen. Sie geben jedoch keine Garantie dafür, dass die Interaktion ohne Probleme abläuft. Es ist daher wichtig, die Benutzeroberfläche zu evaluieren. Hierfür existieren zwei Ansätze, welche im Folgenden vorgestellt werden [RF10].

Experten Reviews

Ein Experten Review ist eine unabhängige und neutrale Begutachtung der Benutzeroberfläche durch einen Usability-Experten. Ein Review kann sowohl in frühen als auch in späten Phasen des Entwicklungsprozesses eingesetzt werden. Die Untersuchung erfolgt dabei nach allgemeinen Richtlinien, Normen sowie der eigenen Erfahrung. Die Ergebnisse der Untersuchung werden den Entwicklern in Form von Gutachten präsentiert, in denen die Stärken und Schwächen aufgeführt sind. Auf Verbesserungsvorschläge wird dabei verzichtet.

Der Vorteil von Experten Reviews ist der relativ geringe zeitliche und finanzielle Aufwand. Problematisch gestalten sich Experten Reviews allerdings bei sehr speziellen Anwendungen, bei denen die Usability Experten keine ausreichende Kenntnisse über den Anwendungsbereich sowie die Zielgruppe besitzen. Neben Experten Reviews sollten daher auch Usability Tests durchgeführt werden [RF10].

Usability Tests

Ein Usability Test sieht vor, dass eine Gruppe von Testpersonen eine vorgegebene Menge von Aufgaben bewältigen muss. Die Anzahl der Testpersonen hängt dabei von der Art der Anwendung sowie der aktuellen Entwicklungsphase ab. Um eine qualitative Aussage zur Verbesserung und Anpassung der Benutzeroberfläche zu erhalten, reichen 4-6 Testpersonen pro Iteration aus. Bei Systemen mit hohen Anforderungen sollten es 7-15 sein, bei kritischen Systemen nicht unter 15.

Zur Evaluation wird ein Beispielszenario formuliert und Aufgaben definiert, die von den Testpersonen zu erledigen sind. Die Aufgaben sollten dabei weder zu einfach noch zu kompliziert sein. Zum Beispiel ist „Überweise Sie den Betrag von ...“ besser als „Gehen Sie im Menü auf Einzahlungen ...“. Für den Test sollte die Anwendung so weit

vorbereitet sein, dass die Testpersonen den Eindruck eines möglichst realistischen, lauffähigen Systems bekommen.

Zur Befragung der Testpersonen können sowohl Einzelinterviews durchgeführt als auch Fragebogen verwendet werden. Fragebögen eignen sich vor allem für eine größere Anzahl von Testpersonen und haben den Vorteil, dass deren Ergebnisse besser verglichen werden können. Abhängig vom Ziel der Befragung können selbst erstellte Fragebögen oder Standardfragebögen eingesetzt werden.

Zur Durchführung eines Usability Tests gibt es zwei unterschiedliche Methoden [RF10].

Bei der ersten Methode wird der Test in einem eigens für die Evaluation eingerichteten Usability Lab durchgeführt. Unter der Beobachtung von Entwicklern und Usability-Experten arbeitet die Testperson selbstständig die Aufgaben laut denkend durch. Mit Hilfe von Kameras und Screencasting-Systemen werden dabei alle Schritte genau protokolliert. Anschließend werden in einer Nachbesprechung die Testpersonen zu einzelnen Stellen interviewt, um so ein genaueres Bild über die Schwachpunkte zu bekommen [RF10].

Die zweite Testmethode ist ein Usability Walk Through. Anstatt die Testpersonen in einem abgetrennten Raum arbeiten zu lassen, begleitet der Testleiter den Benutzer bei der Durchführung der Aufgaben. Der Testleiter kann somit direkt eingreifen, Fragen stellen oder bestimmte Abläufe mit dem Nutzer durchgehen. Die Methode kann schon sehr früh im Entwicklungstest eingesetzt werden, um einzelne Teile der Benutzeroberfläche zu evaluieren [RF10].

3 Anforderungsanalyse

In diesem Kapitel werden die mit der zu entwerfenden Benutzeroberfläche einhergehenden Anforderungen analysiert. Dazu wird in Abschnitt 3.1 die Zielgruppe des Systems betrachtet. Anschließend wird in Abschnitt 3.2 die Architektur und Funktionsweise des Gesamtsystems erläutert, bevor in den Abschnitten 3.3 und 3.4 die funktionalen sowie nichtfunktionalen Anforderungen an die Benutzeroberfläche hergeleitet werden.

3.1 Zielgruppe und Anwendungsgebiete

Die Hauptzielgruppe des Systems sind Personen, welche noch keine oder nur wenig Erfahrung mit TM-Systemen haben und nach einer einfachen Möglichkeit suchen, relevante Informationen aus einem oder mehreren Textdokumenten ermitteln zu lassen. Dies kann beispielsweise ein Journalist sein, der sich alle in einem Text genannten Personen anzeigen lassen will. Möglich wäre auch ein Forscher, der mit dem System naturwissenschaftliche Artikel kategorisieren will. Auch könnte die Anwendung dazu verwendet werden, aus verschiedenen Webseiten konkrete Meinungen über ein spezielles Produkt zu ermitteln. Durch den Einsatz verschiedener externer Textanalysedienste gibt es vielfältige Anwendungsgebiete. Die Benutzeroberfläche sollte daher möglichst einfach und intuitiv sein, sodass sie von Nutzern aus verschiedenen Bereichen bedient werden kann. Zusätzlich soll sie aber auch Personen mit TM-Kenntnissen einen Mehrwert bieten, indem verschiedene Zusatzfunktionen zur Korrektur und Bewertung der Ergebnisse angeboten werden.

3.2 Architektur und Funktionsweise des Text Mining Systems

Um die Anforderungen an die zu entwerfende Benutzeroberfläche zu verstehen, wird im Folgenden die Architektur und Funktionsweise des Gesamtsystems erläutert.

Das TM-System, für das die Benutzeroberfläche entworfen werden soll, verfolgt einen serviceorientierten Ansatz. Nach [SS11] werden den Nutzern auf Basis bestehender,

externer TM-Dienste, wie OpenCalais [Cal11b] oder AlchemyAPI [Orc11], die folgenden TM-Aufgaben zur Verfügung gestellt:

- Erkennung von Entitäten (NER)
- Erkennung von Relationen zwischen Entitäten (ERD)
- Textklassifikation (TK)
- Konzeptzuweisung (KoZ)
- Stichwortextraktion (SE)
- Sentiment-Analyse (SA)

Da die Zielgruppe hauptsächlich aus unerfahrenen Nutzern besteht, sollen die Dienste weitestgehend unsichtbar bleiben. Die Nutzer sollen sich auf die Auswahl der zu analysierenden Textdaten und genannten TM-Aufgaben konzentrieren.

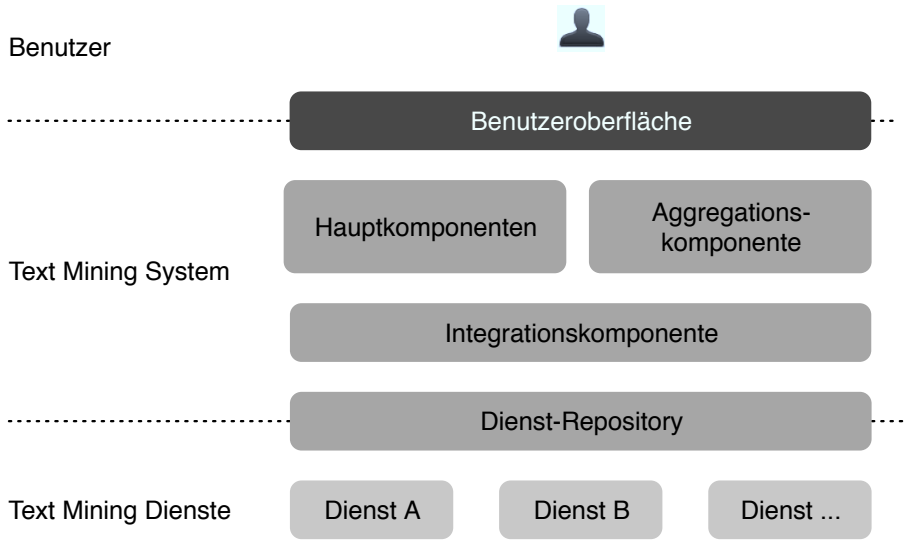


Abbildung 3.1: Architektur des Text Mining Systems

Abbildung 3.1 zeigt die Architektur des Systems. Die unterste Schicht bildet das Dienst-Repository. Hier wird jeder TM-Dienst mit einer entsprechenden Beschreibung registriert. In der Beschreibung sind neben den verfügbaren TM-Aufgaben verschiedene Aspekte wie unterstützte Sprachen, Nutzungskosten sowie Sicherheits- und Datenschutzrichtlinien der Dienste definiert. Auf Basis dieser Dienstbeschreibung und der vom Nutzer ausgewählten TM-Aufgaben werden in der Integrationskomponente die passenden TM-Dienste herausgesucht und aufgerufen.

Eine TM-Aufgabe wird somit in bestimmten Fällen nicht nur von einem, sondern von mehreren TM-Diensten abgearbeitet. Um die Vollständigkeit und Genauigkeit der Ergebnisse zu erhöhen, werden die Resultate der einzelnen Dienste mit Hilfe der Aggregationskomponente zu einem Gesamtergebnis zusammengeführt. Abbildung 3.2

zeigt dies an einem konkreten Beispiel für die NER. In dem dargestellten Beispiel wird die Vereinigungsmenge der Ergebnisse von Dienst A und B gebildet.

Text	David Eun verlässt Google und geht zu AOL
TM-Dienst A	**David Eun** verlässt Google und geht zu AOL
TM-Dienst B	David Eun verlässt **Google** und geht zu **AOL**
Aggregiert	**David Eun** verlässt **Google** und geht zu **AOL**

Abbildung 3.2: Beispiel einer Aggregation bei der NER

Die Hauptkomponenten enthalten alle notwendigen Funktionen zur Eingabe und zum Speichern von Textdaten, sowie zur Verwaltung der Nutzereinstellungen und Analyseergebnisse. Eine genaue Betrachtung dieser Funktionen sowie der damit einhergehenden Anforderungen an die Benutzeroberfläche erfolgt in den nächsten Abschnitten.

3.3 Funktionale Anforderungen

Die in dieser Arbeit zu entwerfende Benutzeroberfläche soll die Grundfunktionalität zur Bedienung des TM-Systems bereitstellen. Auf spezielle Funktionen für konkrete Anwendungsgebiete, wie Business Intelligence oder einer semantischen Suche, soll dabei verzichtet werden.

Zu den Grundfunktionen gehören die Eingabe der zu analysierenden Textdokumente, die Konfiguration der Textanalyse sowie die Ausgabe der Analyseergebnisse. Des Weiteren soll davon ausgegangen werden, dass das System lernfähig ist. Die Nutzer sollen daher in der Lage sein, dem System Feedback zu geben. Diese vier grundlegenden Anforderungen werden in den folgenden Abschnitten genauer betrachtet. Eine Zusammenfassung der genauen Anforderungen befindet sich Abschnitt 3.3.5.

3.3.1 Eingabe und Verwaltung der Textdaten

Bevor eine Textanalyse durchgeführt werden kann, müssen die zu analysierenden Textdaten in das System geladen werden. Hierfür soll die Benutzeroberfläche eine Möglichkeit bieten, verschiedene Textdokumente wie PDFs oder Word-Dateien auszuwählen. Die Benutzeroberfläche soll dabei so flexibel gestaltet werden, dass später auch Textdaten aus anderen Quellen wie beispielsweise Suchmaschinen, Nachrichtenfeeds oder Foren eingebunden werden können.

Genauso wie das Hinzufügen von Dokumenten soll auch das Löschen von Dokumenten möglich sein. Für sehr vielen Dokumenten ist es hilfreich, wenn diese nach bestimmten Kriterien gruppiert werden können. Hierfür soll der Nutzer Dokumentensammlungen erstellen können. Ähnlich wie die Dokumente sollen diese auch gelöscht werden können.

In Anbetracht dessen, dass nicht alle Inhalte innerhalb eines Dokuments interessant sind, wie beispielsweise Werbebanner in Webseiten, sollte es eine Funktion zum Markieren relevanter Inhalte geben. Diese Funktion sei jedoch nur als Idee genannt. Des Weiteren sollen die Nutzer in der Lage sein, sich den Inhalt einzelner Dokumente anzusehen. Da der Nutzer mitunter die hinzugefügten Dokumente und Dokumentensammlungen noch für spätere Textanalysen verwenden will, sollten diese im System gespeichert werden können.

Eine weitere wichtige Funktion auf Dokumentenbasis stellt das Annotieren von Dokumenten dar. Hierdurch sollen erfahrene Nutzer die Möglichkeit bekommen, Trainingsdaten oder Musterlösungen für die unterschiedlichen TM-Aufgaben zu erstellen.

3.3.2 Konfiguration der Textanalyse

Hat der Nutzer die zu analysierenden Dokumente eingegeben, muss er festlegen, welche Informationen analysiert werden sollen. Hierfür soll der Nutzer aus den sechs verfügbaren TM-Aufgaben eine oder mehrere auswählen können.

Zur genauen Konfiguration einer TM-Aufgabe werden sowohl vom System als auch von den TM-Diensten verschiedene Optionen bereitgestellt. Damit diese genutzt werden können, muss die Benutzeroberfläche verschiedene, aufgabenspezifische Einstellungen anbieten. Speziell für die NER oder ERD soll dabei die Möglichkeit bestehen, die zu erkennenden Typen von Entitäten bzw. Relationen festzulegen. Dadurch können von vornherein die Dienste, die nicht in der Lage sind bestimmte Typen zu finden, ausgeschlossen werden. Zusätzlich soll für jede TM-Aufgabe ein minimaler Konfidenzwert eingestellt werden können, sodass die Anzahl der falschen Ergebnisse reduziert werden.

Generell soll die Dienstauswahl weitestgehend automatisch erfolgen. Jedoch gibt es einige Kriterien bei der Auswahl, auf die der Nutzer mitunter Einfluss nehmen will. Hierzu zählt beispielsweise die Ausführungsdauer. Jeder TM-Dienst benötigt eine bestimmte Zeit für die Durchführung der Aufgaben. Weil bei einigen Textanalysen die Zeit von Relevanz ist, sollte der Nutzer das System anweisen können, möglichst schnelle Dienste einzusetzen. Die Laufzeit kann dabei über die Anzahl an Transaktionen, die ein Dienst pro Sekunde durchführen kann, berechnet werden. Damit der Nutzer sieht, wie lange die Textanalyse dauert, soll die berechnete Laufzeit mit angezeigt werden.

Neben der Ausführungsdauer spielen auch die Sicherheit und der Datenschutz eine wichtige Rolle. In bestimmten Fällen kann es vorkommen, dass sensible Daten untersucht werden sollen. Einige Dienste speichern jedoch die gesendeten Daten zur internen Auswertung ihrer Ergebnisse. Um einen Missbrauch der Daten zu vermeiden, sollte der Nutzer explizit festlegen können, dass seine Daten nicht gespeichert werden, sodass dies bei der Auswahl der Dienste berücksichtigt wird. Sensible Daten sollten zudem über eine gesicherte Verbindung übertragen werden. Da nicht alle Dienste solch eine Verbindung unterstützen, muss es auch hierfür eine Option geben.

Ein weiteres Kriterium für die Auswahl der Dienste, sind die Kosten. Im System stehen sowohl kostenfreie als auch kostenpflichtige Dienste zur Verfügung. Da der Nutzer das System zu Beginn erst einmal ausprobieren will und dafür nichts bezahlen möchte, soll er einstellen können, dass nur kostenlose Dienste verwendet werden. Für den Fall, dass der Nutzer auch kostenpflichtige Dienste einsetzen will, sollten die dafür anfallenden Nutzungskosten entsprechend ersichtlich werden.

Hat der Nutzer die Textanalyse nach seinen Wünschen konfiguriert, will er sie mitunter für spätere Analysen weiterverwenden. Der Nutzer soll daher die Konfiguration in einem Textanalyseprofil speichern können.

3.3.3 Ausgabe der Ergebnisse

Nach Ausführen der Textanalyse stehen je nach TM-Aufgaben verschiedene Analyseergebnisse bereit. Diese sollen in der Benutzeroberfläche ausgegeben werden. Um die Ergebnisse überprüfen zu können, soll der Nutzer die Möglichkeit bekommen, sich für jedes Dokument die genauen Ergebnisse anzusehen. Die Ergebnisse der NER und ERD sollen dabei direkt im Text hervorgehoben werden.

Wie in Abschnitt 3.2 beschrieben, werden die verschiedenen Ergebnisse einer TM-Aufgabe zu einem Gesamtergebnis zusammengeführt. Die Aggregation kann dabei nach verschiedenen Methoden vorgenommen werden und führt dementsprechend zu unterschiedlichen Ergebnissen. Da die automatische Aggregation nicht immer optimal ist, sollen erfahrene Nutzer selbst einstellen können, wie die Ergebnisse aggregiert werden. Eine wichtige Voraussetzung dafür ist, dass der Nutzer sieht, welche Ergebnisse die einzelnen Dienste liefern und welche am besten seinen Erwartungen entsprechen. Es muss also für jede TM-Aufgabe möglich sein, die Resultate einzelner Dienste einzusehen und wenn vorhanden mit anderen zu vergleichen.

Des Weiteren sollte die Benutzeroberfläche eine Funktion anbieten, um die Analyseergebnisse zu exportieren, sodass diese in anderen Anwendungen weiterverarbeitet werden können.

3.3.4 Feedbackmöglichkeiten

Ausgehend von einem lernfähigen System, sollen in der Benutzeroberfläche verschiedene Feedbackfunktionen zur Bewertung und Korrektur der Ergebnisse bereitgestellt werden. Um eine möglichst genaue Bewertung vorzunehmen, soll der Nutzer nicht nur das aggregierte Ergebnis, sondern auch die Ergebnisse einzelner Dienste bewerten können. Mitunter kann es passieren, dass ein aggregiertes Ergebnis nur falsch ist, weil die Mehrheit aller Dienste ein falsches Ergebnis ermittelt haben. In diesem Fall muss der Nutzer dem System mitteilen können, welcher Dienst das richtige Ergebnis liefert. Hat kein Dienst ein richtiges Ergebnis ermittelt, sollte der Nutzer die Möglichkeit bekommen, dieses zu korrigieren. Zudem muss es möglich sein, fehlende Ergebnisse wie beispielsweise nicht erkannte Entitäten zu ergänzen. Je nach TM-Aufgabe müssen also unterschiedliche Korrekturfunktionen bereitgestellt werden.

3.3.5 Übersicht aller Anforderungen

Damit ergeben sich zusammengefasst die folgenden funktionalen Anforderungen an den Entwurf der Benutzeroberfläche:

FA 1 **Eingabe und Verwaltung der Textdokumente**

- FA 1.1 Hinzufügen, Betrachten und Löschen von Textdateien
- FA 1.2 Erstellen, Betrachten und Löschen von Dokumentensammlungen
- FA 1.3 Speichern von Dokumenten und Dokumentensammlungen
- FA 1.4 Annotation von Dokumenten zur Erstellung von Musterlösungen

FA 2 **Konfiguration der Textanalyse**

- FA 2.1 Auswahl der zu analysierenden Dokumente
- FA 2.2 Auswahl der TM-Aufgaben
- FA 2.3 Konfiguration der aufgabenspezifischen Einstellungen
- FA 2.4 Auswahl der zu ermittelnden Entitäts- und Relationstypen
- FA 2.5 Festlegen einer möglichst kurzen Ausführungsdauer
- FA 2.6 Festlegen der Verbindungssicherheit und Datenschutzrichtlinien
- FA 2.7 Festlegen der Kostenoption
- FA 2.8 Anzeigen der Kosten für die Textanalyse
- FA 2.9 Speichern der Textanalysekonfiguration

FA 3 **Ergebnisausgabe**

- FA 3.1 Übersicht aller untersuchten Dokumente
- FA 3.2 Detailansicht der Dokumentenergebnisse

FA 3.3 Vergleich der unterschiedlichen Dienstergebnisse

FA 3.4 Exportieren der Ergebnisse

FA 3.5 Speichern der Analyseergebnisse

FA 4 **Feedbackoptionen**

FA 4.1 Bewertung der Ergebnisse

FA 4.2 Korrektur der Ergebnisse

FA 4.3 Konfiguration der Aggregationsmethode

3.4 Nichtfunktionale Anforderungen

Grundlegend soll eine einfach und intuitiv zu bedienende Benutzeroberfläche erstellt werden, die sowohl erfahrene als auch unerfahrene Anwender befähigt, verschiedene TM-Aufgaben zu lösen. Das TM-System soll sowohl für die Analyse einzelner Dokumente als auch zur Untersuchung großer Textmengen eingesetzt werden können.

Da diese Arbeit hauptsächlich dazu dient, die Grundfunktionalität zur Bedienung des TM-Systems bereitzustellen, soll bei der Umsetzung auf Erweiterbarkeit und Variabilität geachtet werden. Umgesetzt werden soll die Benutzeroberfläche als Webanwendung. Dies hat den Vorteil, dass die Anwendung von überall aus erreichbar ist. Zudem hat der Nutzer keinen Installationsaufwand und verwendet immer die aktuellste Version des Systems.

4 Verwandte Arbeiten

Dieses Kapitel gibt einen Einblick in die Benutzeroberflächen existierender TM-Systeme. Da eine vollständige Betrachtung aller TM-Systeme aufgrund der riesigen Anzahl im Rahmen dieser Arbeit nicht möglich ist, werden anhand einiger ausgewählter Beispiele grundlegende Aspekte verwandter Benutzeroberflächen betrachtet. Ausgehend von den in Kapitel 3 erhobenen Anforderungen werden dazu in Abschnitt 4.1 verschiedene Möglichkeiten zur Eingabe und Verwaltung von Textdaten präsentiert. Weiterhin wird in Abschnitt 4.2 gezeigt, wie der Nutzer in bestehenden Systemen seine Textanalyse definieren und konfigurieren kann. Abschnitt 4.3 geht auf die Darstellungsmöglichkeiten der TM-Ergebnisse ein. Zusätzlich werden in Abschnitt 4.4 verschiedene Ansätze zur Bewertung und Korrektur von TM-Ergebnissen untersucht. Als Beispiele dienen die folgenden Systeme, da sie einen Großteil der verschiedenen Ansätze abdecken:

- **BC-VisCon**: BC-VisCon ist ein service-orientiertes, speziell für den biomedizinschen Bereich entwickeltes TM-System zur Erkennung von Genen, Proteinen sowie deren Verbindungen innerhalb von wissenschaftlichen Abstracts biomedizinischer Artikel [SLVL09].
- **Wandora**: Mit Wandora können strukturierte Informationen aus unterschiedlichen Quellen extrahiert und zu Topic Maps verknüpft werden [Wan11].
- **Leximancer**: Diese webbasierte TM-Software dient zur konzeptbasierten Analyse großer Textsammlungen. Das System lässt sich für verschiedene Domänen einsetzen [Wan11].
- **GATE Developer:** Dieses System erlaubt es mit Hilfe von bestehenden Bausteinen speziell angepasste Softwarekomponenten zur Textanalyse zu erstellen und auszuwerten [CMBT02].
- **KNIME**: Der Konstanz Information Miner ist eine open-source Plattform zur Analyse und Exploration von strukturierten und unstrukturierten Daten [KNI11].
- **AdaptIE**: AdaptIE bietet eine Entwicklungsplattform zur Erstellung und Durchführung von IE-Aufgaben auf Basis bestehender Analysekomponenten [BFB10].
- **OpenCalais-Viewer**: Der OpenCalais-Viewer gibt Interessierten eine Möglichkeit die Funktionen des OpenCalais TM-Dienstes an einem Text auszupro-

bieren [Cal11a].

4.1 Dokumenteneingabe und -verwaltung

Die Eingabe und Verwaltung von Dokumenten gehört zu den Grundfunktionen eines jeden TM-Systems. Dieser Abschnitt zeigt, welche unterschiedlichen Ansätze dafür existieren.

In BC-VisCon können nur Dokumente aus einer vorgegebenen Dokumentendatenbank eingebunden werden. Abbildung 4.1 zeigt einen Ausschnitt der Benutzeroberfläche. Über das dargestellte Textfeld können die Nutzer mit Hilfe von Stichwörtern oder einer konkreten ID die Dokumentenquelle nach Dokumenten durchsuchen. Die Ergebnisse werden unterhalb des Eingabefelds anzeigt. Die zu analysierenden Dokumente müssen zunächst im Abstract Cart auf der linken Seite abgelegt werden. Daraus kann der Nutzer dann einzelne Dokumente zur Analyse auswählen.

Abbildung 4.1: Dokumentensuche in BC-VisCon

GATE bietet zur Auswahl von Dokumenten einen Dialog an. Über diesen können die Nutzer lokale Dokumente wie PDFs, Textdateien oder Word-Dokumente in das System laden. In Abbildung 4.2 ist der Dialog zur Auswahl einer Datei zu sehen.

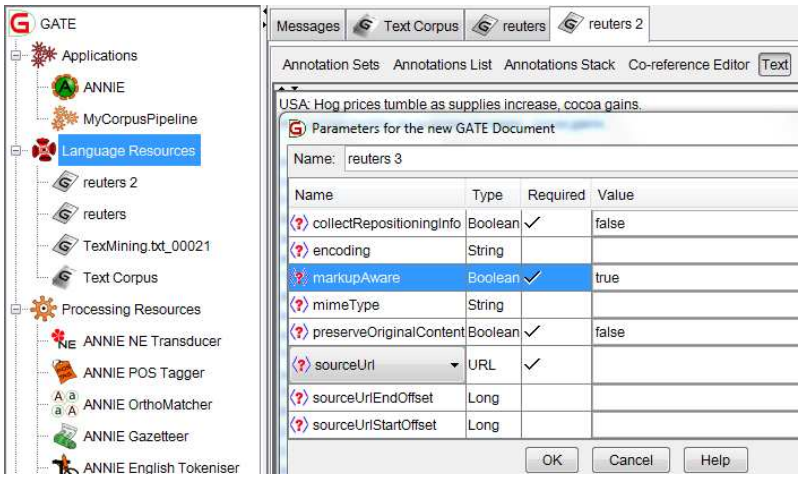


Abbildung 4.2: Dokumentenverwaltung in GATE

In dem Dialog können neben dem Dateipfad verschiedene Parameter wie beispiels-

weise ein Textausschnitt festgelegt werden. Aufgelistet werden die Textdaten auf der linken Seite unter Language Resources. Bevor eine Textanalyse durchgeführt werden kann, müssen die einzelnen Textdokumente zu einem Textkorpus hinzugefügt werden. Ein Textkorpus wird dabei über das Kontextmenü erstellt und erscheint auch unter Language Resources.

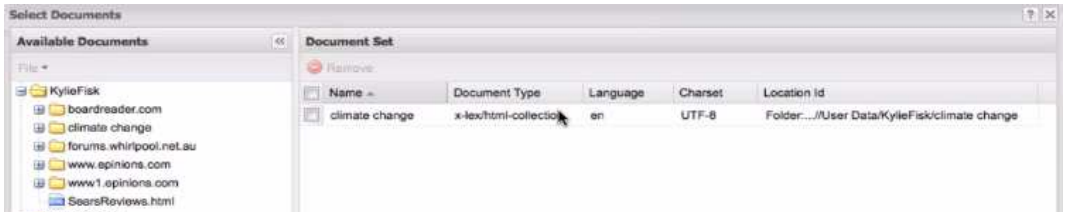


Abbildung 4.3: Dokumentenverwaltung in Leximancer

In Leximancer kann der Nutzer sowohl lokale Dateien als auch Webseiten einbinden. Alle eingebundenen Textdaten lassen sich, wie in Abbildung 4.3 dargestellt, in Ordnern gruppieren. Über eine Auswahlbox kann der Nutzer dann einzelne Ordner oder Dokumente zur Textanalyse festlegen. In dem dargestellten Beispiel könnte somit das Dokument „climate change“ ausgewählt werden.

Im OpenCalais-Viewer ist die Texteingabe sowie -verwaltung weniger umfangreich. Hier wird lediglich ein Textfeld bereitgestellt, in welches der Nutzer Text einfügen kann. Eine Verwaltung der Textdaten ist nicht möglich.

Eine weitere Variante, wie Dokumente ausgewählt werden können, kommt in KNIME zum Einsatz. Hier bestimmen Importoperatoren über die Dokumentequelle sowie, welche Dokumente ausgewählt werden können. Da bei diesem System der Ablauf des Analyseprozesses im Vordergrund steht, wird auf eine Funktion zur Verwaltung der Textdokumente verzichtet.

4.2 Aufgabendefinition und -konfiguration

Dieser Abschnitt analysiert, wie in den genannten TM-Systemen die Auswahl der durchzuführenden Aufgaben erfolgt und welche zusätzlichen Einstellungs- bzw. Konfigurationsmöglichkeiten dabei zur Verfügung stehen.

In BC-VisCon ist die TM-Aufgabe fest vorgegeben und umfasst immer die Erkennung von biomedizinschen Entitäten. Eine Auswahl ist daher nicht nötig. Durchgeführt werden die Aufgaben von zwölf voneinander unabhängigen Annotationsdiensten aus dem Bereich der Biomedizin, wobei alle Ergebnisse zu einem Gesamtergebnis aggregiert werden. Über Konsenseinstellungen, zu sehen in Abbildung 4.4, kann der Nutzer hierbei selbst festlegen, nach welcher Methode die Ergebnisse aggregiert werden. Als Standardeinstellung wird die Vereinigungsmenge aller Ergebnisse gebildet. Eine

weitere Option ist die Schnittmengenbildung. Bei dieser werden nur Entitäten zurückgegeben, die in der Ergebnismenge aller Dienste auftauchen. Des Weiteren kann der Konsens auch anhand eines Schwellenwertes oder durch ein Mehrheitsvoting gebildet werden. Beim Mehrheitsvoting wird die Gewichtung der Dienste herangezogen, welche der Nutzer über das in Abbildung 4.4 dargestellte Menü einstellen kann.

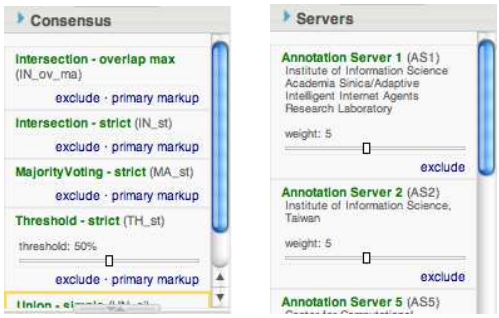


Abbildung 4.4: Konsens- und Gewichtungseinstellungen in BC-VisCon

Ein weiteres TM-System, welches externe Dienste zur Ausführung von IE- und TM-Aufgaben verwendet, ist Wandora. Im Gegensatz zu BC-VisCon bietet Wandora nicht nur Dienste zur Erkennung von Entitäten, sondern auch Dienste zur Klassifikation von Dokumenten an. Durch die Wahl eines entsprechenden Dienstes kann der Nutzer die TM-Aufgabe selbst bestimmen. Eine Aggregation der Ergebnisse erfolgt dabei nicht. Die Einstellungsmöglichkeiten beschränken sich daher auf die von den Diensten vorgegebenen Parameter. Ausgewählt werden die Dienste über die Menüleiste. Abbildung 4.5 zeigt die verschiedenen Auswahlmöglichkeiten.

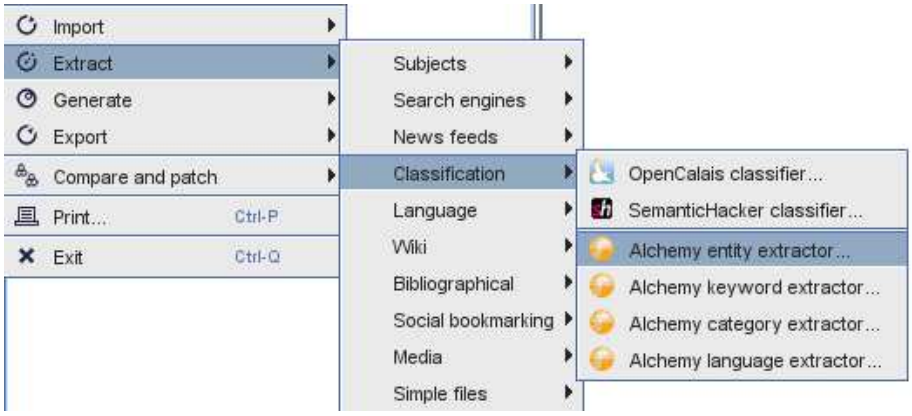


Abbildung 4.5: Auswahl der TM-Aufgabe in Wandora

Ähnlich wie in BC-VisCon ist auch bei Leximancer die TM-Aufgabe fest vorgegeben. Die Konfiguration der Textanalyse wird dabei als fest definierter Analyseprozess beginnend bei der Dokumenteneingabe bis hin zur Ergebnisauswertung dargestellt. In Abbildung 4.2 sind die verschiedenen Schritte des Analyseprozesses zu sehen. Für

jeden Schritt werden dem Nutzer verschiedene Konfigurationsmöglichkeiten angeboten.

Abbildung 4.6: Konfiguration der Textanalyse in Leximancer

Während bei Leximancer die einzelnen Analyseschritte noch fest vorgegeben sind, müssen bei KNIME und GATE diese selbst ausgewählt und zu einer Textanalysekette zusammengebaut werden. Ein Beispiel einer solchen Analysekette, wie sie in KNIME erstellt werden kann, ist in Abbildung 4.7 dargestellt. Bei diesem Beispiel wird aus den in einem Dokument vorkommenden Wörtern eine Tagcloud erstellt. Neben dem Wissen über die in den Dokumenten relevanten Informationen werden bei diesen Systemen auch Kenntnisse über die Funktionsweise von TM-Verfahren benötigt. Der Nutzer muss also nicht nur Domänenexperte, sondern auch TM-Experte sein. Dies stellt für viele Nutzer ein Problem dar. Ein System, was versucht dieses Problem zu lösen, ist AdaptIE.

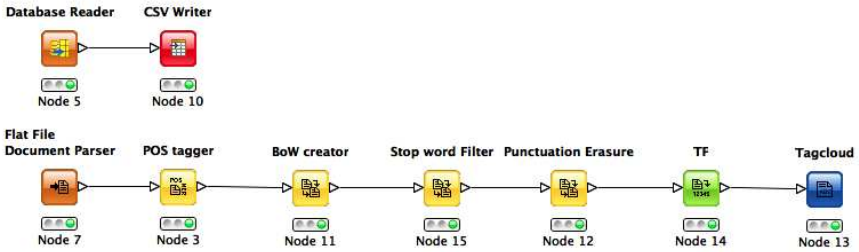


Abbildung 4.7: Analysekette in KNIME zur Erstellung einer Tagcloud

AdaptIE beschränkt sich auf die IE. Die Konfiguration der IE erfolgt über zwei Ansichten, dargestellt in Abbildung 4.8. In der ersten Ansicht können IE-Experten

ähnlich wie bei KNIME oder GATE domänenspezifische IE-Aufgaben aus einzelnen Operatoren zusammenbauen. Diese IE-Aufgaben werden dann von Domänenexperten genutzt, um in der zweiten Ansicht komplexe IE-Anwendungen zu erstellen.

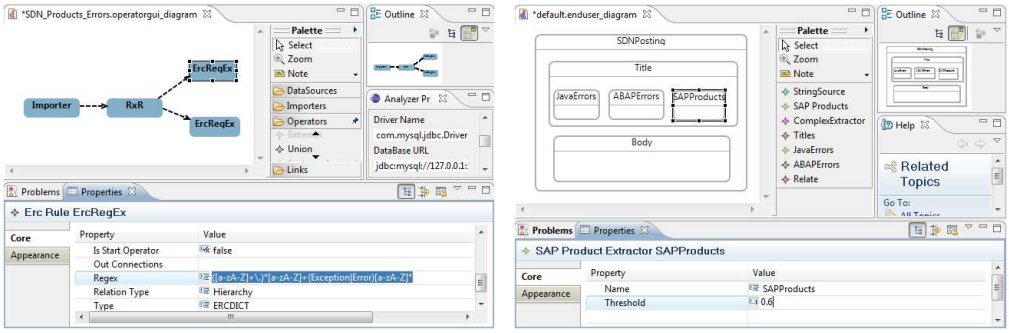


Abbildung 4.8: Ansichten für IE- und Domänen-Experten in AdaptIE

4.3 Ergebnisausgabe

Die dritte große Anforderung an die zu entwerfende Benutzeroberfläche umfasst die Ausgabe der Ergebnisse. Dieser Abschnitt erörtert die verschiedenen Ansätze verwandter Systeme. Generell muss bei der Ergebnisausgabe zwischen einer dokumenten- und konzeptorientierten Ansicht unterschieden werden.

4.3.1 Dokumentenorientierte Ansicht

In der dokumentenorientierten Ansicht werden nur die Ergebnisse eines Dokuments dargestellt. Eine Variante, wie die Ergebnisse dargestellt werden können, ist in Abbildung 4.9 zu sehen. Hierbei werden die erkannten Entitäten und Relationen direkt im Text hervorgehoben. Durch unterschiedliche Farben wird dabei auf den jeweiligen Typ einer Entität bzw. einer Textmarkierung hingewiesen. Beim OpenCalais-Viewer werden zudem verschiedene Zusatzinformationen angezeigt, sobald der Nutzer mit der Maus über ein Ergebnis fährt.

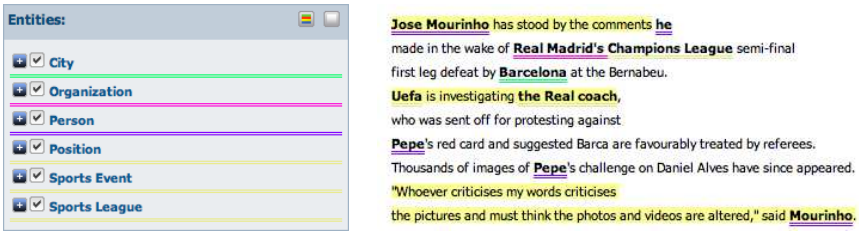


Abbildung 4.9: Ergebnisansicht im OpenCalais-Viewer

Eine Besonderheit bei der Darstellung von Entitäten stellt BC-VisCon dar. Wie in Abschnitt 4.2 bereits erwähnt, aggregiert das System die Ergebnisse der verschiedenen Dienste zu einem Gesamtergebnis. Die aggregierten Entitäten werden ähnlich wie bei anderen Systemen innerhalb des Textes farbig markiert (siehe Abbildung 4.10). Die Besonderheit dabei ist, dass unterhalb der Textmarkierungen einzelne Unterstreichungen die Ergebnisse der einzelnen Dienste visualisieren. Damit ist der Nutzer in der Lage die Ergebnisse der unterschiedlichen Dienste zu vergleichen.

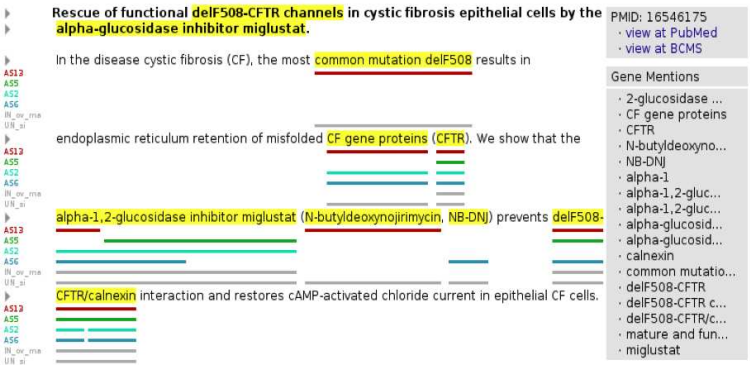


Abbildung 4.10: Annotiertes Dokument in BC-VisCon

Die Darstellung der Ergebnisse anderer TM-Aufgaben hängen vom entsprechenden System ab. Eine Möglichkeit ist, die Ergebnisse neben dem Text aufzulisten.

4.3.2 Konzeptorientierte Ansichten

Konzeptorientierte Ansichten dienen zur dokumentenübergreifenden Auswertung der Ergebnisse. Hierfür werden verschiedene Techniken der Daten- und Informationsvisualisierung eingesetzt, um dem Nutzer bei der Auswertung der TM-Ergebnisse zu helfen. Ein Beispiel einer solchen Ansicht ist Abbildung 4.11 zu sehen.

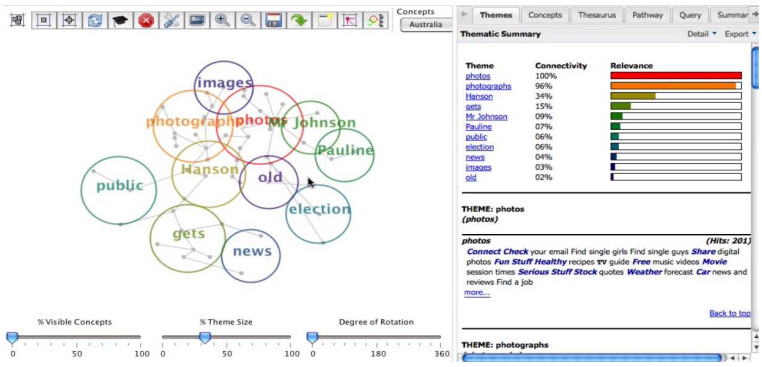


Abbildung 4.11: Beispiel einer Konzept-Karte aus Leximancer

Hierbei wird eine Konzeptkarte verwendet, um die Zusammenhänge zwischen den in

den Dokumenten auftauchenden Themen zu visualisieren. Mit Hilfe solcher Ansichten lassen sich Sachverhalte wie statistische Verteilungen oder bestimmte Zusammenhänge besser darstellen und wichtige oder neue Informationen effektiver ablesen.

4.4 Feedbackmöglichkeiten

In der zu entwerfenden Benutzeroberfläche spielt neben der Ausgabe auch die Bewertung und Korrektur der Ergebnisse eine wichtige Rolle. Die folgenden Abschnitte stellen die dafür gefundenen Ansätze vor.

4.4.1 Bewertung der Ergebnisse

Bei den betrachteten TM-Systemen wurden keine Funktionen zur Bewertung der Ergebnisse gefunden. Lediglich die Konsens- und Gewichtungseinstellungen in BC-VisCon können als eine Methode zur indirekten Bewertung der Analyseergebnisse angesehen werden.

Das Bewerten von Inhalten wird jedoch in vielen Web-Anwendungen wie Amazon oder YouTube angeboten und ist sogar in der UI Patterns Library [Tox11] als Entwurfsmuster zu finden. Anhand der dort gegebenen Beispiele wurden drei Varianten identifiziert. Eine erste Variante ist, den Inhalt über eine Skala zu bewerten. Eine zweite Variante besteht darin, den Inhalt entweder positiv oder negativ zu bewerten. Als dritte Variante kann auch nur eine positive Bewertung erlaubt sein. Beispiele für diese drei Varianten sind in Abbildung 4.12 zu finden.

Abbildung 4.12: Varianten zur Bewertung von Inhalten bzw. Ergebnissen aus [Tox11]

4.4.2 Korrektur der Ergebnisse

Wie die Bewertung der Ergebnisse wird auch die Korrektur der Ergebnisse in den meisten Systemen nicht unterstützt. Eine Ausnahme stellt GATE dar. GATE besitzt eine Annotationskomponente, mit welcher der Nutzer, wie in Abbildung 4.13 dargestellt, Annotationen im Text vornehmen kann. Dazu muss der entprechende Text markiert und mit einem Typen versehen werden.

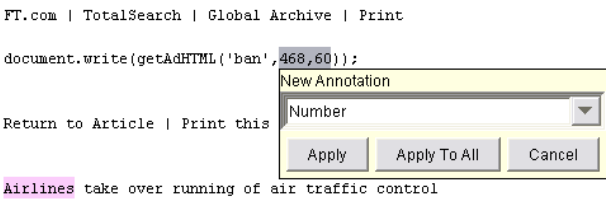


Abbildung 4.13: Annotationsmöglichkeiten in GATE

Ein weiterer Ansatz für eine Benutzeroberfläche zur Korrektur von IE-Ergebnissen wird in [CKMV06] beschrieben. Anders als in GATE kann der Nutzer hier keine eigenen Annotationen vornehmen, sondern nur die Typen der erkannten Entitäten bearbeiten.

4.5 Fazit

Die untersuchten Arbeiten stellen nur einen kleinen Ausschnit der verfügbaren TM-Systeme dar. Trotzdem werden verschiedene Ansätze für die jeweiligen Funktionen deutlich. So zeigt sich, dass eine Dokumentenverwaltung, obwohl diese sehr praktisch ist, nicht in allen Systemen vorhanden ist. Des Weiteren werden verschiedene Varianten zur Konfiguration der Textanalyse erkennbar. So können die TM-Aufgaben fest vorgegeben sein, frei auswählbar oder müssen selbst vom Nutzer zusammengebaut werden. Die Ausgabe der jeweiligen Ergebnisse ist immer abhängig von der Menge der untersuchten Dokumente sowie der Intention der Ansicht. Die Feedbackmöglichkeiten der Systeme halten sich stark in Grenzen. Tabelle 4.5.1 fasst nochmal die verschiedenen Aspekte der untersuchten Systeme zusammen.

	BC-VisCon	**Wandora**	**Leximancer**	**AdaptIE**	**GATE**	**OpenCalais-Viewer**
Dok. - quelle	Datenbank	Webseiten, Lokale Dateien, Textfeld	Webseiten, Lokale Dateien	frei	Lokale Dateien	Textfeld
Dok. - verwaltung	Abstract Cart	Keine	Ordner	keine	Textsammlungen	keine
Aufgaben-definition	fest vorgegeben	durch Auswahl eines Dienstes	fest vorgegeben	flexibel	flexibel	fest vorgegeben
Verwendet externe Dienste	ja	ja	nein	nein	möglich	ja
Ausgabe der Ergebnisse	Liste, Markierungen	Baum, Konzeptkarte	Listen, Karten	Tabelle	Tabelle, Graphisch	Liste, Markierungen
Feedback-möglich-keiten	Bewertung der Dienste	keine	keine	keine	Korrektur der Ergebnisse der IE	keine

Tabelle 4.5.1: Vergleich der untersuchten TM-Systeme

5 Entwurf

Dieses Kapitel zeigt, welche Überlegungen beim Entwurf der Benutzeroberfläche angestellt wurden. Im ersten Abschnitt wird dazu der Arbeitsabaluf des Nutzers betrachtet sowie das sich daraus ergebende Anwendungslayout. Anschließend werden in Abschnitt 5.2 und 5.3 die Entwürfe der Hauptansichten beschrieben.

5.1 Arbeitsablauf und allgemeines Layout

Abbildung 5.1 zeigt den typischen Arbeitsablauf des Nutzers zur Durchführung einer Textanalyse. Generell muss davon ausgegangen werden, dass der Nutzer mehrere Dokumente auf einmal analysieren möchte. Er lädt dafür die entsprechenden Dokumente in das Programm, konfiguriert seine Textanalyse, führt diese aus und lässt sich die Ergebnisse ausgeben.

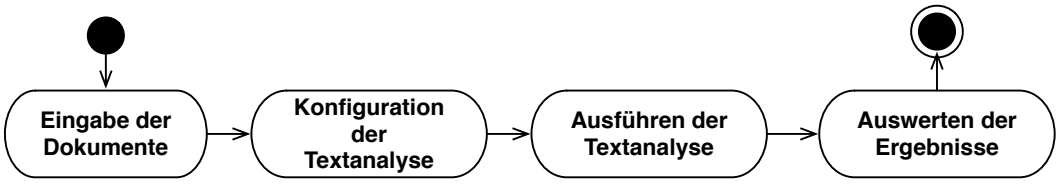


Abbildung 5.1: Allgemeiner Arbeitsablauf für eine Textanalyse

Folgende Fälle müssen dabei berücksichtigt werden:

1. Der Nutzer möchte nach einer gewissen Zeit die bereits konfigurierte Textanalyse für weitere Dokumente verwenden. Er will jedoch die schon betrachteten Dokumente nicht erneut analysieren.
2. Dem Nutzer fällt auf, dass er noch weitere Informationen aus den bereits analysierten Dokumenten benötigt.
   Er möchte die bestehende Konfiguration jedoch nicht verändern.
3. Der Nutzer hat eine Textanalyse durchgeführt, möchte die Analyseergebnisse jedoch erst später auswerten.

Wie bereits in FA 1.2 gefordert, ist es für den ersten Fall notwendig, dass der Nutzer verschiedene Dokumentensammlungen anlegen kann, sodass er die schon analysierten

von den noch zu analysierenden Dokumente trennen kann. Die Eingabe der Dokumente wird daher wie in Abbildung 5.2 zu sehen, in die Teilschritte „Erstellen / Auswählen einer Dokumentensammlung“ und „Importieren / Löschen von Dokumenten“ unterteilt. Dadurch lassen sich die Dokumente auch inhaltlich besser ordnen. Des Weiteren muss es möglich sein, dass eine Textanalysekonfiguration für verschiedene Dokumentensammlungen verwendet werden kann. Die Konfiguration sollte daher nicht strikt an eine spezielle Dokumentensammlung gebunden sein.

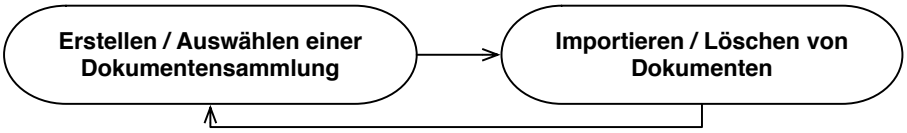


Abbildung 5.2: Ablauf zur Eingabe von Dokumenten

Der zweite Fall erfordert, dass der Nutzer verschiedene Konfigurationen anlegen kann. Hierfür sollen Textanalyseprofile dienen. Entsprechend des Arbeitsablaufs in Abbildung 5.3 wird der Schritt „Konfiguration der Textanalyse“ daher in die Teilschritte „Anlegen / Auswählen eines Textanalyseprofils“, „Konfigurieren eines Textanalyseprofils“ und „Ausführen eines Textanalyseprofils“ gegliedert.

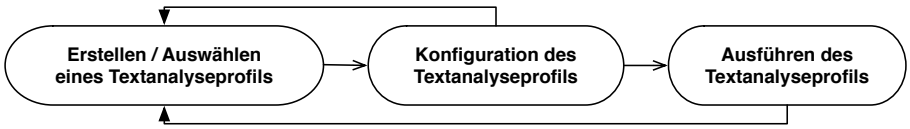


Abbildung 5.3: Ablauf zur Konfiguration der Textanalyse

Beim dritten Fall wird davon ausgegangen, dass der Nutzer seine Arbeit nach der durchgeführten Textanalyse abbricht und später an dieser Stelle wieder aufnehmen will. Das Gleiche kann bei der Konfiguration der Textanalyse oder der Eingabe der Dokumente passieren. Jeder Arbeitsablauf beginnt also mit einem der drei genannten Arbeitsobjekte (Dokumentensammlung, Textanalyseprofil oder Analyseergebnisse). Da das System nicht weiß, mit welchem Objekt der Nutzer anfangen will, sollten alle verfügbaren Arbeitsobjekte von Beginn an in einem zentralen Bereich zu finden sein.

Das Layout, dargestellt in Abbildung 5.4, wird daher nach dem „Übersicht-Detail “-Muster in zwei Bereiche geteilt. Der Übersichtsbereich (5.4.A) enthält alle Arbeitsobjekte. Der größere, rechte Bereich des Layouts (5.4.B) dient zur Darstellung der Details eines ausgewählten Arbeitsobjektes. Die folgenden Abschnitte erläutern die Entwürfe dieser beiden Bereiche.

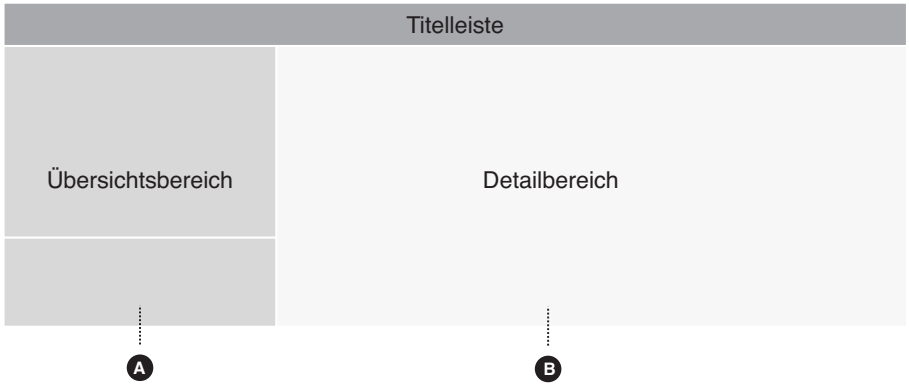


Abbildung 5.4: Anwendungslayout

5.2 Übersichtsbereich

Der Übersichtsbereich soll dem Nutzer einen schnellen Zugriff auf alle vorhandenen Arbeitsobjekte bieten. Da in der Höhe relativ viel Platz zur Verfügung steht, soll dieser Bereich noch eine weitere Aufgabe erfüllen.

Alle Prozesse, wie zum Beispiel die Abarbeitung von Analyseaufgaben oder das Laden von Dateien, nehmen eine bestimmte Zeit in Anspruch. Damit der Nutzer trotz dieser Aktivitäten weiterarbeiten kann, sollen diese im Hintergrund ausgeführt werden. Entsprechend der dritten Regel von Shneiderman (S. 10) sollte dabei erkennbar bleiben, inwieweit die einzelnen Aktivitäten vorangeschritten sind. Der Übersichtsbereich wird daher vertikal in einen Arbeits- und einen Aktivitätsbereich unterteilt.

5.2.1 Arbeitsbereich

Abbildung 5.5 zeigt die Ansicht des Arbeitsbereiches. Alle hier enthaltenen Objekte werden, wie nach FA 1.3, FA 2.9 und FA 3.5 gefordert, persistent im System gespeichert. Damit der Nutzer die einzelnen Arbeitsobjekte besser unterscheiden kann, werden diese hierarchisch in die Gruppen Dokumentensammlungen, Textanalyseprofile sowie Textanalyseergebnisse gegliedert sowie durch vorangestellte Icons gekennzeichnet. Jedes Objekt erhält zu dem einen Namen, der vom Nutzer selbst festgelegt werden kann. Bei den Dokumentensammlungen dient die Anzahl der enthaltenen Dokumente als zusätzliches Unterscheidungsmerkmal.

Wenn der Nutzer die Anwendung zum ersten Mal startet, ist der Arbeitsbereich noch leer. Der Nutzer muss also zunächst eine Dokumentensammlung sowie ein Analyseprofil anlegen. Um diese Objekte zu erstellen, werden ihm in der Werkzeugleiste (5.5.A) zwei Buttons zur Verfügung gestellt. Der erste Button dient zum Erstellen einer Dokumentensammlung. Mit Hilfe des zweiten Buttons kann ein neues Textana-

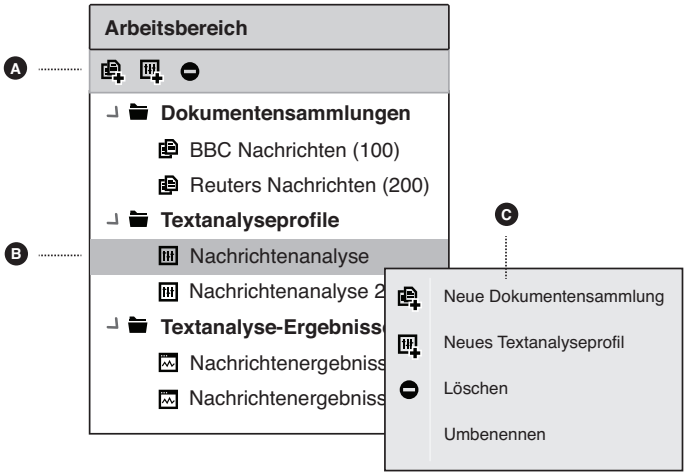


Abbildung 5.5: Entwurf des Arbeitsbereichs

lyseprofil erstellt werden. Klickt der Nutzer auf einen dieser Buttons, wird er nach dem Namen für das jeweilige Objekt gefragt. Auf eine Funktion zum Erstellen eines Textanalysergebnisses wird mit Absicht verzichtet, da dieses automatisch nach dem Ausführen einer Textanalyse erstellt und hinzugefügt wird. Damit die einzelnen Objekte auch entfernt werden können, wird in der Werkzeugleiste zusätzlich ein Button zum Löschen des aktuell ausgewählten Objekts (5.5.B) bereitgestellt.

Nach der Regel der universellen Benutzerfreundlichkeit (S. 10) ist es sinnvoll, wenn den Nutzern verschiedene Möglichkeiten zum Zugriff auf eine Funktion gegeben werden. Es wird daher zusätzlich ein Kontextmenü (5.5.C) zur Verfügung gestellt, was durch einen Rechtsklick im Arbeitsbereich aufgerufen wird. Das Menü bietet dem Nutzer die selben Funktionen wie die Werkzeugleiste und enthält zudem eine Funktion zum Umbenennen der einzelnen Objekte.

Der Arbeitsbereich erlaubt somit einen schnellen und konsistenten Zugriff auf alle grundlegenden Arbeitsobjekte. Die Details und Einstellungen zu den einzelnen Arbeitsobjekten werden, wie bereits erwähnt, im Detailbereich dargestellt. Bevor auf die dafür notwendigen Ansichten eingegangen wird, erfolgt die Betrachtung des Aktivitätsbereichs.

5.2.2 Aktivitätsbereich

Der Aktivitätsbereich dient dazu alle momentan laufenden Aktivitäten aufzulisten. Der Entwurf dieser Ansicht ist in Abbildung 5.6 dargestellt. Damit der Nutzer die einzelnen Aktivitäten auseinander halten kann, erhält jede Aktivität entsprechend ihres Typs einen eigenen Namen (5.6.A). Aktivitäten zum Laden von Dokumenten werden gekennzeichnet durch „Dokumentenimport in ...“ und den Namen der entsprechenden

Dokumentensammlung. Laufende Textanalysen erhalten den Namen des jeweiligen Textanalyseprofils. Um den Fortschritt jeder Aktivität besser zu veranschaulichen, wird ein Fortschrittsbalken (5.6.B) eingesetzt. Damit der Nutzer zudem sieht, wie viel Zeit eine Aktivität noch benötigt, wird unterhalb des Fortschrittbalkens die zu verbleibende Ausführungsdauer (5.6.C) angezeigt. Zusätzlich werden die Aktivitäten nach diesem Wert sortiert, sodass immer die als nächstes fertig werdende Aktivität an oberster Stelle steht.

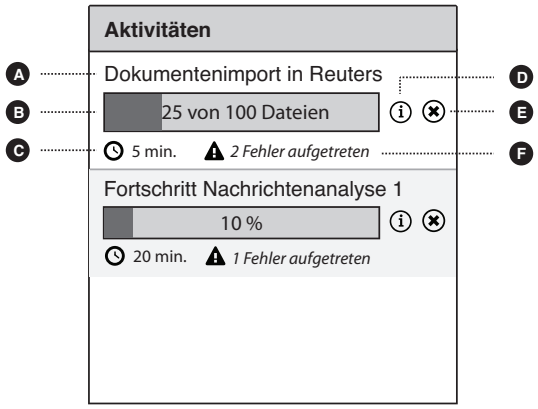


Abbildung 5.6: Entwurf des Aktivitätsbereichs

Zu beachten ist, dass während der Durchführung der Aktivitäten Fehler auftreten können. Zum Beispiel kann es beim Importieren von Textdokumenten dazu kommen, dass beschädigte Dateien nicht gelesen werden können. Der Nutzer sollte auf solche Fehler aufmerksam gemacht werden. Da die Auflistung der genauen Fehlerursachen die Ansicht zu unübersichtlich machen würde, wird neben der verbleibenden Ausführungsdauer nur ein Vermerk über die Anzahl der aufgetretenen Fehler (5.6.F) vorgenommen. Will sich der Nutzer die genauen Fehlerursachen ansehen, kann er über den Info-Button (5.6.D) zur Detailansicht der Aktivität wechseln.

Ist eine Aktivität abgeschlossen, soll diese nicht einfach verschwinden, sondern zunächst an oberster Stelle der Liste verbleiben. Somit kann der Nutzer nach Abschluss nochmals die Details der Aktivität aufrufen. Um eine abgeschlossene Aktivität letztendlich aus der Liste zu entfernen, wird ein Schließen-Button (5.6.E) bereitgestellt. Dieser Button dient gleichzeitig zum Abbrechen von noch laufenden Aktivitäten. Damit der Nutzer nicht aus Versehen eine Aktivität beendet, muss jeder Abbruch erst bestätigt werden.

5.3 Detailbereich

Im Detailbereich sollen die Ansichten zur Betrachtung und Bearbeitung der Arbeitsobjekte dargestellt werden. Wie in Abbildung 5.7 zu sehen, war der erste Entwurf

dafür so angelegt, dass jedes Arbeitsobjekt in einem eigenen Tab geöffnet wird. Die Intention dabei war, dass der Nutzer, wie im Entwurfsmuster „Parallele Ansichten “ beschrieben, schneller zwischen bereits geöffneten Arbeitsobjekten wechseln kann. Abbildung 5.7 zeigt dies an einem Beispiel mit zwei geöffneten Dokummtensammlungen und einem Textanalyseprofil. Es wurde jedoch festgestellt, dass Tabs an dieser Stelle unnötig sind, da zum einen auch über den Arbeitsbereich auf die gewünschten Objekte zugegriffen werden kann und zum anderen die Wahrnehmeung weiterer Tabs in den Ansichten selbst eingeschränkt wird.

Abbildung 5.7: Erster Entwurf des Detailbereichs

Aufgrund dieser Überlegungen wurden die Tabs verworfen und ein anderer Ansatz verfolgt, bei dem nur das gerade ausgewählte Arbeitsobjekt im Detailbereich angezeigt wird. Der Entwurf dafür ist in Abbildung 5.8 zu sehen. Ein großer Vorteil dieses Entwurfs ist, dass sich der Nutzer so besser auf ein Arbeitsobjekt konzentrieren kann.

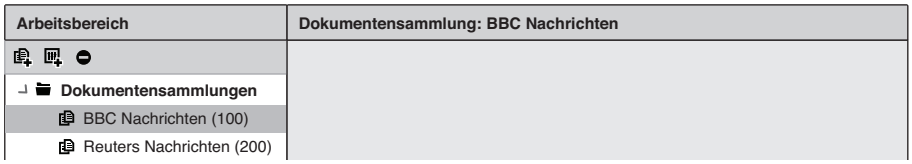


Abbildung 5.8: Zweiter Entwurf des Detailbereichs

Wird die Anwendung das erste Mal gestartet, sind noch keine Arbeitsobjekte wie Dokumentensammlungen, Textanalyseprofile oder Ergebnisse vorhanden. Der Detailbereich ist folglich leer. Nach dem Entwurfsmuster „Leerer Status “ sollte dieser Platz genutzt werden, um neuen Nutzern die Funktionen der Anwendung näher zu bringen. Neben einer Willkommensnachricht wird daher eine kurze Anleitung zur Verwendung der Anwendung angezeigt. Damit der Nutzer sofort mit der Bedienung anfangen kann, werden für die ersten Schritte wie dem Erstellen einer Dokumentensammlung oder eines Textanalyseprofils entsprechende Buttons (5.9.A und 5.9.B) bereitgestellt. Die Buttons werden dabei mit den im Arbeitsbereich verwendeten Icons ausgestattet, sodass der Nutzer die Bedeutung der Icons lernt.

Nachdem der grundlegende Aufbau des Detailbereichs erklärt ist, werden in den nächsten Abschnitten die Entwürfe zur Betrachtung der genannten Arbeitsobjekte

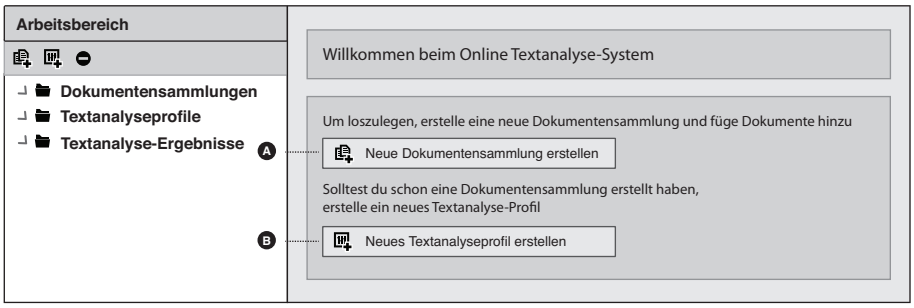


Abbildung 5.9: Willkommensansicht

erläutert. Dem Arbeitsablauf folgend, wird als erstes der Entwurf für die Ansicht der Dokumentensammlung betrachtet.

5.3.1 Ansicht der Dokumentensammlung

Entsprechend der Anforderung FA 1.2 müssen in dieser Ansicht als erstes die enthaltenen Dokumente sichtbar werden. Abbildung 5.10 zeigt die dafür entworfene Ansicht. Da zu jedem Dokument verschiedene Metainformationen bereitstehen, wird hierfür das Entwurfsmuster „Tabellen “ angewandt. Neben dem Dateinamen werden der Typ, die Größe sowie ein Ausschnitt aus der Datei aufgelistet. Damit können die Dokumente schneller wiedererkannt werden. Des Weiteren ist nach FA 1.4 gefordert, dass Textdokumente annotiert werden können. Damit der Nutzer weiß, welche Dokumente annotiert sind, wird eine entsprechende Spalte hinzugefügt.

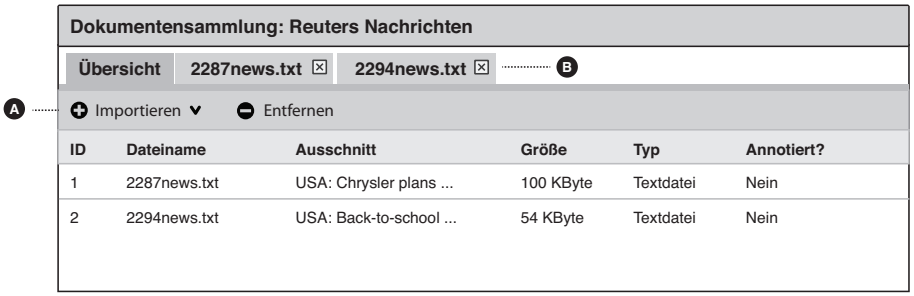


Abbildung 5.10: Ansicht einer Dokumentensammlung

Zu Beginn ist jede Dokumentensammlung leer. Wie nach FA 1.1 gefordert, müssen erst Dokumente hinzugefügt werden. Hierfür wird im Kopfbereich der Tabelle ein Importieren-Button (5.10.A) bereitgestellt. Dieser Button öffnet ein Untermenü, in welchem der Import-Dialog für eine bestimmte Dokumentenquelle aufgerufen werden kann. Dahinter steckt folgende Überlegung. Das System soll später noch so erweitert werden, dass auch Dokumente aus Nachrichtenfeeds, Suchmaschinen oder sozialen Plattformen eingebunden werden können. Jede Quelle verlangt dabei einen

speziell angepassten Dialog zur Auswahl der Dokumente. Zum Beispiel wäre es zur Einbindung von Suchmaschinenergebnissen notwendig, die gewünschte Suchmaschine festzulegen, ein Suchwort einzugeben sowie die zu importierenden Suchergebnisse auszuwählen. Diese drei Funktionen sind jedoch bei der Auswahl von lokalen Dateien völlig überflüssig, da hier nur die Dateien selbst ausgewählt werden müssen. Für jede Quelle soll daher ein speziell angepasster Dialog bereitgestellt werden, den der Nutzer, wie in Abbildung 5.11 dargestellt, über das Untermenü aufrufen kann.

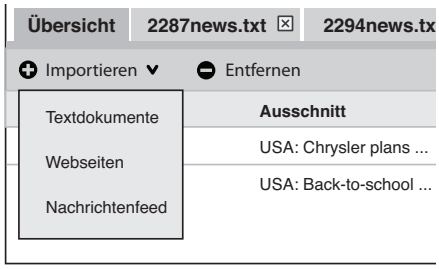


Abbildung 5.11: Untermenü „Importieren“

Um einzelne Dokumente auch wieder zu löschen, wird neben dem Importieren- ein Entfernen-Button bereitgestellt, der alle selektierten Elemente löscht. In FA 1.1 wird weiter verlangt, dass sich der Nutzer den Inhalt eines Dokuments ansehen kann. Wie von den meisten Anwendungen gewohnt, soll er dazu das entsprechende Dokument doppelt anklicken.

Für die Darstellung der Dokumenteninhalte wurden verschiedene Möglichkeiten betrachtet. Eine erste Möglichkeit ist, den gesamten Bereich horizontal nach dem Entwurfsmuster „Übersicht-Detail “ in einen Übersichts- und Ansichtsbereich zu unterteilen. Durch diese Aufteilung werden jedoch nicht nur der Übersichts-, sondern auch der Ansichtsbereich sehr eingeschränkt, wodurch nur wenig Platz für den Inhalt eines Dokuments bleibt.

Um diesem Problem aus dem Weg zu gehen, wäre eine zweite Möglichkeit, den Inhalt der Dokumente in einem Popup-Fenster zu präsentieren. Durch ein Popup wird jedoch die Tabelle der Dokumente verdeckt werden, was nachteilig ist, wenn der Nutzer mehrere Dokumente öffnen will. Außerdem wird die gesamte Ansicht sehr unübersichtlich, sobald mehrere Fenster geöffnet sind.

Es wurde daher ein dritter Entwurf verfolgt, bei dem die Inhalte der einzelnen Dokumente, wie in Abbildung 5.10 zu sehen, in Tabs dargestellt werden (5.10.B) . Wählt der Nutzer ein Dokument aus der Übersicht aus, öffnet sich ein neuer Tab, in dem der Inhalt des Dokuments präsentiert wird. Somit können mehrere Dokumente gleichzeitig geöffnet sein, ohne dass die Ansicht zu sehr eingeschränkt oder unübersichtlich wird. Damit der Nutzer die Dokumente auch wieder schließen kann, wird in jedem

Tab ein entsprechender Schließen-Button (5.10.B) zur Verfügung gestellt. Durch die Tabs wird ein schnelles Umschalten zwischen den einzelnen Dokumenten sowie der Dokumentenübersicht ermöglicht. Zudem wird bei dieser Variante fast der komplette Platz des Bildschirms ausgenutzt, was zum einen für lange Text sinnvoll ist und zum anderen Möglichkeiten für Erweiterungen bietet.

Hat der Nutzer eine Dokumentensammlung erstellt und mit den gewünschten Dokumenten gefüllt, soll er über die im nächsten Abschnitt betrachtete Ansicht seine Textanalyse konfigurieren.

5.3.2 Ansicht des Textanalyseprofils

In der Ansicht für das Textanalyseprofil soll der Nutzer die Textanalyse konfigurieren können. Bevor der Nutzer wie nach FA 2.2 die gewünschten TM-Aufgaben auswählt, sollen in einem Textanalyseprofil die zu untersuchenden Dokumentensammlungen festgelegt werden (siehe FA 2.1). Dies hat folgende Gründe.

Nach FA 2.8 sollen schon während der Konfiguration die Kosten für die Textanalyse ersichtlich werden. Da sich der Kostenbetrag aus der Anzahl der zu untersuchenden Dokumente sowie den Preisen der jeweiligen Dienstanbieter ergibt, müssen zuerst die Dokumente bzw. Dokumentensammlungen festgelegt werden. Ein weiterer Grund ist die Sprachabängigkeit der TM-Dienste. Mit Hilfe einer automatischen Sprachanalyse im Hintergrund kann das System somit unpassende TM-Dienste von vornherein aussortieren.

Abbildung 5.12 zeigt den ersten Entwurf für die Konfigurationsansicht des Textanalyseprofils. Die Idee dafür lieferte der Einstellungsdialog von Eclipse[1]. Wie in Abbildung 5.12 zu sehen ist, teilt sich die Ansicht in drei Spalten. In der linken Spalte (5.12.A) sind die verschiedenen Gruppen von Einstellungen wie Dokumentensammlungen, Textanalayseaufgaben und allgemeine Einstellungen aufgelistet. Wählt der Nutzer eine dieser Gruppen aus, hat er im Mittelteil (5.12.B) die Möglichkeit die entsprechenden Einstellungen dafür vorzunehmen. In der rechten Spalte (5.12.C) werden Zusatzinformationen wie Kosten, ungefähre Dauer und die Anzahl der Dienste angezeigt.

Auch wenn es möglich ist, die Textanalyse über diese Ansicht zu konfigurieren, weist dieser Entwurf mehrere grundlegende Nachteile auf. Ein erster Nachteil ist, dass die Ansicht zu viele Auswahlmöglichkeiten bietet, wodurch unerfahrene Nutzer überfordert wären. Des Weiteren wird der Nutzer bei den Einstellungen nicht angemessen geleitet. Er könnte somit wichtige Einstellungen vergessen oder gar nicht erst wahrnehmen. Ein dritter Nachteil ist, dass in der Spalte für die Zusatzinformationen die

[1] http://www.eclipse.org/

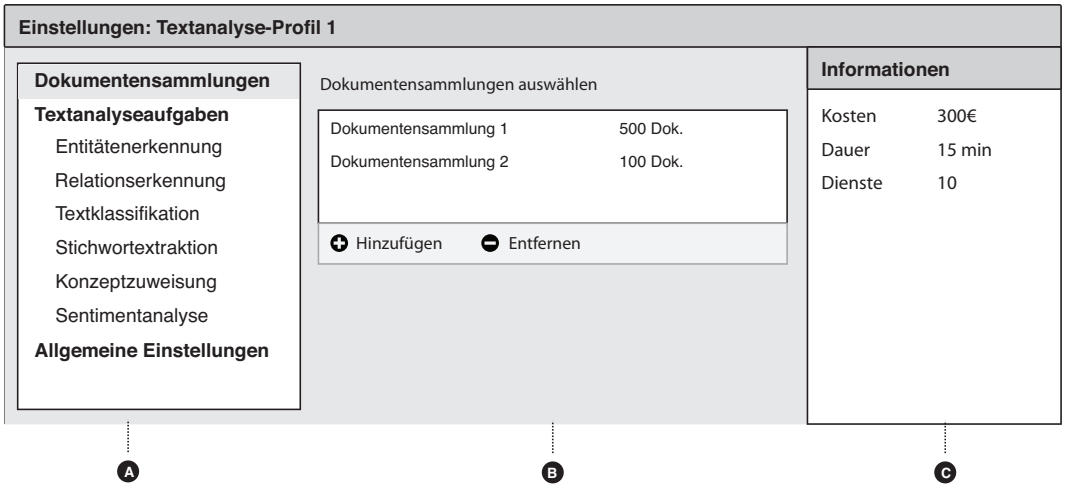


Abbildung 5.12: Erster Entwurf für die Ansicht des Textanalyseprofils

Kosten, die Dauer und die Anzahl der Dienste nicht genau genug aufgeschlüsselt werden. Nutzer, die sich dafür interessieren, würde dies nicht zufriedenstellen. Der Entwurf wurde daher nochmals überarbeitet. Das Ergebnis ist in Abbildung 5.13 zu sehen. Das Ziel bei diesem Entwurf war es die Auswahlmöglichkeiten zu verringen und den Nutzer angemessen zu leiten.

Wie zu Beginn erklärt, soll der erste Schritt darin bestehen, die zu analysierenden Dokumentensammlungen festzulegen. Diese Auswahl wird daher an oberster Stelle (5.13.A) platziert. Durch das Klicken auf den Hinzufügen-Button können die Dokumentensammlungen über ein Popup-Fenster ausgewählt werden. Um ausgewählte Dokumentensammlungen auch wieder zu löschen, wird zusätzlich ein Löschen-Button bereitgestellt, der alle in der Liste selektierten Elemente entfernt.

Im zweiten Schritt soll der Nutzer die gewünschten TM-Aufgaben auswählen. Da der Begriff TM für unerfahrene Nutzer verwirrend ist, wird die Bezeichnung Textanalyseaufgaben verwendet. Um die Bedienung bei der Aufgabenauswahl konsistent zu halten, wird auch hier ein Hinzufügen-Button eingesetzt. Klickt der Nutzer auf den Button, erscheint ein Popup-Fenster, in welchem die verschiedenen TM-Aufgaben ausgewählt werden können. Da unerfahrene Nutzer mitunter nicht wissen, was die einzelnen Aufgaben bedeuten, wird jede TM-Aufgabe mit einer Beschreibung aufgelistet. Abbildung 5.14 zeigt den Dialog dafür. Die vom Nutzer bereits ausgewählten TM-Aufgaben werden nicht mehr mit angezeigt.

Sind die TM-Aufgaben und die zu analysierenden Dokumente ausgewählt, kann das System die Kosten der Textanalyse berechnen. Da die Kosten sowie die Ausführungsdauer direkt von den Aufgaben abhängen, wird hinter jeder ausgewählten TM-Aufgabe die Anzahl der passenden Dienste, die zu erwartenden Kosten und ungefähre Ausführungsdauer aufgeführt (5.13.C) . Damit der Nutzer zudem sieht, was

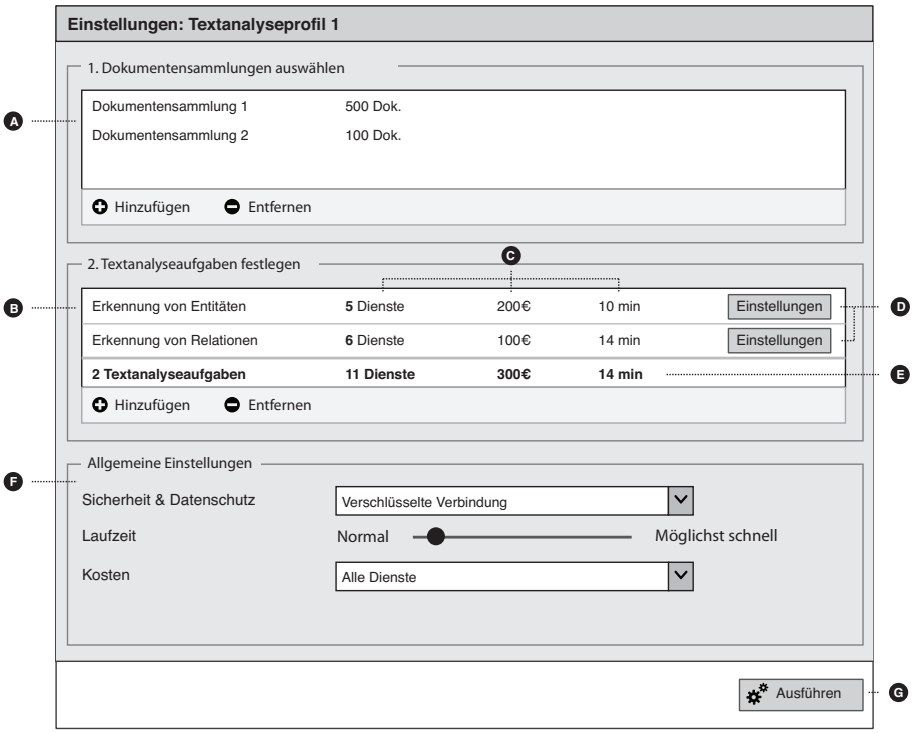


Abbildung 5.13: Finaler Entwurf für die Ansicht des Textanalyseprofils

die Textanalyse insgesamt kostet und wie lange diese dauert, wird unterhalb der TM-Aufgaben eine Zusammenfassung (5.13.E) angezeigt.

Des Weiteren muss es nach FA 2.3 möglich sein, einzelne TM-Aufgabe zu konfigurieren. Es wird daher für jede TM-Aufgabe ein Einstellungsbutton (5.13.D) bereitgestellt, der ein Popup-Fenster öffnet, in dem die aufgabenspezifischen Einstellungen vorgenommen werden können. Der Entwurf dieser Ansicht wird in Abschnitt 5.3.2 erläutert.

Nach FA 2.6 ist weiter gefordert, dass der Nutzer die Dienste, welche keine sichere Verbindung verwenden, ausschließen kann. Auch Dienste, die Daten bei sich speichern, sollen vermieden werden können. Beide Optionen beziehen sich auf das Sicherheitslevel und werden daher zu folgenden drei Optionen zusammengefasst:

1. Alle Verbindungen
2. Nur sichere Verbindungen
3. Nur sichere Verbindungen und kein Speichern der Daten

Um Platz in der Höhe zu sparen, wird für die Optionen eine Combobox verwendet.

Die zweite Anforderung (FA 2.7) der allgemeinen Einstellungen bezieht sich auf die Kosten. Der Nutzer soll festlegen können, dass für die gesamte Textanalyse nur kostenfreie Dienste verwendet werden. Dementsprechend wird ihm über eine weitere Combobox die Wahl zwischen kostenfreien und allen weiteren Diensten gestellt.

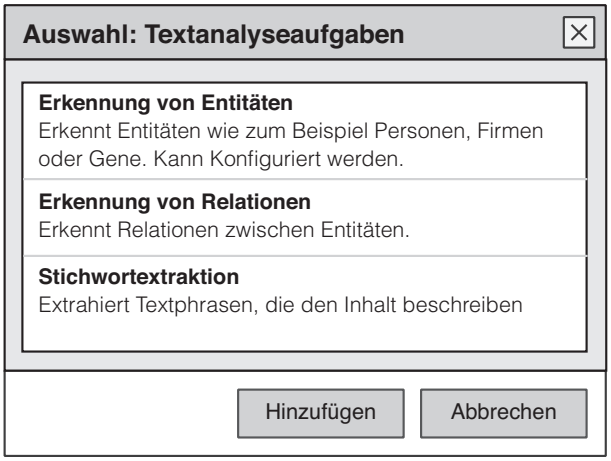


Abbildung 5.14: Dialog zur Auswahl der TM-Aufgaben

Entsprechend der Anforderung FA 2.5 soll der Nutzer zusätzlich einstellen können, dass das System schnelle Dienste bevorzugt. Da die Laufzeit davon abhängt, wie viele Transaktionen ein Dienst pro Sekunde abarbeiten kann, wird dem Nutzer hierfür ein Schieberegler zur Verfügung gestellt. Indem der Nutzer den Regler in Richtung „Möglichst Schnell“ verschiebt, erhöht er die minimale Anzahl an Transaktionen pro Sekunde und schließt somit langsame Dienste schrittweise aus.

Wie in Abbildung 5.13 zu sehen ist, werden die eben beschriebenen Einstellungsmöglichkeiten im unteren Drittel (5.13.F) angezeigt. Somit werden alle notwendigen Einstellungen in einer Übersicht präsentiert.

Sind alle Einstellungen getroffen, will der Nutzer die Textanalyse ausführen. Hierfür wird an unterster Stelle ein Ausführen-Button (5.13.G) platziert. Da es durch die Textanalyse zu Kosten für den Nutzer kommen kann, muss jede Ausführung erst durch einen Dialog bestätigt werden. Sobald diese bestätigt ist, wird wie in Abschnitt 5.2.2 beschrieben, eine neue Aktivität im Aktivitätsbereich angezeigt und die Textanalyse durchgeführt.

Um den Entwurf dieser Ansicht abzuschließen, werden in den folgenden Abschnitten die Popup-Fenster für die aufgabenspezifischen Einstellungen betrachtet.

Ansicht der aufgabenspezifischen Einstellungen

In Abbildung 5.15 ist das Fenster für die aufgabenspezifischen Einstellungen dargestellt. Dieses muss je nach Aufgabe verschiedene Optionen bereitstellen. Um die unterschiedlichen Optionen inhaltlich voneinander zu trennen, werden Tabs (5.15.A) verwendet. Im ersten Tab finden sich die allgemeinen Einstellungen zu einer Aufgabe. Hier kann der Nutzer, wie in Abschnitt 3.3.2 gefordert, über einen Schieberegler (5.15.B) den minimalen Konfidenzwert festlegen.

Unterhalb dieser Einstellung befindet sich die Liste der für die TM-Aufgabe verfügbaren Dienste (5.15.C). Diese wurde hinzugefügt, weil erfahrene Nutzer mitunter wissen wollen, von welchen Diensten die einzelnen TM-Aufgaben ausgeführt werden. Die Nutzer, die sich bereits besser mit den Diensten auskennen, bekommen zudem die Möglichkeit über die Auswahlboxen einzelne Dienste aus- und anzuschalten.

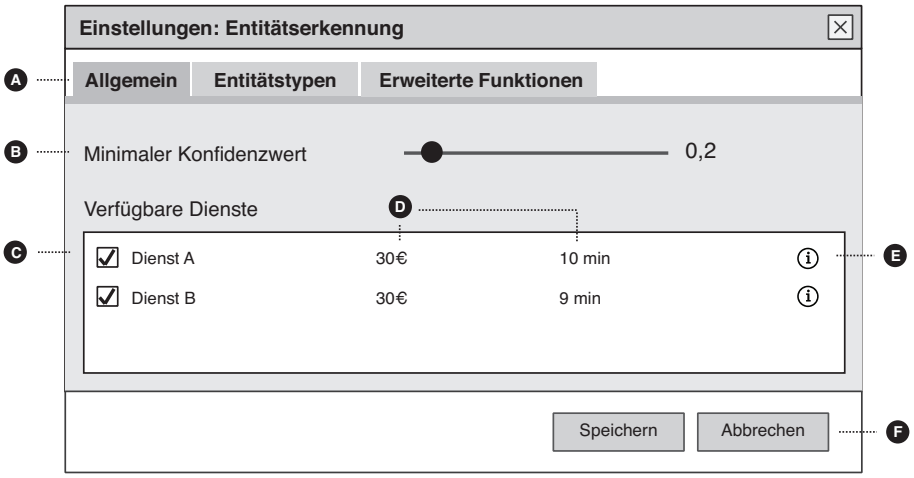


Abbildung 5.15: Dialog für aufgabenspezifische Einstellungen

Um zusätzlich die Zusammensetzung der Gesamtkosten und Zeiten transparenter zu gestalten, werden wie bei den TM-Aufgaben die anfallenden Kosten sowie die zu erwartenden Ausführungsdauer (5.15.D) angezeigt. Des Weiteren wird ein Info-Button (5.15.E) bereitgestellt, über den sich der Nutzer genauere Informationen zu den Diensten einholen kann. Dazu gehören beispielsweise, welche Sprachen der Dienst unterstützt, für welche Domänen dieser geeignet ist sowie ein Link zur Webseite des Dienstes. Die genannten Informationen werden dafür in einem weiteren Popup dargestellt.

Um die Konfiguration zu speichern oder zu verwerfen, werden in der unteren Leiste ein Speichern- und Abbrechen-Button (5.15.F) platziert.

Wie zu Beginn erläutert, richtet sich der Inhalt der weiteren Tabs nach der entsprechenden TM-Aufgabe. Handelt es sich bei der TM-Aufgabe um die NER oder ERD, wird zur Erfüllung der Anforderung FA 2.4 ein Tab zur Typauswahl bereitgestellt. Die Ansicht für eine solche Typauswahl ist Gegenstand des nächsten Abschnitts.

Ansicht zur Auswahl von Entitäts- und Relationstypen

Um die Ergebnisse der NER oder ERD genauer einzuschränken, sollen die gewünschten Entitäts- und Relationstypen aus einer vorgegebenen Typhierarchie ausgewählt werden können. Die oberste Ebene dieser Hierarchie enthält alle Grundtypen wie zum Beispiel „Person“ oder „Organisation“. In den darunter liegenden Ebenen folgen

die Untertypen wie zum Beispiel „Künstler“ oder „Sportler“.

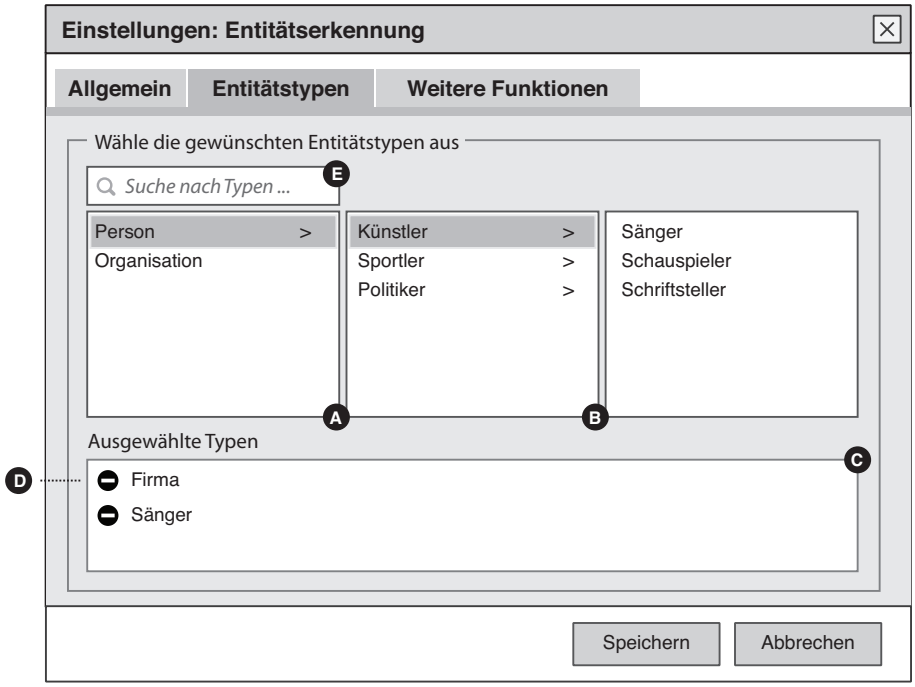


Abbildung 5.16: Typauswahl am Beispiel der Konfiguration der NER

Abbildung 5.16 zeigt die Ansicht für die Auswahl der Entitätstypen. Zur Navigation in der Typhierarchie wird das Entwurfsmuster „Miller-Spalten “ angewandt. Dazu werden in der ersten Spalte (5.16.A) alle Grundtypen aufgelistet. Klickt der Nutzer ein Element dieser Liste an, erscheint, insofern das Element Unterelemente besitzt, eine weitere Spalte (5.16.B), in der diese zu sehen sind. Reicht der Platz für mehr als drei Spalten nicht aus, wird eine Scrollbar unterhalb der Spalten eingeblendet. Mit Hilfe dieser Spalten kann der Nutzer explorativ nach den gewünschten Typen suchen. Ausgewählt werden diese durch einen Doppelklick oder indem sie per Drag’n’Drop in die Auswahlliste (5.16.C) gezogen werden. Eine Anleitung dafür wird in der leeren Liste der ausgewählten Typen angezeigt (5.16.C). Löschen lassen sich die ausgewählten Elemente über den vor dem Namen befindlichen Button (5.16.D).

Um Nutzern, die mit den Typen bereits vertraut sind, eine Abkürzung zu bieten, wird zusätzlich eine Stichwortsuche (5.16.E) zur Verfügung gestellt. Unabhängig von der Ebene der gefundenen Typen werden die Suchergebnisse in der ersten Spalte aufgelistet. Somit kann der Nutzer schneller auf Typen tieferer Ebenen zugreifen. Enthält ein Ergebnis hierbei Untertypen, erfolgt die Navigation wieder nach dem „Miller-Spalten “-Entwurfsmuster. Sobald das Suchfeld leer ist, werden wieder die Typen der obersten Ebene angezeigt. Bringt die Suche keine Ergebnisse, wird dies entsprechend in der Liste vermerkt.

5.3.3 Übersicht der Textanalyseergebnisse

Für die Ansicht der Textanalyseergebnisse wird nach FA 3.1 gefordert, dass alle analysierten Textdokumente in einer Übersicht präsentiert werden. Wie in Abbildung 5.17 dargestellt, wird analog zur Dokumentenübersicht das Entwurfsmuster „Tabellen “ angewandt. Damit der Nutzer einen ersten Eindruck über die Ergebnisse seiner Textanalyse bekommt, sollen in der Tabelle (5.17.A) nicht nur die Namen der Dokumente, sondern auch signifikante Ausschnitte der entsprechenden Ergebnisse aufgelistet werden. Hierbei werden zur besseren Übersicht bewusst nur Ausschnitte gewählt, da mitunter sehr viele TM-Ergebnisse zu einem Dokument entstehen können. Folgende Werte sollen für die Ergebnisse der einzelnen TM-Aufgaben angezeigt werden:

- **Erkennung von Entitäten:** Anzahl der gefundenen Entitäten
- **Erkennung von Relationen:** Anzahl der gefundenen Relationen
- **Textklassifikation:** Hauptkategorie des Textes
- **Stichwortextraktion:** Signifikantes Stichwort
- **Konzept-Zuweisung:** Maximal drei Konzepte
- **Sentimentanalyse:** Polarität

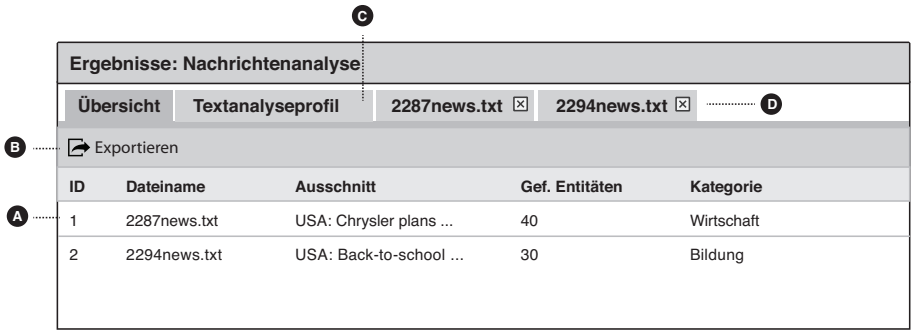


Abbildung 5.17: Entwurf für die Übersicht der Textanalyseergebnisse

Um Platz zu sparen, werden nur die Spalten bzw. TM-Aufgaben dargestellt, welche auch analysiert werden sollten. Die Ansicht gibt somit einen ersten Überblick über die Ergebnisse der Textanalyse. Es werden hierbei jedoch noch keine Details deutlich. Aus Platzgründen werden die genauen Ergebnisse zu einem Dokument in einer eigenen Ansicht dargestellt, welche sich analog zur Dokumentensammlung durch einen Doppelklick auf das entsprechende Dokument öffnen lässt. Die einzelnen Ansichten werden dabei wieder in Tabs (5.17.D) dargestellt. Somit bleibt die Benutzeroberfläche konsistent. Damit die Analyseergebnisse entsprechend der Anforderung FA 3.4 auch exportiert werden können, wird sowohl in der Ergebnisübersicht als auch in den Ansichten der Ergebnisdetails ein Exportieren-Button in der Werkzeugleiste bereitgestellt (5.17.B).

Neben den Tabs für die Detailansichten wird zusätzlich ein Tab für das verwendete Textanalyseprofil (5.17.C) bereitgestellt. Darüber kann der Nutzer das Profil später noch einmal einsehen. Damit der Nutzer zudem sieht, wann die Textanalyse durchgeführt wurde, wird zusätzlich das Datum der Textanalyse in diesem Tab vermerkt. Im Unterschied zur Konfigurationsansicht des Textanalyseprofils, kann der Nutzer in dieser Ansicht keine Einstellungen ändern. Um das Profil jedoch nochmals zu verwenden, soll er eine Kopie davon im Arbeitsbereich erstellen können. Zu beachten dabei ist, dass sich die Dienste mit der Zeit ändern können, wodurch bestimmte Profile mitunter nicht mehr ausführbar sind. Darauf wird der Nutzer durch ein Popup hingewiesen, sobald er das Profil rekonstruieren will.

5.3.4 Dokumentenspezifische Ansicht der Ergebnisse

Die im vorhergehenden Abschnitt beschriebene Tabelle gibt dem Nutzer einen ersten Überblick über die verschiedenen Ergebnisse und ermöglicht einen groben Vergleich einzelner Dokumente. Jedoch verrät die Ansicht keine Details, wie beispielsweise an welcher Textstelle eine bestimmte Entität in einem Dokument gefunden wurde oder welche Konfidenzwerte die Ergebnisse besitzen. Der in diesem Abschnitt beschriebene Entwurf schließt diese Lücke gemäß Anforderung FA 3.2.

Für jedes Dokument wird eine wie in Abbildung 5.18 zu sehende Ansicht bereitgestellt. Um den Zusammenhang zwischen Inhalt und den Ergebnissen nicht zu verlieren, wird die Ansicht in einen Text- (5.18.A) und Ergebnisbereich (5.18.B) eingeteilt. Der rechte Bereich (5.18.B) dient als Übersicht, in welchem der Nutzer alle Analyseergebnisse zum aktuellen Dokument findet. Um die Ergebnisse der unterschiedlichen TM-Aufgaben strukturell voneinander zu trennen, wird für jede TM-Aufgabe eine entsprechende Box bereitgestellt.

Für die TM-Aufgaben SE, TK und KoZ wird jeweils eine flache Liste von Ergebnissen zurückgegeben. Um die Ansichten dafür nicht unnötig kompliziert zu machen, werden die Ergebnisse dieser Aufgaben, wenn vorhanden, zusammen mit den jeweiligen Konfidenzwerten als Liste dargestellt. Als Ergebnis der SA wird die Polarität entsprechend ausgegeben.

Die Ansichten für die Ergebnisse der Entitätenerkennung sowie Relationserkennung sind etwas umfangreicher, da sie folgende Anforderungen erfüllen müssen. Erstens muss der Nutzer sehen können, welche Entitäten und Relationen im Text enthalten sind und zweitens muss deutlich werden, an welchen Stellen im Text die Entitäten oder Relationen entdeckt wurden. Um die erste Aufgabe zu erfüllen, werden die ermittelten Typen zusammen mit den im Text gefundenen Entitäten und Relationen ähnlich wie beim OpenCalais-Viewer (siehe S. 30) als Baum (5.18.C und D) dargestellt. Hinter jeder Entität wird dabei die Anzahl der Instanzen mit angegeben.

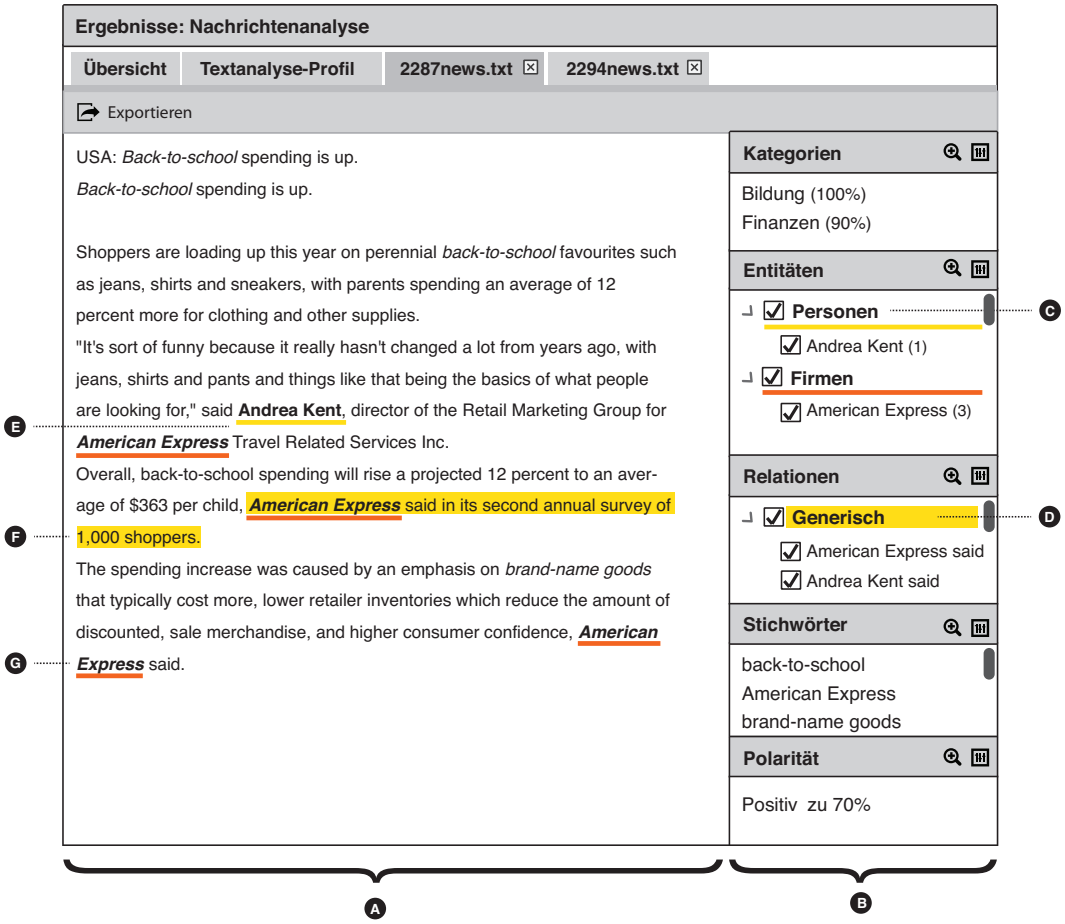


Abbildung 5.18: Ansicht der Dokumentenergebnisse

Zur Erfüllung der zweiten Anforderung werden die Entitäten und Relationen direkt im Text markiert. Damit der Nutzer dabei die Entitäten und Relationen besser auseinander halten kann, werden Entitäten unterstrichen (5.18.E) und Relationen farbig hinterlegt (5.18.F). Wie in Abbildung 5.18 zu sehen ist, hat diese Variante den Vorteil, dass sich Entitäten und Relationen auch überlappen können. Um zusätzlich die visuelle Wahrnehmung von Entitäten zu erhöhen, werden diese fett hervorgehoben.

Des Weiteren sollen im Text auch die Typen der Entitäten und Relationen deutlich werden. Eine Variante wäre, die Typen direkt mit anzuzeigen, jedoch würde der Text dadurch zu unübersichtlich werden. Es wurde daher der Ansatz des OpenCalais-Viewer übernommen, bei dem die Markierungen eine vom Typ abhängige Farbe erhalten. Um die Farben jeweils zuzuordnen, werden diese im rechten Bereich (5.18.C und D) mit angezeigt. Jedem Typ wird dabei eine feste Farbe zugeordnet, sodass die Farbmarkierungen durchweg konsistent sind.

Da sich nicht nur die Entitäten und Relationen, sondern auch die Resultate der Stichwortextraktion auf Textstellen beziehen, werden diese entsprechend hervorgehoben.

Um sie dabei visuell von den bereits genannten Elementen abzuheben, werden die extrahierten Stichwörter rekursiv (5.18.G) gestellt.

Zu jedem Ergebnis, welches in dieser Ansicht zu sehen ist, können weitere Zusatzinformationen wie zum Beispiel genaue Daten zu einer Person oder bestimmte Informationen über eine Kategorie zur Verfügung stehen. Diese soll der Nutzer abrufen können, indem er mit der Maus über ein entsprechendes Ergebnis fährt und wie in Abbildung 5.19 zu sehen, ein Info-Fenster erscheint, in dem die Zusatzinformationen angezeigt werden.

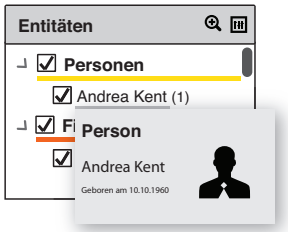


Abbildung 5.19: Info-Fenster für Zusatzinformationen

Neben der Darstellung der Ergebnisse soll auch eine Bewertung und Korrektur dieser möglich sein (siehe FA 4.1 und FA 4.2). Der folgende Abschnitt zeigt den Entwurf der dafür notwendigen Ansichten.

Bewertung der Ergebnisse

Um die Bewertung der Ergebnisse möglichst intuitiv zu gestalten, werden in dem bereits erläuterten Info-Fenster verschiedene Buttons zur Bewertung und Korrektur bereitgestellt. Abbildung 5.20 zeigt das erweiterte Info-Fenster für eine im Text erkannte Entität.

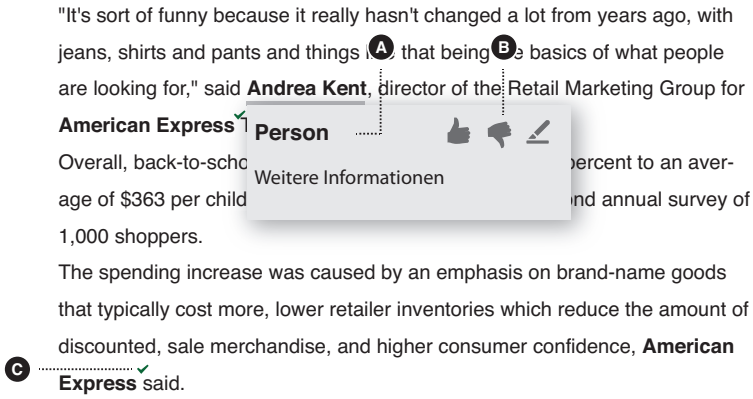


Abbildung 5.20: Ansicht zur Korrektur und Bewertung der Ergebnisse der NER

Entsprechend der in Abschnitt 4.4.1 vorgestellten Varianten zur Bewertung von Er-

gebnissen werden zwei Buttons (5.20.B) bereitgestellt, um das Ergebnis als richtig oder falsch zu bewerten. Zusätzlich wird hierbei noch ein dritter Button zum Aufruf der Korrekturfunktion hinzugefügt. Damit die Bedeutung der Buttons ersichtlicher werden, erhält jeder Button ein entsprechendes Tooltip[2].

Bei der Bewertung ist die Einteilung in Richtig oder Falsch nicht für alle Ergebnistypen ausreichend. Zum Beispiel hängt die Richtigkeit einer Entität nicht nur vom Typ, sondern auch von der Textstelle ab. Aus diesem Grund soll der Nutzer, wenn er solch ein Ergebnis als falsch bewertet, über einen Dialog festlegen, was konkret falsch ist. Tabelle 5.3.1 zeigt die verschiedenen Optionen für die entsprechenden TM-Aufgaben.

NER + ERD	**TK**	**SE + KoZ + SA**
Typ falsch	Zu ungenau	Komplett falsch
Textstelle falsch	Komplett falsch	
Komplett falsch		

Tabelle 5.3.1: Optionen zur Bewertung von falschen Ergebnissen

Reicht dem Nutzer die Bewertung nicht aus, so kann er das Ergebnis auch korrigieren. Der Entwurf für diese Ansicht folgt im nächsten Abschnitt.

Hat der Nutzer ein Ergebnis bewertet oder korrigiert, sollte dies erkennbar sein. Es wird daher bei jedem bewertenden Ergebnis ein grüner Haken (5.20.C) oder ein rotes Kreuz platziert. Im Fall, dass der Nutzer das Ergebnis korrigiert hat, wird ein kleiner Stift angezeigt.

Korrektur der Ergebnisse

Mit Hilfe der Korrekturfunktion sollen in der Dokumentenansicht die bestehenden Ergebnisse korrigiert und fehlende Ergebnisse hinzugefügt werden können (siehe Anforderung FA 4.2). Je nach Ergebnistyp müssen dabei unterschiedliche Korrekturmethoden bereitgestellt werden.

Abbildung 5.21 zeigt die Ansicht zur Korrektur einer bestehenden Entität. Aufgerufen wird die Ansicht über den als Stift dargestellten Button (siehe Abbildung 5.20.B).

Um eine Entität zu berichtigen, muss der Nutzer in der Lage sein, eine konkrete Textstelle zu definieren sowie den entsprechenden Typen dafür festzulegen. Um dies möglichst intuitiv zu gestalten, kann die Textstelle mit Hilfe der dargestellten Schieberegler (5.21.A) direkt im Text markiert werden. Damit die Ansicht auch zur Korrektur von Relationen oder Stichwörtern verwendet werden kann, muss anschließend

[2]Kleines Popup-Fenster mit kurzem Beschreibungstext

der Typ der Markierung ausgewählt werden. Hierfür wird eine entsprechende Combobox mit den Optionen „Entität“, „Relation“ und „Stichwort“ (5.21.B) zur Verfügung gestellt. Abhängig vom ausgewählten Ergebnistypen werden darunter die typspezifischen Einstellungen angezeigt.

Für Entitäten und Relationen wird eine wie in Abschnitt (5.3.2) beschriebene Typauswahl angeboten (5.21.C). Der ausgewählte Typ wird dann am unteren Rand (5.21.D) sowie in der rechten Seitenleiste angezeigt. Erlaubt ist immer nur ein Typ. Für Stichwörter ist keine Typeinstellung erforderlich. Die Typauswahl wird daher entsprechend ausgeblendet.

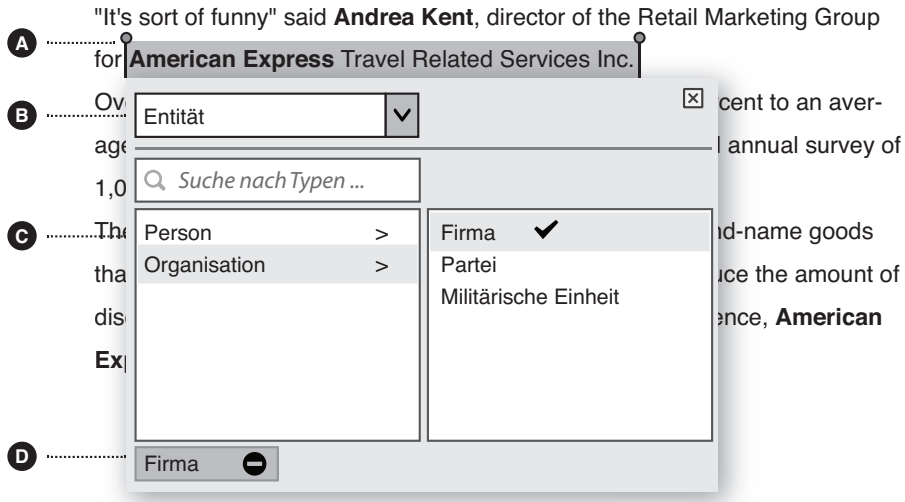


Abbildung 5.21: Ansicht zur Korrektur der Ergebnisse der NER, ERD oder SE

Ähnlich wie die Korrektur einer bestehenden Entität erfolgt auch das Hinzufügen einer nicht erkannten Entität. Hierzu kann der Nutzer die entsprechende Textstelle markieren, worauf ein Fenster mit dem bereits gezeigten Korrektur-Button erscheint. Klickt der Nutzer diesen an, wird wieder das in Abbildung 5.21 zu sehende Popup-Fenster geöffnet. Um ein selbst erstelltes Ergebnis auch wieder zu entfernen, wird in dem Popup-Fenster mit den Buttons zur Bewertung und Korrektur ein zusätzlicher Entfernen-Button bereitgestellt.

Die Korrektur bestehender Kategorien oder Konzepte wird ähnlich wie bei den Entitäten in einem Popup-Fenster durchgeführt. Ein Beispiel für die Korrektur der TK-Ergebnisse ist in Abbildung 5.22 zu sehen. Das Popup-Fenster erscheint, sobald der Nutzer auf den Korrektur-Button (5.22.A) klickt. Um neue, nicht ermittelte Kategorien oder Konzepte hinzuzufügen, wird zusätzlich ein Button im Kopf der Box (5.22.B) bereitgestellt. Über beide Buttons wird ein wie in Abbildung 5.22 dargestelltes Popup-Fenster geöffnet, in dem der Nutzer die Kategorie bzw. das Konzept aus einer vorgegebenen Kategorie- oder Konzepthierarchie auswählen kann. Zur Aus-

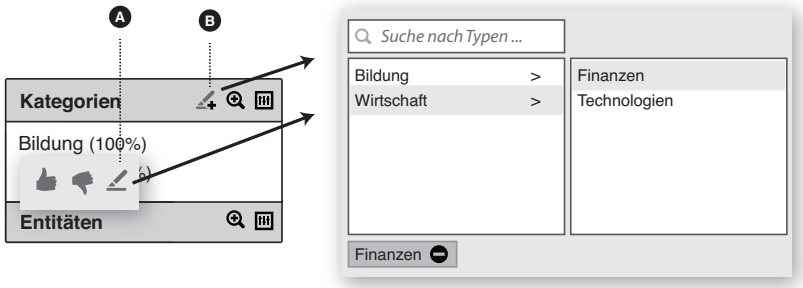


Abbildung 5.22: Ansichten zur Korrektur der TK

wahl der Kategorien kommt dabei wieder das „Miller-Spalten “-Entwurfsmuster zum Einsatz.

Die Korrektur für das Ergebnis der SA erfolgt über einen Schieberegler, der über den entsprechenden Korrektur-Button (5.23.A) aufgerufen wird. Hierbei muss der Nutzer nur die empfundene Polarität des Textes einstellen. Ein Zurücksetzen-Button setzt das Ergebnis auf das aggregierte Ergebnis zurück. Wie bei den anderen Ergebnissen, erscheint auch hier ein kleines Stiftsymbol, sobald das Ergebnis geändert wurde.

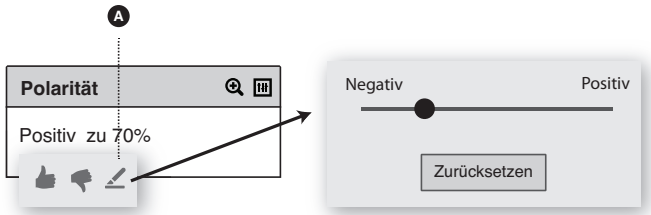


Abbildung 5.23: Ansicht zur Korrektur der SA

Die in diesem Abschnitt beschriebenen Ansichten können auch zur Erstellung von Musterlösungen verwendet werden, wonach Anforderung FA 1.4 erfüllt wird.

5.3.5 Ansichten zum Vergleich der Ergebnisse

Nach Anforderung FA 3.3 sollen erfahrene Nutzer die Möglichkeit bekommen, die unterschiedlichen Ergebnisse der Dienste vergleichen zu können. Des Weiteren ist nach FA 4.3 gefordert, dass erfahrene Nutzer auch Einfluss auf die Aggregation der Ergebnisse nehmen können. Die folgenden Abschnitte führen die Entwürfe der dafür notwendigen Ansichten ein. Aufgerufen werden die Ansichten über die in den Ergebnisboxen verfügbaren Buttons. Zur Erinnerung zeigt Abbildung 5.3.5 noch einmal einen Ausschnitt der Ergebnisansicht mit den entsprechenden Buttons.

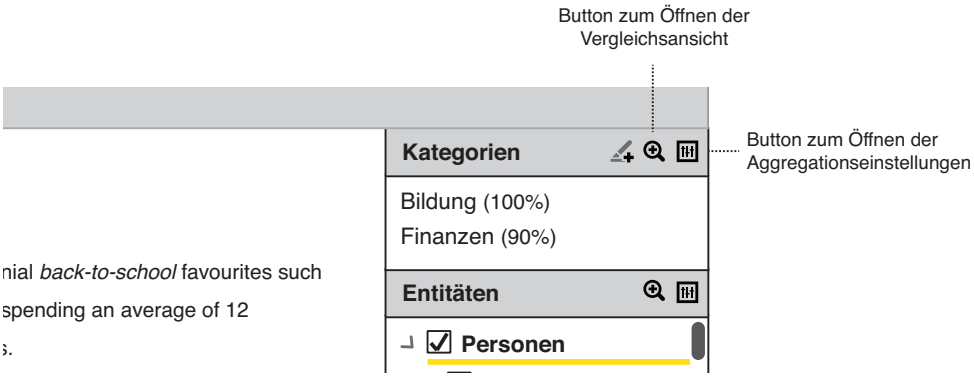


Abbildung 5.24: Ausschnitt der Ansicht der Dokumentenergebnisse

Ansichten zum Vergleich der Ergebnisse der Textklassifikation, Konzeptzuweisung und Sentimentanalyse

Da sich die Ergebnisse der TK, KoZ und SA nicht auf einzelne Textstellen, sondern den gesamten Inhalt beziehen, werde diese, wie in Abbildung 5.25 zu sehen, in einer Tabelle dargestellt. Die Tabelle befindet sich dabei in einem Popup-Fenster. Dies hat den Vorteil, dass gleichzeitig auch die Aggregationseintellungen, auf welche später noch eingegangen wird, geöffnet sein kann.

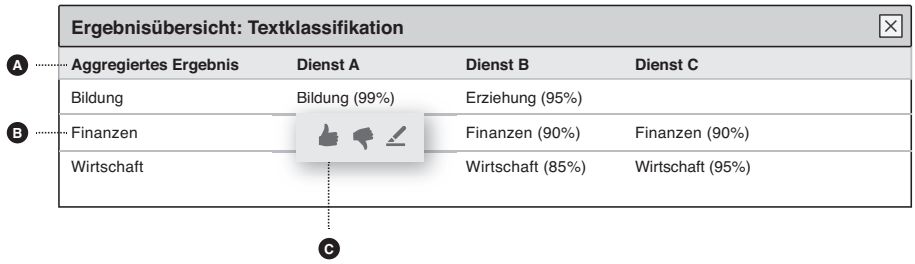


Abbildung 5.25: Expertenansicht zur Bewertung und Korrektur der Ergebnisse der TK

Die erste Spalte der Tabelle beinhaltet die aggregierten Ergebnisse. Sie soll dem Nutzer zeigen, wie die unterschiedlichen Ergebnisse aggregiert werden. In den darauf folgenden Spalten (5.25.A) werden die Ergebnisse der einzelnen TM-Dienste aufgelistet. Die Ergebnisse werden dabei entprechend der Aggregation sortiert. In diesem Fall befinden sich alle semantisch übereinstimmenden Ergebnisse wie beispielsweise die Themen „Bildung“ und „Erziehung“ in der selben Zeile. Somit erkennt der Nutzer schneller die Gemeinsamkeiten und Unterschiede zwischen den Resultaten der einzelnen Dienste.

Des Weiteren soll es in dieser Ansicht möglich sein, die Ergebnisse zu bewerten. Um die Bedienung einheitlich zu halten, wird hierfür wieder ein Info-Fenster mit

entsprechenden Buttons (5.25.C) verwendet. Bewertet der Nutzer ein Ergebnis, das bei verschiedenen Diensten auftaucht, als richtig oder falsch, werden alle Ergebnisse entsprechend markiert.

Vergleichsansicht für die Ergebnisse der Entitäts- und Relationserkennung sowie Stichwortextraktion

Im Gegensatz zu den bereits betrachteten TM-Aufgaben, beziehen sich die Ergebnisse der Entitäts- und Relationserkennung sowie der Stichwortextraktion direkt auf einzelne Textstellen. Ein Vergleich dieser Ergebnisse ist daher nur innerhalb des Textes möglich. Die entworfene Ansicht zum Vergleich der Ergebnisse der NER, ERD und SE soll im Folgenden anhand der Ergebnisse der NER erläutert werden (siehe Abbildung 5.3.5).

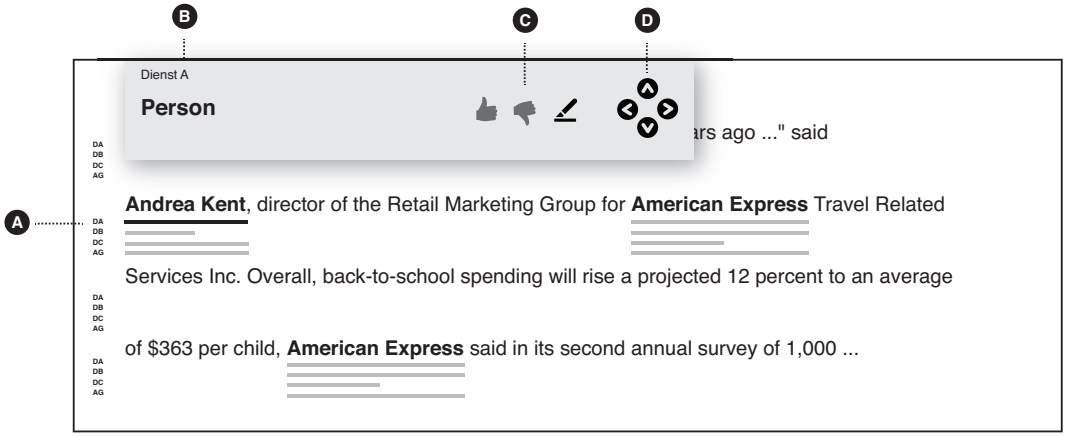


Abbildung 5.26: Expertenansicht zur Bewertung und Korrektur der Ergebnisse der NER

Die Ergebnisse der unterschiedlichen Dienste können sich sowohl hinsichtlich der Textstelle als auch hinsichtlich des Typs unterscheiden. Zum Beispiel kann Dienst A ermittelt haben, dass Textstelle X eine Person ist, während Dienst B sagt, dass es eine Firma ist. Um die Unterschiede bei den Textstellen zu visualisieren, sollen ähnlich wie bei BC-VisCon (siehe Seite 31) Unterstreichungen dienen. Jede Unterstreichung stellt dabei eine erkannte Textstelle eines konkreten Dienstes dar. Damit der Nutzer die einzelnen Dienste zuordnen kann, werden an der linken Seite die entsprechenden Abkürzungen der Dienste (5.3.5.A) aufgeführt. Zum besseren Vergleich ist zusätzlich zu den Dienstenergebnissen auch das jeweils aggregierte Ergebnis (als AG gekennzeichnet) enthalten. Somit sieht der Nutzer sofort, welche unterschiedlichen Textstellen die einzelnen Dienste erkannt haben und wie sich das aggregierte Ergebnis zusammensetzt.

Was dabei aber noch nicht deutlich wird, ist der Entitätstyp. Dafür wird ein extra

Fenster an oberster Stelle des Textes bereitgestellt. Der Inhalt des Fensters bezieht sich immer auf eine konkrete Unterstreichung. Da die Unterstreichungen sehr schmal sind, soll das Fenster nicht erst angezeigt werden, wenn der Nutzer auf eine Linie zeigt, sondern sobald er auf den Button für die Vergleichsansicht klickt (siehe Abbildung 5.3.5). Damit der Nutzer sieht, auf welche Unterstreichung sich der Inhalt des Fensters bezieht, werden alle übrigen Unterstreichungen ausgegraut.

Im Fenster selbst werden der jeweilige Dienst sowie der Entitätstyp der unterstrichenen Textstelle (5.3.5.B) angezeigt. Somit kann der Nutzer die Abkürzungen an der Seite besser zuordnen und sieht den zugeordneten Entitätstypen. Zusätzlich stehen die drei bereits beschriebenen Buttons (5.3.5.C) zur Bewertung und Korrektur bereit. Um zwischen den einzelnen Unterstreichungen zu wechseln, dient das Steuerkreuz (5.3.5.D) auf der rechten Seite. Mit Hilfe der Hoch- und Buttons kann der Dienst ausgewählt werden. Die Links- und Rechts-Buttons dienen zur Auswahl der Textstelle. Mit Hilfe dieser Ansicht lassen neben den Ergebnissen der NER auch die Ergebnisse der ERD und SE vergleichen.

Die selbe Ansicht kann zudem eingesetzt werden, um die verschiedenen Dienstergebnisse mit einer Musterlösung zu vergleichen. Durch Einfärben der Unterstreichungen können dabei die Unterschiede und Übereinstimmungen zur Musterlösung visuell hervorgehoben werden. Eine mögliche Farbgebung für die von den Diensten ermittelten Ergebnissen ist in Tabelle 5.3.2 aufgeführt.

	Übereinstimmung					
Textstelle	ja	teilweise	ja	teilweise	ja	nein
Typ	ja	ja	teilweise	teilweise	nein	-
Farbe						

Tabelle 5.3.2: Mögliche Farbgebung zur Visualisierung der Übereinstimmungen

Ansicht zur Konfiguration der Aggregationsmethode

Nach Anforderung (FA 4.3) soll der Nutzer auch Einfluss auf die Aggregationsmethode nehmen können. Hierfür wird für jede TM-Aufgabe das in Abbildung 5.27 dargestellte Popup-Fenster zur Verfügung gestellt. Geöffnet wird das Fenster über den rechten der in Abbildung 5.3.5 dargestellten Buttons. Die Einstellungsmöglichkeiten für die Aggregation der Ergebnisse wurde von BC-VisCon (siehe Abschnitt 4.4) übernommen.

In der Combobox (5.27.A) kann der Nutzer die gewünschte Aggregationsmethode wählen. Da die Dienste mitunter völlig falsche Ergebnisse liefern, können in der darunter befindlichen Liste (5.27.B) einzelne Dienste auch deaktiviert werden. Des Wei-

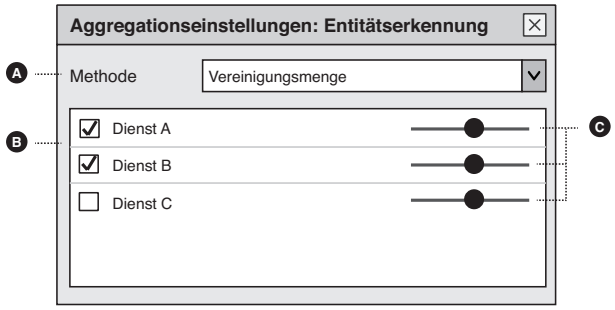


Abbildung 5.27: Ansicht zur Konfiguration der Aggregationsmethode

teren werden für die Aggregationsmethoden, welche eine Gewichtung der Dienste mit einbeziehen, Schieberegler (5.27.C) bereitgestellt. Alle Änderungen wirken sich dabei direkt auf die Ergebnisse in der Dokumentenansicht sowie der vorher beschriebenen Vergleichsansicht aus. Hierdurch sieht der Nutzer sofort, welche Einstellungen die besten Ergebnisse für ihn liefern und wie sich die verschiedenen Aggregationsmethoden auswirken.

5.4 Übersicht der umgesetzten Anforderungen

Nachdem der Entwurf für die in Kapitel 3 erhobenen Anforderungen beschrieben wurde, gibt Tabelle 5.4.1 nochmal einen Überblick, in welchen Abschnitten die einzelnen Anforderungen behandelt wurden. Insgesamt wurden alle Anforderungen betrachtet.

5.1	5.2.1	5.3.1	5.3.2	5.3.3	5.3.4	5.3.5
FA 1.2	FA 1.2	FA 1.1	FA 2.1	FA 3.1	FA 1.4	FA 3.3
	FA 1.3	FA 1.2	-		FA 3.2	FA 4.3
		FA 1.4	FA 2.9		FA 3.4	
		FA 2.9			FA 4.1	
		FA 3.5			FA 4.2	

Tabelle 5.4.1: Übersicht der behandelten Anforderungen

6 Implementierung

Um den Entwurf aus Kapitel 5 möglichst genau zu evaluieren, wurde ein entsprechender Prototyp erstellt. Dieses Kapitel zeigt die wichtigsten Kernpunkte der Implementierung. Bevor näher auf die einzelnen Punkte eingangen wird, zeigt Abschnitt 6.1, welche Ansichten des Entwurfs umgesetzt wurden. Anschließend wird in Abschnitt 6.2 die Architketur des Prototyps erläutert und die generelle Funktionsweise vorgestellt. In den Abschnitten 6.3 und 6.4 folgt eine Betrachtung der Algorithmen zum Zugriff auf die Typhierarchien der NER und ERD sowie zum Zugriff auf die Dienstbeschreibungen. Abschließend wird in Abschnitt 6.5 der Algorithmus zum Rendern der Textmarkierungen eingeführt.

6.1 Umgesetzte Anforderungen

Für den ersten Prototyp wurde der Fokus auf die Grundfunktionalität der Anwendung gelegt. Folgende Ansichten wurden dafür umgesetzt:

- Arbeitsbereich (S. 37)
- Aktivitätsbereich (S. 38)
- Dokumentenübersicht und -ansicht (S. 37)
- Ansicht zur Konfiguration des Textanalyseprofils (S.43)
- Ansicht für aufgabenspezifische Einstellungen (teilweise, S. 46)
- Ansicht zur Auswahl der Entitätstypen (S. 47)
- Ergebnisübersicht (S. 49)
- Ansicht der dokumentenspezifischen Ergebnisse (teilweise, S. 50)

Abbildung 6.1 zeigt einen Screenshot der Benutzerschnittstelle mit einer geöffneten Dokumentensammlung. Als Sprache für die Oberfläche wurde Englisch gewählt. Weitere Screenshots befinden sich im Anhang auf Seite 85. Neben den genannten Ansichten wurden entsprechende Funktionen zum Hochladen und Auslesen von Textdateien, zum persistenten Speichern von Dokumenten und zur Abfrage externer Dienste implementiert. Eine Anbindung an die TM-Dienste wurde nicht umgesetzt. Alle in diesem Prototyp verwendeten Textanalyse-Ergebnisse sind Dummy-Ergebnisse für eine vorgegebene Menge von Textdokumenten. Das Durchführen einer Textanalyse wird also nur simuliert.

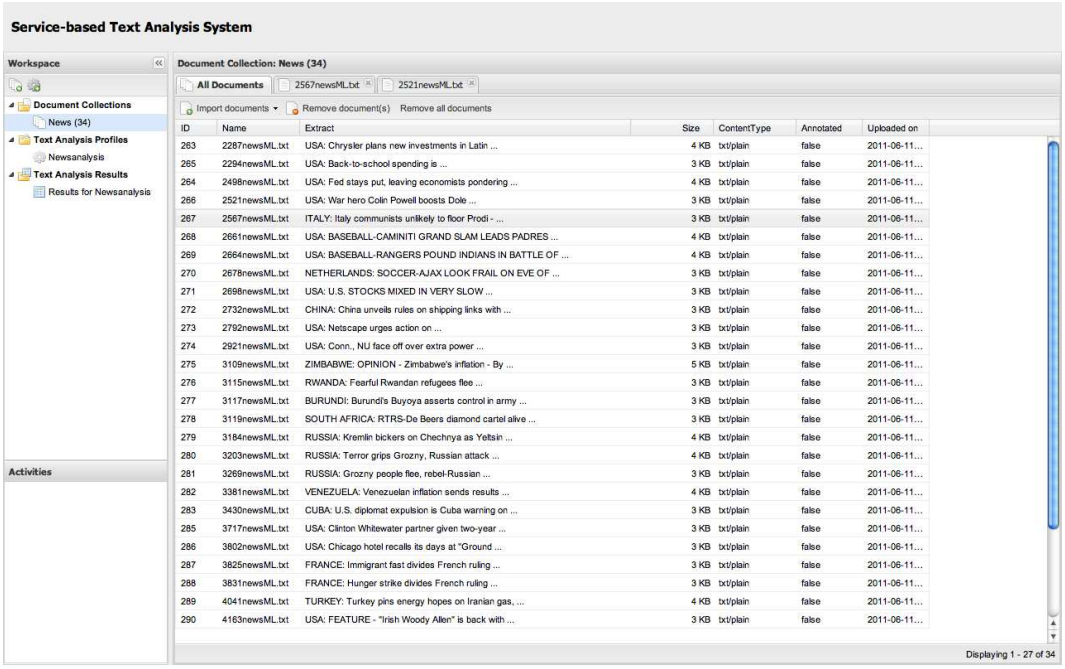


Abbildung 6.1: Benutzeroberfläche des Prototyps

6.2 Architektur und verwendete Technologien

Zur Implementierung wurde das Google Web Toolkit (GWT) [Goo11] eingesetzt. Entsprechend der Empfehlungen von Google wurde der Prototyp nach dem Model-View-Presenter (MVP) Entwurfsmuster umgesetzt. Die Architektur, dargestellt in Abbildung 6.2, gliedert sich in die fünf Schichten *Views*, *Presenter*, *Services* und *Managers*, wobei *Services* und *Managers* das *Model* darstellen.

Die oberste, für den Nutzer sichtbare Schicht bilden die *Views*. Die Aufgabe einer *View* besteht darin, die Ein- und Ausgabeelemente darzustellen. Die Widgets lieferte das ExtGWT Framwork [Sen11]. Im Gegensatz zu den GWT-Widgets sind die ExtGWT-Widgets umfangreicher und bieten mehr Konfigurationsmöglichkeiten. Als Beispiel ist das integrierte Sortieren von Tabellen-Spalten zu nennen.

Für die Steuerung der *View* und Verarbeitung der Nutzereingaben ist jeweils ein *Presenter* zuständig. Dadurch werden Darstellung und Steuerlogik strikt voneinander getrennt. Um die Komplexität der Implementierung zu vermindern, wurde die Benutzeroberfläche in mehrere kleinere *Views* aufgeteilt, wobei jede *View* ihren eigenen *Presenter* besitzt. Die Kommunikation zwischen den einzelnen *Views* bzw. *Presentern* erfolgt über den *Eventbus*. An diesen können die *Presenter* Ereignisse schicken oder sich als Empfänger für Ereignisse registrieren.

Die eigentliche Anwendungslogik befindet sich auf Serverseite und wird durch die

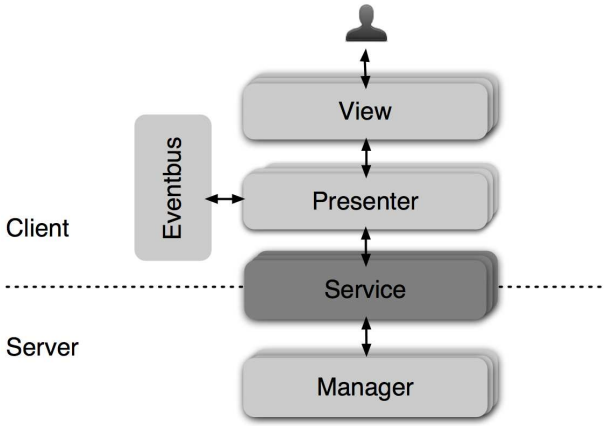


Abbildung 6.2: Architektur des Prototyps

Services bereitgestellt. Die *Services* bilden gleichzeitig das Bindeglied zwischen *Client* und *Server*. Der Datenaustautsch erfolgt dabei durch Datentransferobjekte, die als reine Daten-Container verstanden werden können. Unterstützt werden die Services durch die *Manager*, welche verschiedene Funktionen wie zum Beispiel das persistente Speichern von Daten, das Lesen von Dateien oder den Zugriff auf externe Dienste bereitstellen.

6.3 Abfragen zur Ermittlung der Entitätstypen

Um die Auswahl der Entitätstypen in der Evaluation zu testen, wurde der Entwurf aus Abschnitt 5.3.2 entsprechend umgesetzt. Als Datenquelle für die Entitätstypen wurde DBpedia [DBp11] verwendet. DBpedia ist ein Informationsdienst, welcher Inhalte aus Wikipedia extrahiert und in strukturierter Form zur Verfügung stellt. Dazu zählt auch eine Begriffstaxonomie, nach welcher die in Wikipedia erklärten Begriffe kategorisiert werden. Diese Begriffstaxonomie spiegelt die verschiedenen Entitätstypen sehr gut wieder und wurde daher im Prototyp eingesetzt. Später kann die Taxonomie noch durch eine andere ersetzt werden. Für die Ansicht der Typauswahl mussten die folgenden drei Funktionen bereitgestellt werden:

1. Ermitteln aller Grundtypen
2. Ermitteln aller Untertypen eines ausgewählten Typs
3. Suchen von Typen über eine Zeichenkette

Da DBpedia nur über eine SPARQL-Schnittstelle verfügt, wurde für jede Funktion eine entsprechende Abfrage erstellt. Die erste Abfrage zur Ermittlung aller Grundtypen ist in Listing 6.1 aufgeführt. Damit in der Ansicht die Typen mit Untertypen entsprechend markiert werden können, werden für jeden Typ (?typ) die existierenden Untertypen (?untertyp) mitbestimmt (Zeile 13). Normalerweise würde dafür ein

Ausdruck zum Zählen der Untertypen ausreichen, jedoch wird dieser bei dem von DBpedia verwendeten RDF Server nicht unterstützt. Da die Typen (?typ) nur URIs[1] darstellen, wird zusätzlich die englische Bezeichnung (?name) für jeden Begriff ermittelt (Zeile 11 und 14). Dieser Wert dient gleichzeitig zum Sortieren der Ergebnisliste (Zeile 15).

Listing 6.1: SPARQL-Abfrage zur Ermittlung der Grundtypen

```
1  PREFIX rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#>
2  PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
3  PREFIX owl: <http://www.w3.org/2002/07/owl#>
4  SELECT distinct
5    ?typ       # Typ
6    ?name      # Name des Typs
7    ?untertyp # Untertyp von ?typ
8  WHERE {
9    ?typ rdf:type owl:Class .
10   ?typ rdfs:isDefinedBy <http://dbpedia.org/ontology/> .
11   ?typ rdfs:label ?name .
12   ?typ rdfs:subClassOf owl:Thing .
13   OPTIONAL {?untertyp rdfs:subClassOf ?typ } .
14   FILTER (lang(?name) = "en")
15 } ORDER BY ?name
```

Die zweite Abfrage, dargestellt in Listing 6.2, hat die Aufgabe, alle Untertypen zu einem konkreten Typ (?typ) zu laden. Identifiziert wird ein Typ über einen URI, der bei der Abfrage entsprechend übergeben wird. In dem dargestellten Beispiel ist dies *<http://dbpedia.org/ontology/Person>* (Zeile 10). Analog zur ersten Abfrage wird auch hier wieder die Existenz der Untertypen mit abgefragt, um die entsprechenden Typen markieren zu können (Zeile 7).

Listing 6.2: SPARQL-Anfrage zur Ermittlung der Unterbegriffe

```
1  ...
2  SELECT distinct ?typ ?name ?untertyp
3  WHERE {
4   ?typ rdf:type owl:Class .
5   ?typ rdfs:isDefinedBy <http://dbpedia.org/ontology/> .
6   ?typ rdfs:label ?name .
7   OPTIONAL {?untertyp rdfs:subClassOf ?typ } .
8   FILTER (lang(?name) = "en") .
9   ?typ rdfs:subClassOf ?t .
10  FILTER (sameTerm(?t,<http://dbpedia.org/ontology/Person>))
11 } ORDER BY ?name
```

Die in Listing 6.3 aufgeführte Abfrage dient zur Stichwortsuche nach Entitätstypen. Der einzige Unterschied zu der bereits gezeigten Abfrage besteht darin, dass zusätzlich ein Ausdruck zum Filtern des Namen (?name) enthalten ist (Zeile 5).

[1] URI - Uniform Resource Identifiers

Listing 6.3: SPARQL-Anfrage zur Ermittlung der Unterbegriffe

```
1  ... # Präampel analog zu Listing 6.2
2  SELECT distinct ?typ ?name ?untertyp
3  WHERE {
4    ... # WHERE-Ausdrücke analog zu Listing 6.2
5   FILTER regex(?name, "per") .
6  } ORDER BY ?name
```

6.4 Abfragen zur Ermittlung der Dienstinformationen

Dieser Abschnitt erläutert, wie anhand der Dienstbeschreibungen und des vom Nutzer konfigurierten Textanalyseprofils die passenden Dienste ermittelt und die Kosten sowie die geschätzte Laufzeit zu jeder TM-Aufgabe berechnet werden. Zur Realisierung der genannten Funktionen wurden die bestehenden Dienstbeschreibungen um folgende Angaben erweitert:

1. Kosten pro Transaktion (Null bei kostenlosen Diensten)
2. Geschätzte Dauer pro Transaktion (in Sekunden)
3. Verfügbare Verbindungstypen (sichere, unsicher)
4. Speichern der Daten auf Dienstseite (niemals, optional oder immer)

Die Dienstbeschreibungen liegen im RDF[2]-Format vor und werden durch das Sesame RDF Framework [ope11] verwaltet. Zur Abfrage der Dienstbeschreibungen wird das vom Nutzer konfigurierte Textanalyseprofil in eine SPARQL-Anfrage umgewandelt. Listing 6.4 zeigt ein den ersten Teil dieser Anfrage. In den Zeilen 1-10 werden die Prefixes für die Namespaces der verwendeten Schema festgelegt und die Variablen für die Abfrage definiert. Die Bedeutung der Variablen ist im Listing mit aufgeführt.

Listing 6.4: Erster Teil der SPARQL-Anfrage zur Ermittlung der Dienstinformationen

```
1  PREFIX tm:<http://www.sap.com/textmining/description/ontology#>
2  PREFIX owl:<http://www.w3.org/2002/07/owl#>
3  PREFIX rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#>
4  SELECT
5   ?service          # TM Dienst
6   ?task             # Konkrete TM-Aufgabe
7   ?tasktype         # Typ der TM-Aufgabe
8   ?connectiontype   # Verbindungstyp
9   ?privacytype     # Datenschutztyp
10  ?costs                 # Transaktionskosten
11  ?transactions     # Transaktionen pro Sekunde
12  ...
```

[2]RDF - Resource Description Framework

Listing 6.5: Zweiter Teil der SPARQL-Anfrage zur Ermittlung der Dienstinformationen

```
1  WHERE {
2   ?service tm:supportsExtractionType ?task .
3   ?task rdf:type ?tasktype .
4   FILTER ( ?tasktype = tm:NamedEntityRecognition || ?tasktype = ... ) .
5   ?service tm:isPricedPerTransaction ?costs .
6   FILTER ( ?costs = 0)
7   ?service tm:supportsConnectionType ?connectiontype .
8   FILTER ( ?connectiontype = tm:SecureConnection )
9   ?service tm:supportsPrivacyType ?privacytype .
10  FILTER ( ?privacytype = tm:RetainsContentNever || ?privacytype = tm:
        RetainsContentOptional )
11  ?service tm:maxTransactionsPerSecond ?transactions .
12  FILTER ( ?transactions > 6)
13 }
```

Der zweite Teil der Anfrage ist in Listing 6.5 zu finden. Hier werden anhand der vom Nutzer ausgewählten TM-Aufgaben die entsprechenden Dienste gefiltert (Zeile 2-4). Die Variable ?task entspricht dabei einer konkret vom Dienst bereitgestellten TM-Aufgabe bzw. Funktion, deren Typ (?tasktype) sich auf eine der sechs Aufgabentypen bezieht.

In Zeile 5 erfolgt die Abfrage der Transaktionskosten. Im Fall, dass der Nutzer nur kostenlose Dienste wünscht, wird zusätzlich der darauf folgende Filter-Ausdruck hinzugefügt. Die Sicherheits- und Datenschutzeinstellungen werden wie folgt übersetzt. Wünscht der Nutzer nur sichere Verbindungen, wird der Ausdruck aus Zeile 8 eingefügt. Des Weiteren wird im Fall, dass die Daten nicht gespeichert werden sollen, der in Zeile 10 aufgeführte Filter-Ausdruck hinzugefügt. Mit Hilfe dieses Ausdrucks werden nur Dienste genommen, die entweder niemals die Daten speichern oder deren Datenspeicherung optional ist. Die Abfragen in den Zeile 7 und 9 dienen dabei zur Ermittlung der entsprechenden Verbindungs- und Datenschutztypen.

Um die geschätzte Laufzeit für eine TM-Aufgabe zu berechnen, wird im letzten Teil der Anfrage die maximale Anzahl der Transaktionen pro Sekunde bestimmt (Zeile 11). Werden nur schnelle Dienste gefordert, kann über den letzten Filter-Ausdruck ein höheres Minimum für die Anzahl der Transkationen festgelegt werden (Zeile 12).

Sind die Transaktionskosten sowie die maximale Anzahl an Transaktionen pro Sekunde ermittelt, können auf Basis dieser Daten und der zu analysierenden Textdokumente die genauen Kosten und die genaue Laufzeit der Textanalyse berechnet werden. Da es sich um einen Prototyp handelt, wurde die Berechnung vereinfacht, indem anstatt der Größe der einzelnen Textdokumente nur die Anzahl der Dokumente berücksichtigt wird.

6.5 Algorithmus zum Rendern der Textmarkierungen

Um einzelne Entitäten, Relationen oder Stichwörter direkt im Text zu markieren, wurde eine entsprechender Algorithmus entwickelt. Dieser wird im Folgenden näher erklärt.

Die Eingabe stellt eine Liste von Textmarkierungen dar. Jede Textmarkierung ist durch eine Start- und Endposition sowie einen Typen definiert. Die Aufgabe des Algorithmus ist es, für jede Textmarkierung ein sogenanntes Start- und End-Tag in den Text einzufügen. Dabei muss beachtet werden, dass sich der Text verändert, sobald ein Start- und End-Tag eingefügt wird. Dies hat zur Folge, dass die Positionen der Textmarkierungen nicht mehr mit dem Text übereinstimmen. Des Weiteren muss berücksichtigt werden, dass innerhalb einer Textmarkierung verschiedene weitere Textmarkierungen liegen können. Um diese Probleme zu lösen wird ein rekursiver Ansatz verfolgt.

Im ersten Schritt dieses Ansatzes wird die Liste der Textmarkierungen in einen sortierten Baum eingefügt. Das Wurzelelement dieses Baums bildet eine Textmarkierung, welche den gesamten Text markiert. Sie wird im Folgenden als W bezeichnet. In dieses Wurzelelement werden die einzelnen Textmarkierungen wie folgt eingefügt. Angenommen A sei eine bereits in W eingefügte Textmarkierung und B eine noch einzufügende Textmarkierung, so wird B wie folgt hinzugefügt:

- Wenn($Start_B \geq Start_A \wedge Ende_B \leq Ende_A$), mache B zur Untermarkierung von A.
- Wenn ($Start_B \leq Start_A \wedge Ende_B \geq Ende_A$), mache A zur Untermarkierung von B, entferne A aus W und füge B hinzu.
- Wenn ($Start_B \leq Start_A \wedge Ende_B \leq Start_A$), füge B in der Liste vor A ein.
- Wenn ($Start_B \geq Start_A \wedge Ende_B \geq Start_A$), füge B in der Liste nach A ein.
- Ansonsten verwerfe B.

Somit entsteht ein Baum, der sowohl in der Tiefe als auch in der Breite sortiert ist. Sind alle Textmarkierungen entsprechend in W eingefügt und sortiert, werden über die in Listing 6.6 aufgeführte *render*-Methode die Start- und End-Tags in den Text eingefügt. Das Prinzip der Methode besteht darin, dass jede Textmarkierung ihren Bereich aus dem unmarkierten Text (text) ausschneidet (Zeile 11), an die Enden die entsprechenden Start- und Endtags anhängt und dies als Zeichenkette zurückgibt (Zeile 12).

Enthält eine Textmarkierung mehrere Untermarkierungen (Zeile 6), so wird für jede Untermarkierung jeweils die *render*-Methode mit dem unmarkierten Text aufgerufen (Zeile 8) und deren Ergebnis mit den Stellen vor und nach den Untermarkierungen

zusammengefügt (Zeile 5-11). Dadurch wird der Bereich der Textmarkierung entsprechend „gerendert“ und, wie bereits erwähnt, mit den Start- und End-Tags umschlossen. Mit Hilfe dieser Tags können den Textstellen dann verschiedene Textstile wie Unterstreichungen, Schriftschnitte oder Schriftgrößen zugeordnet werden. Des Weiteren können für jedes Tag verschiedene Interaktionsereignisse definiert werden, um zum Beispiel Zusatzinformationen anzuzeigen, sobald der Nutzer mit der Maus über eine Textstelle fährt.

Listing 6.6: Vereinfachter Code zum Rendern einer Textmarkierung

```
1  public class TextMarkNode{
2    ...
3    public String render(String text){
4      String html = "";
5      int s = this.start;
6      for (TextMarkNode textMarkNode : this.textMarkNodes){
7        html += text.substring(s,textMarkNode.start);
8        html += textMarkNode.render(text);
9        s = textMarkNode.stop;
10     }
11     html += text.substring(s,this.stop);
12     return this.startTag + html + this.endTag;
13   }
14   ...
15 }
```

7 Evaluation

Um den Entwurf der Benutzeroberfläche zu evaluieren wurde ein Usability-Test durchgeführt. Abschnitt 7.1 beschreibt, wie dieser durchgeführt wurde und welche Ziele dabei verfolgt wurden. Im zweiten Abschnitt werden die Ergebnisse der Evaluation vorgestellt. Hierbei werden die gefundenen Schwachpunkte des getesteten Prototyps analysiert und Verbesserungsvorschläge gegeben. Der letzte Abschnitt zeigt, wie die Testpersonen die Benutzeroberfläche bewerten und welche zusätzlichen Erwartungen sie daran stellen.

7.1 Methode und Ziele

Zur Evaluation wurde ein wie in Abschnitt 2.2.3 beschriebener Usability Walk Through mit sechs Testpersonen durchgeführt. Als Probanden wurden Personen gewählt, die schon ein Mal mit Softwareanwendungen gearbeitet haben, jedoch noch keine Erfahrungen mit TM-Systemen besaßen. Den Rahmen für den Usability Walk Through gab ein selbst formuliertes Anwendungsssezenario vor. Die Testpersonen sollten sich vorstellen, dass sie Journalisten sind und eine Menge älterer Nachrichtentexte analysieren wollen. Die grundlegende Aufgabe bestand darin, mit Hilfe des umgesetzten Prototyps herauszufinden, von welchen Themen die einzelnen Nachrichtentexte handeln und welche Firmen sowie Personen darin genannt werden.

Um die Testpersonen schrittweise zu leiten und zu motivieren, wurden entsprechend des Arbeitsablaufs fünf Aufgaben mit mehreren Teilaufgaben definiert (siehe Anhang A.2). Jeder Schritt der Probanden wurde dabei genau protokolliert. Wenn die Probanden an bestimmten Stellen nicht weiter wussten, wurden die Probleme entsprechend notiert.

Als erste Aufgabe mussten die Testpersonen die Startansicht der Benutzeroberfläche beschreiben (siehe Abbildung 7.1). Hierbei wurde untersucht, inwieweit der Arbeitsbereich und die zu Beginn dargestellte Anleitung wahrgenommen werden. In der zweiten Aufgabe wurden die Testpersonen aufgefordert, die Nachrichtentexte in das System zu laden. In der Aufgabenstellung wurde dabei explizit nicht erwähnt, dass zunächst eine Dokumentensammlung dafür erstellt werden muss. Ziel war es, zu überprüfen, inwieweit die Anleitung der Startansicht hilft, das Prinzip von Dokumentensammlungen zu verstehen. Des Weiteren mussten die Testpersonen einzelne

Dokumente in der Dokumentenübersicht öffnen und auch löschen. Hiermit wurde getestet, ob die Bedienung der Übersicht für die Nutzer intuitiv ist.

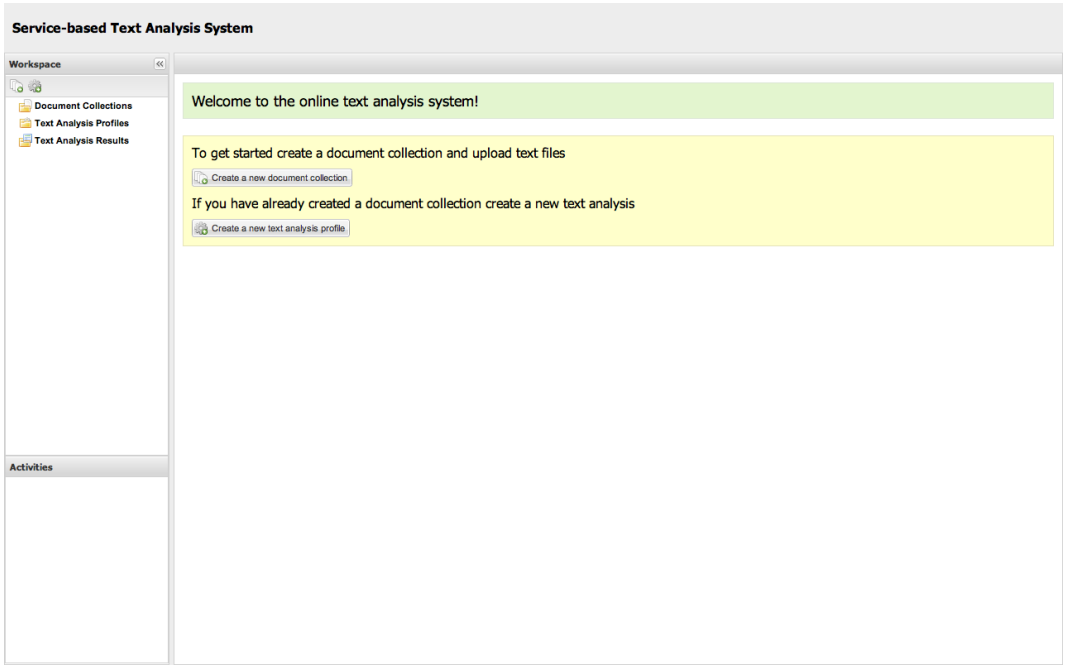


Abbildung 7.1: Startansicht des Prototyps

Die dritte Aufgabe bestand darin, die Textanalyse zu konfigurieren. Analog zur zweiten Aufgabe wurde auch hier nicht erwähnt, dass es notwendig ist, ein Textanalyseprofil zu erstellen. Hiermit wurde überprüft, ob der im Arbeitsbereich vorgesehene Button ausreicht und ob die Probanden die Bedeutung eines Textanalyseprofils verstehen. Nachdem das Profil angelegt war, sollten die zu analysierenden Dokumentensammlungen und durchzuführenden TM-Aufgaben (TK und NER) festgelegt werden. Dies sollte zeigen, ob die Beschreibungen für die TM-Aufgaben ausreichen und die richtigen TM-Aufgaben ausgewählt werden. Des Weiteren wurde beobachtet, ob die Probanden den Zusammenhang zwischen der Konfiguration und den zur Verfügung stehenden Diensten sowie anfallenden Kosten wahrnehmen. Hierfür sollten die Testpersonen festlegen, dass nur kostenlose Dienste verwendet werden. Neben diesen allgemeinen Einstellungen wurde zusätzlich die Typauswahl für die NER evaluiert. Dazu sollten die Testpersonen einstellen, dass nur Personen und Firmen bei der NER gefunden werden sollen.

Die vierte Aufgabe diente zur Evaluation der Ergebnisansicht. Analog zur ersten Aufgabe, sollten die Testpersonen zunächst beschreiben, was sie sehen. Wichtig dabei war, herauszufinden, ob die Testpersonen sowohl die Ergebnisübersicht als auch die dokumentenspezifischen Ergebnisse richtig interpretieren. Um den Ansatz zur Korrektur der Resultate zu überprüfen, wurden die Testpersonen des Weiteren gefragt,

wie sie Ergebnisse der NER korrigieren oder ergänzen würden.

Das Ziel der fünften Aufgabe bestand darin, die Expertenansichten zu evaluieren. Die Evaluation erfolgte mit Hilfe von zwei Mockups[1], da die dafür notwendigen Ansichten im Prototyp nicht umgesetzt wurden. Der Untersuchungsschwerpunkt lag dabei auf der Expertenansicht der NER. Es wurde untersucht, inwieweit die in Abschnitt 5.3.5 beschriebene Ansicht verstanden wird, ohne dass eine Interaktion möglich ist.

Unter Berücksichtigung dieser Untersuchungsschwerpunkte wurde der gesamte Ablauf genau beobachtet und dokumentiert. Die dabei gefundenen Schwachstellen werden im folgenden Abschnitt diskutiert.

7.2 Ergebnisse des Usability-Tests

In diesem Abschnitt werden die Ergebnisse des durchgeführten Usability Walk Throughs ausgewertet und Verbesserungsvorschläge für gefundene Schwachstellen gegeben. Als erstes werden dafür die Startansicht und Dokumentenverwaltung ausgewertet.

7.2.1 Startansicht und Dokumenteneingabe

Die erste Aufgabe des Tests bestand darin, die Startansicht des Prototyps zu beschreiben. Die in der Startansicht dargestellte Anleitung wurde von allen Testpersonen wahrgenommen und insofern verstanden, dass es möglich ist, Dokumentensammlungen sowie Textanalyseprofile anzulegen. Die Hälfte der Probanden konnte sich jedoch noch nichts unter einer Dokumentensammlung oder einem Textanalyseprofil vorstellen. Das Gleiche gilt für den Arbeitsbereich auf der linken Seite. Es ist daher zu überlegen, anstatt einer Anleitung ein kurzes Einführungsvideo anzubieten sowie beim ersten Start eine leere Dokumentensammlung und ein vorkonfiguriertes Textanalyseprofil bereitzustellen.

Auch bei der zweiten Aufgabe zeigte sich, dass einigen Testpersonen die genaue Bedeutung bzw. Funktion einer Dokumentensammlung nicht verstanden. Hierbei wurde gefordert, die verschiedenen Nachrichtentexte in das System zu laden. Die technisch versierten Probanden erstellten dafür wie erwartet eine Dokumentensammlung. Weniger versierte Testpersonen versuchten trotz der beschriebenen Vorgehensweise, die Dokumente direkt zu importieren. Entsprechend der Aufgabenstellung wäre dies zwar logisch, jedoch ist dies im Entwurf bzw. Prototyp nicht vorgesehen. Zu Beginn der Anwendung eine leere Dokumentensammlung zur Verfügung zu stellen, kann eine Lösung für dieses Problem sein. Eine weitere Lösung ist ein Wizard zum Importieren von Dokumenten. Dabei muss ein Schritt des Wizards dazu dienen, entweder eine neue Dokumentensammlung zu benennen oder eine bereits vorhandene auszuwählen.

[1] Grafische Attrappe der Benutzeroberfläche

Das Hochladen der Dokumente innerhalb einer Dokumentensammlung stellte keine Probleme für die Testpersonen dar. Auch das Betrachten verschiedener Dokumente verlief problemlos. Alle Probanden verwendeten intuitiv den Doppelklick und erkannten, dass man über die Tabs zurück zur Übersicht gelangt. Im Anschluss sollten die Testpersonen das größte Dokument löschen (Aufgabe 2.1). Hierfür nutzten alle Testpersonen die Sortierfunktion, um zum größten Dokument zu gelangen. Um das entsprechende Dokument zu entfernen, verwendeten nicht alle Tetpersonen sofort den bereitgestellten Löschen-Button. Eine Testperson versuchte das Dokument durch Drücken der Entfernen-Taste zu löschen. Eine andere durch einen Rechtsklick auf das Dokument in der Erwartung, dass ein Kontextmenü erscheint. In einer überarbeiteten Version sollte dies entsprechend berücksichtigt werden.

7.2.2 Konfiguration der Textanalyse

Bei der dritten Aufgabe ging es darum, eine Textanalyse zu konfigurieren. Die Testpersonen hätten dafür als erstes ein Textanalyseprofil anlegen müssen. Drei Probanden erkannten relativ schnell, dass sie über den im Arbeitsbereich befindlichen Button ein Textanalyseprofil erstellen können. Die anderen Testpersonen hatten jedoch enorme Probleme. Zwei Testpersonen suchten innerhalb der Dokumentenübersicht nach einer Möglichkeit die Textanalyse zu konfigurieren. Eine andere Testperson versuchte wieder zurück zur Anleitung zu gelangen. Beide Varianten waren nicht möglich. Der im Arbeitsbereich vorgesehene Button zum Erstellen eines Textanalyseprofils wurde also nicht von allen Testpersonen ausreichend wahrgenommen. Der Button muss daher mehr betont werden. Um die Benutzeroberfläche dabei konsistent zu halten, muss auch der Button zum Erstellen der Dokumentensammlung deutlicher werden. In Abbildung 7.2 ist eine Variante dargestellt, wie die Buttons mehr Aufmerksamkeit erlangen können. Der Entfernen-Button fällt dabei weg, wobei die Funktion weiterhin im Kontextmenü des Arbeitsbereichs vorhanden ist. Zusätzlich kann, wie in Abbildung 7.2 zu sehen, auch der Button für den Import-Wizard an dieser Stelle platziert werden.

Nachdem das Textanalyseprofil erstellt wurde, sollten die Testpersonen darin die neu angelegte Dokumentensammlung auswählen. Weiterhin sollte eingestellt werden, dass die Themengebiete der Dokumente sowie alle Personen- und Firmennamen ermittelt werden. Die Auswahl der Dokumentensammlung sowie der richtigen TM-Aufgaben (TK und NER) verlief dabei ohne Probleme. Ein Verbesserungsmöglichkeit ist jedoch, jede Aufgabe zusätzlich durch ein spezielles Icon zu kennzeichnen, sodass die Nutzer die verschiedenen Aufgaben schneller wiedererkennen. Des Weiteren sollte in der Liste der ausgewählten Aufgaben ein Info-Button bereitgestellt werden, sodass sich die Nutzer nochmals über die Bedeutung einer TM-Aufgabe informieren können.

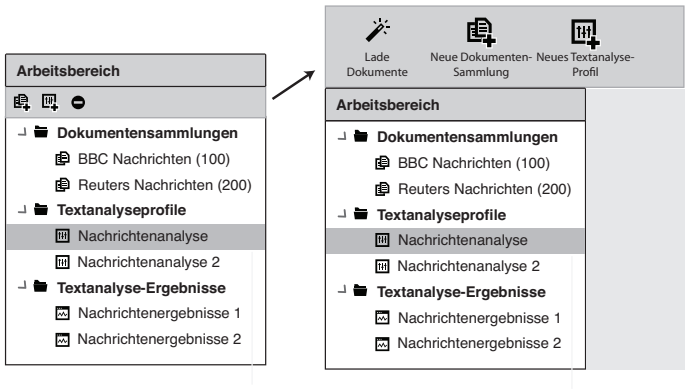


Abbildung 7.2: Verbesserter Arbeitsbereich

Einige Probanden konnten sich die Definition der Aufgaben nicht merken. Abbildung 7.3 zeigt diese Änderungen. Zu sehen ist ein entsprechendes Icon (7.3.A) vor jeder ausgewählten TM-Aufgabe sowie die Info-Buttons (7.3.B). Des Weiteren wurde die Zeile für die anfallenden Kosten und geschätzen Laufzeit mehr betont, da einige Testpersonen diese nicht sofort als Zusammenfassung erkannt haben (7.3.C).

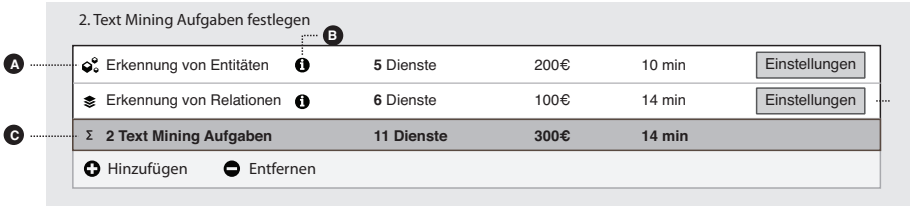


Abbildung 7.3: Verbesserte Ansicht für ausgewählte Textanalyseaufgaben

Als Zusatzaufgabe sollten bei der NER die Entitätstypen „Person“ und „Firma“ eingestellt werden. Zu der entsprechenden Ansicht für die aufgabenspezifischen Einstellungen gelangten alle Testpersonen ohne zusätzliche Hilfe. Die Auswahl der geforderten Entitätstypen erwies sich allerdings als zu schwierig. Der Hinweis, dass die gewünschten Typen durch einen Doppelklick ausgewählt werden können, wurde von den meisten Testpersonen nicht gelesen. Die Funktionsweise erschloss sich daher eher durch Zufall. Somit waren die meisten Probanden noch in der Lage den Entitätstyp „Person“ auszuwählen. Bei dem Versuch den Typ „Firma“ auszuwählen, scheiterten jedoch alle Tester. Dies lag vor allem daran, dass „Firma“ nicht mit in der ersten Spalte aufgeführt wird, sondern als Untertyp von „Organisation“. Die Nutzer verstanden also nicht sofort, dass es sich um eine Typhierarchie handelt. Dies kann deutlicher gemacht werden, indem nicht nur eine Spalte bzw. eine Ebene der Hierarchie zu Beginn angezeigt wird, sondern gleich zwei.

Eine weitere Möglichkeit den Typ „Firma“ zu finden, bot die Stichwortsuche. Diese wurde jedoch von keiner einzigen Testperson wahrgenommen und sollte aus diesem

Grund besser erkennbar gemacht werden. Wie in Abbildung 7.4 zu sehen, wurde daher vor jeden Schritt eine Beschreibung (7.4.A) gesetzt. Des Weiteren wurde die Typauswahl geändert. Diese soll nicht mehr über einen Doppelklick erfolgen, sondern über einen Hinzufügen-Button, der jeweils bei dem aktuell fokusierten Element (7.4.B) erscheint.

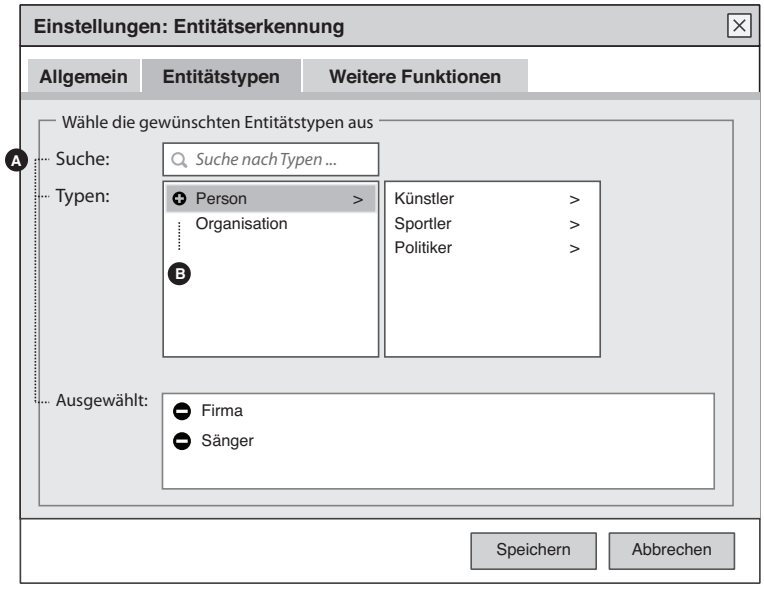


Abbildung 7.4: Verbesserte Ansicht zur Typauswahl

Zusätzlich zu den TM-Aufgaben und Entitätstypen sollten die Testpersonen einstellen, dass nur kostenlose Dienste verwendet werden. Die dafür zuständige Combobox im unteren Bereich der Ansicht des Textanalyseprofils (siehe Abbildung 5.13) wurde von allen Testpersonen gefunden. Klar wurde dabei auch der Zusammenhang zwischen den Kosteneinstellungen und den für die Textanalyse anfallenden Kosten. Bis auf die Typauswahl verlief die Konfiguration der Textanalyse relativ probemlos. Auch das Starten der Textanalyse bereitete den Testpersonen keine Probleme.

7.2.3 Ausgabe der Analyseergebnisse

Die vierte Aufgabe des Tests bestand darin, die Ergebnisse der Textanalyse auszuwerten. Die Überischt der Analyseergebnisse wurde nach Ausführung der Textanalyse automatisch vom System geöffnet. In der Tabelle haben alle Testpersonenen die eingegebenen Dokumente wiedererkannt. Dass in den weiteren Spalten eine erster Ausschnitt der gefundenen Analyseergebnisse zu sehen ist, haben die Probanden zwar wahrgenommen, jedoch hatten sie Schwierigkeiten, die Spalten den jeweiligen TM-Aufgaben zuzuordnen. Dies ist vor allem auf die Namensgebung zurückzuführen. In Abbildung 7.5 sind die Spaltenköpfe der zwei Analyseergebnisse markiert.

Anstatt der Bezeichnung „Kategorie“ ist die Beschreibung „Hauptthema“ passender. Eine alternative Bezeichnung für die Anzahl der gefundenen Entitäten ist schwierig. Ein Tooltip mit zusätzliche Informationen zu den einzelnen Ergebnissen könnte für mehr Klarheit sorgen. Abbildung 7.6 zeigt ein Beispiel. Hierbei wird für das erste Dokument die Anzahl der gefundenen Entitäten genau aufgeschlüsselt, sobald der Nutzer mit der Maus über die Gesamtanzahl fährt.

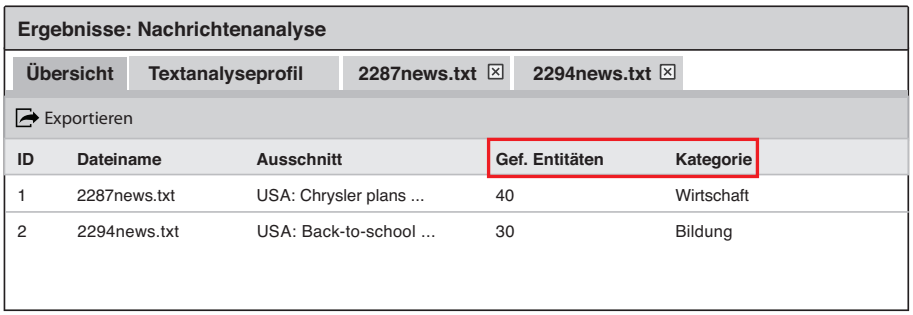


Abbildung 7.5: Spaltenköpfe der Ergebnisüberischt im Prototyp

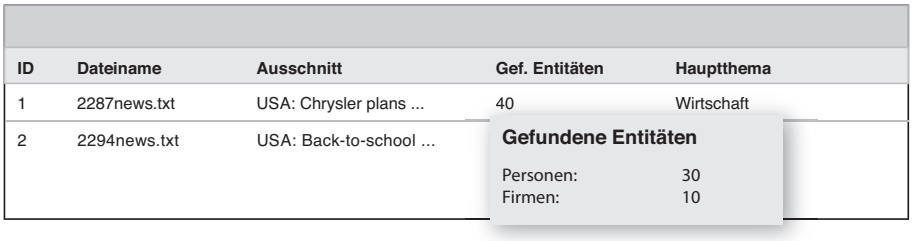


Abbildung 7.6: Verbesserte Ergebnisansicht

Des Weiteren sollten sich die Probanden die genauen Ergebnisse zu einem Dokument anzeigen lassen. Hierfür klickten alle Testpersonen intuiutiv das entsprechende Dokument doppelt an. Dadurch öffnete sich die Ergebnisansicht. Diese galt es in der nächsten Teilaufgabe zu beschreiben. Die Übersicht der genauen Ergebnisse auf der rechten Seite wurde von allen Probanden wahrgenommen. Auch die Unterstreichungen mit den typbezogenen Farben wurden verstanden. Das Einzige, was bei den Testpersonen zu Fragen führte, war der in Klammern dargestellte Prozentwert bei den Kategorien. Dieser Wert spiegelt eigentlich die Konfidenz eines Ergebnisses wieder. Einige Testpersonen interpretierten ihn jedoch als den Anteil, zu welchem der Text einer bestimmten Kategorie angehört. Der Wert muss daher entweder durch ein zusätzliches Tooltip unterstützt oder woanders angezeigt werden.

In der nächsten Teilaufgabe mussten die Testpersonen beschreiben, wie sie vorgehen würden, um sich weitere Zusatzinformationen zu den einzelnen Ergebnissen anzeigen zu lassen. Die meisten Personen gaben an, das entsprechende Ergebnis anzuklicken.

Das im Entwurf beschriebene Prinzip, wonach ein Popup-Fenster angezeigt wird, sobald der Nutzer mit der Maus über ein Ergebnis fährt, kann daher beibehalten werden. Die zwei letzten Teilaufgaben bestanden darin, zu beschreiben, wie man vorgehen würde, um eine Entität zu korrigieren oder zu ergänzen. Alle Probanden wollten hierfür die entsprechende Textstelle markieren. Dies entsprach den Überlegungen aus dem Entwurf und muss daher nicht geändert werden.

Mit dieser Aufgabe endete die Evaluation des Prototyps. Die letzte Aufgabe wurde mit Hilfe zweier Mockups durchgeführt.

7.2.4 Expertenansicht der NER

Bei der letzten Aufgabe lag der Fokus auf der Ansicht zum Vergleich der unterschiedlichen Ergebnisse der NER. Hierfür mussten die Probanden in einem ersten Mockup, zu sehen in Abbildung 7.7, beschreiben, wie sie die Expertenansicht der NER aufrufen würden. Die Hälfte der Testpersonen zeigte auf einen der im Kopfbereich befindlichen Buttons (7.7.A). Die anderen Probanden suchten in der Menüleiste (7.7.B) oder im Text nach einer Option. Es sollte daher zusätzlich ein entsprechendes Menü in der Menüleiste angeboten werden.

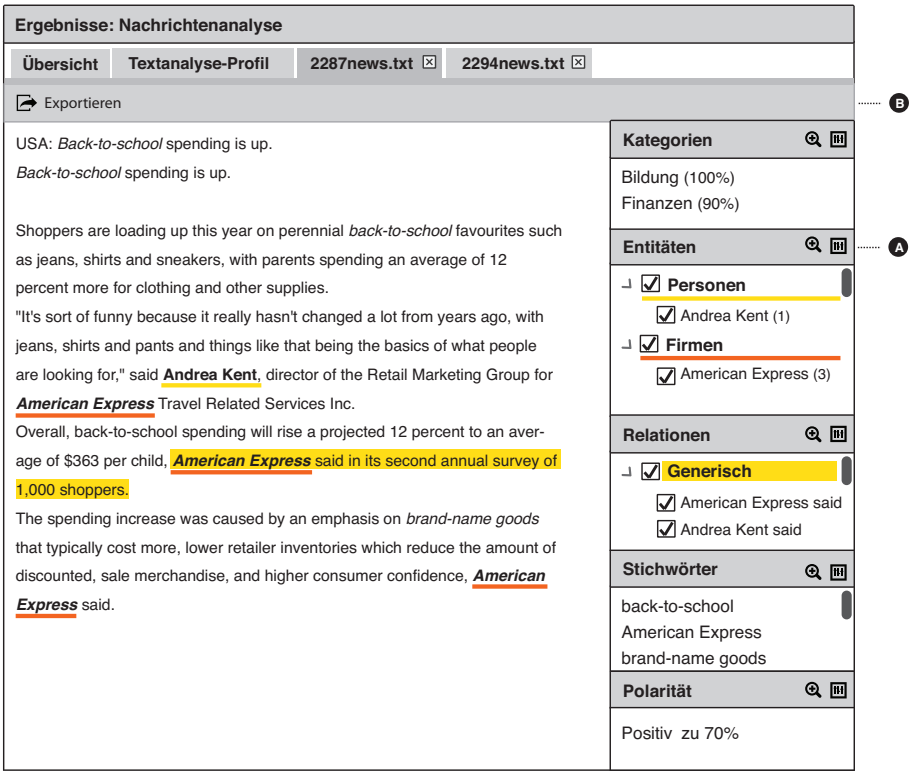


Abbildung 7.7: Erstes Mockup zur Evaluation der Expertenansicht

Als zweite Teilaufgabe mussten die Testpersonen das Mockup der Expertenansicht

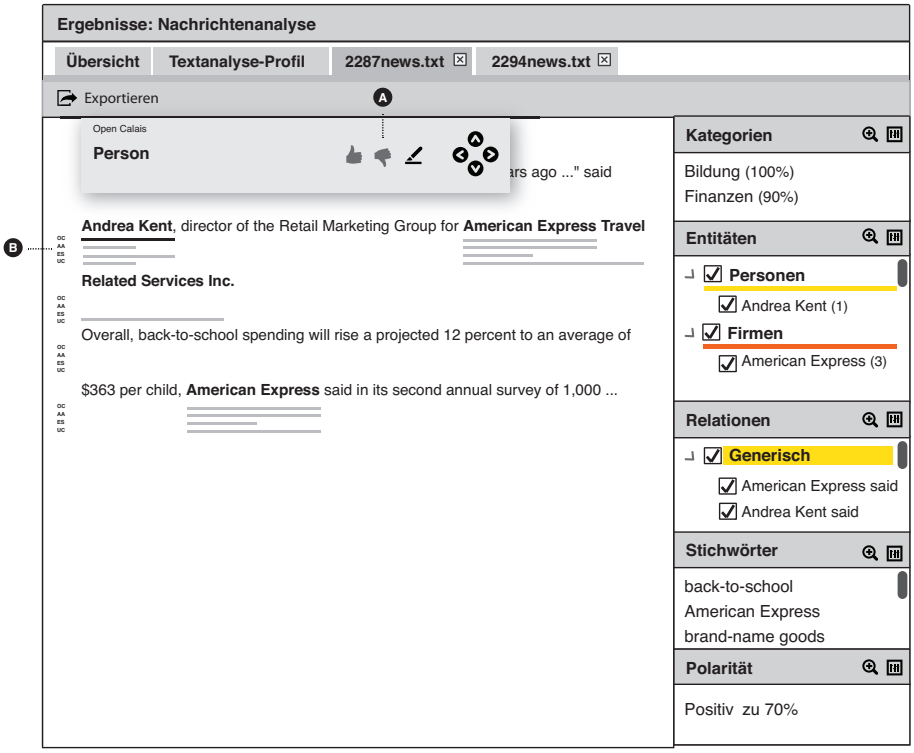


Abbildung 7.8: Zweites Mockup zur Evaluation der Expertenansicht

für die Ergebnisse der NER beschreiben. Zu sehen ist dies in Abbildung 7.8. Die Buttons zur Bewertung (7.8.A) wurde von allen Testpersonen richtig interpretiert. Bei dem Button zur Korrektur der Ergebnisse waren sich einige Probanden nicht ganz sicher. Das Icon für diesen Button muss daher noch mal überdacht werden. Der Zusammenhang zwischen der selektierten Unterstreichung und dem Inhalt des Popupfensters wurde von einem Großteil der Testpersonen erkannt. Auch die verschiedenen Dienste auf der linken Seite des Textes (7.8.B) wurden von den meisten Probanden wahrgenommen, jedoch kam nur eine Testperson darauf, dass sich die einzelnen Unterstreichungen auf die genaue Textstelle beziehen. Fünf von sechs Testpersonen interpretierten die Länge einer Unterstreichung als die Qualität bzw. den Konfidenzwert des entsprechenden Ergebnisses. Diese falsche Interpretation lässt sich größtenteils auf die fehlende Interaktion zurückzuführen. Spätestens beim Wechseln zwischen den einzelnen Unterstreichungen hätten die Testpersonen wahrscheinlich die Bedeutung erfasst. Um den Zusammenhang besser zu verdeutlichen, sollte im Popup-Fenster nicht nur der Typ, sondern auch die markierte Textstelle mit angezeigt werden. Das Problem der fehlenden Interaktion zeigte sich auch beim Steuerkreuz. Trotzdem interpretierten die meisten Testpersonen die Funktion nach genauem Überlegen richtig. Eine Testperson schlug vor, zusätzlich eine Tastatursteuerung anzubieten. Damit war der Walk Through abgeschlossen.

7.3 Allgemeine Bewertung und Verbesserungsvorschläge

Am Ende des Walk Throughs wurden die Testpersonen gebeten, den in Anhang A.3 befindlichen Fragebogen auszufüllen. Hierbei hatten sie die Möglichkeit, einzelne Abschnitte der Interaktion zu bewerten und Stellen zu nennen, die sie besonders verwirrt haben. Des Weiteren konnten sie Verbesserungsvorschläge geben. Bewertet wurden die Eingabe der Dokumente, die Konfiguration der Textanalyse und die Auswahl der Entitätstypen. Auf einer Skala von eins (sehr kompliziert) bis sechs (sehr intuitiv) mussten die Testpersonen angeben, wie intuitiv sie die Interaktion fanden. Die gemittelten Ergebnisse dieser Bewertung sind in Abbildung 7.9 zu sehen.

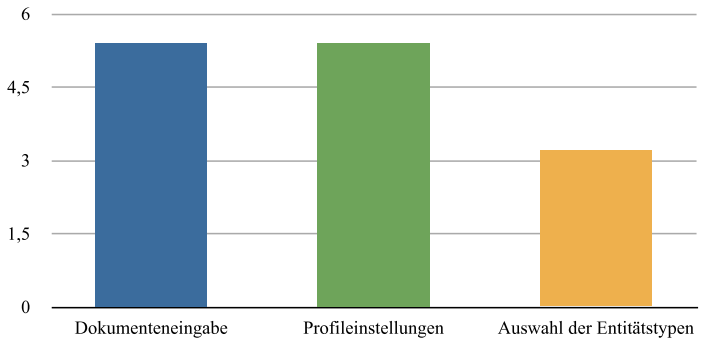


Abbildung 7.9: Gemittelte Bewertungen der Benutzerfreundlichkeit

Die Eingabe der Dokumente sowie die Konfiguration der Textanalyse erhielten eine durchschnittliche Bewertung von 5,4. Dass sie nicht die volle Wertung erreichten, ist auf die bereits genannten Schwachpunkte zurückzuführen. Die Auswahl der Entitätstypen wurde durchschnittlich mit 3,2 bewertet, wodurch noch mal deutlich wird, dass alle Probanden enorme Schwierigkeiten damit hatten. An dieser Stelle ist also unbedingt Verbesserungsbedarf erforderlich.

Auf die Frage, was die Testpersonen von der Anwendung noch erwartet hätten, wurden die folgenden Dinge genannt:

- Mehr visuelle Unterstützung durch Tooltips und größere Buttons
- Mehr (graphische) Möglichkeiten zur Auswertung der Analyseergebnisse
- Eine Funktion, um Aktionen rückgängig zu machen
- Tastaturunterstützung
- Internationalisierung (unterschiedliche Sprachen)

Zusammenfassend ist zu sagen, dass die Evaluation sehr viele Erkenntnisse für die Verbesserung der Benutzeroberfläche gebracht hat. Diese gilt es in weiteren Arbeiten umzusetzen und durch einen erneuten Usability-Test zu überprüfen.

8 Zusammenfassung und Ausblick

Heutzutage liegen immer mehr unternehmensrelevante Informationen in unstrukturierten Daten wie E-Mails, Word-Dokumenten oder Blogeinträgen vor. Damit steigt in vielen Bereichen der Bedarf an Lösungen zur effizienten Auswertung dieser Informationen. Text Mining Systeme stellen eine Möglichkeit dar, dies zu erreichen. Mit Hilfe dieser Systeme lassen sich automatisiert die in Texten enthaltenen Informationen ermitteln und in strukturierter Form aufbereiten. Damit können relevante Informationen schneller entdeckt und leichter mit bereits vorhandenen Daten verknüpft werden. Ausgehend von dem in [SS11] beschriebenen Ansatz für ein serviceorientiertes Text Mining System wurde unter Berücksichtigung von verschiedenen Gestaltungsprinzipien und allgemeinen Entwurfsmustern ein Konzept für eine einfach und intuitiv zu bedienende Benutzeroberfläche entwickelt. Neben den Grundfunktionen zur Eingabe von Dokumenten, zur Konfiguration der Textanalyse und zur Ausgabe der Ergebnisse wurden auch Möglichkeiten zur Bewertung und Korrektur der Analyseergebnisse geschaffen. Besonderer Wert wurde dabei auf die Einstellungen und Ergebnisse der Entitäts- sowie Relationserkennung gelegt. So lassen sich zum einen die zu suchenden Entitäts- oder Relationstypen festlegen und zum anderen die unterschiedlichen, dienstspezifischen Ergebnisse miteinander vergleichen.

Um die Funktionsweise der Benutzeroberfläche entsprechend zu testen, wurden die Grundfunktionen mit Hilfe eines Prototyps evaluiert. Obwohl einige Schwachstellen dabei deutlich wurden, zeigte sich, dass die Benutzerschnittstelle sehr intuitiv ist und selbst unerfahrene Nutzer damit eine Textanalyse durchführen können. Besonders hervorzuheben sind dabei die Dokumenteneingabe und -verwaltung sowie die Konfiguration der Textanalyse. In zukünftigen Arbeiten sollten die gefundenen Schwachstellen wie die Auswahl der Entitätstypen oder die zu kleinen Buttons im Arbeitsbereich ausgebessert und die von den Testpersonen vorgeschlagenen Verbesserungen umgesetzt werden. Zusätzlich sollten Funktionen für spezielle Anwendungsgebiete wie dem Business Intelligence oder der semantischen Suche integriert werden. Somit könnte in Zukunft eine umfangreiche und flexible Text Mining Lösung für eine Vielzahl von Anwendungsbereichen entstehen.

Literaturverzeichnis

[App99] Appelt, Douglas E.: Introduction to Information Extraction. In: *AI Commun.* 12 (1999), Nr. 3, S. 161–172

[App11] Apple: *Apple Human Interface Guidelines: Introduction to Apple Human Interface Guidelines*. `http://developer.apple.com/library/mac/#documentation/UserExperience/Conceptual/AppleHIGuidelines/XHIGIntro/XHIGIntro.html`. Version: Juni 2011

[ASI78] Alexander, Christopher ; Silverstein, Murray ; Ishikawa, Sara: *A Pattern Language: Towns, Buildings, Construction*. Oxford University Press, 1978. – ISBN 0195019199

[BFB10] Barczynski, Wojciech M. ; Foester, Felix ; Brauer, Falk: AdaptIE - Using Domain Language concept to enable Domain Experts in Modeling of Information Extraction Plans. In: *ICEIS 2010 - Proceedings of the 12th International Conference on Enterprise Information Systems* Bd. 1. Funchal, Madeira, Portugal : SciTePress, Juni 2010. – ISBN 978–989–8425–04–1, S. 249–256

[Cal11a] Calais: *Calais Viewer*. `http://viewer.opencalais.com`. Version: April 2011

[Cal11b] Calais: *OpenCalais Documentation / OpenCalais*. `http://www.opencalais.com/documentation/opencalais-documentation`. Version: Januar 2011

[CKMV06] Culotta, A ; Kristjansson, T ; Mccallum, A ; Viola, P: Corrective feedback and persistent learning for information extraction. In: *Artificial Intelligence* 170 (2006), Oktober, Nr. 14-15, 1101–1122. `http://dx.doi.org/10.1016/j.artint.2006.08.001`. – DOI 10.1016/j.artint.2006.08.001. – ISSN 00043702

[CLRR10] Chiticariu, Laura ; Li, Yunyao ; Raghavan, Sriram ; Reiss, Frederick R.: Enterprise information extraction. In: *Proceedings of the 2010 international conference on Management of data - SIGMOD '10*. Indianapolis, Indiana, USA : ACM, 2010, S. 1257–1258

[CMBT02] Cunningham, Hammish ; Maynard, Diana ; Bontcheva, Kalina ; Tablan, Valentin: GATE - A Framework and Graphical Development

Environment for Robust NLP Tools and Applications. In: *Proc. of the 40th Anniversary Meeting of the Association for Computational Linguistics (ACL'02)*. Philadelphia, Juli 2002, S. 168–175

[DBp11] DBpedia: *wiki.dbpedia.org : Home*. http://dbpedia.org. Version: Juni 2011

[DGS99] Dörre, J. ; Gerstl, P. ; Seiffert, R.: Text mining: finding nuggets in mountains of textual data. In: *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, 1999. – ISBN 1581131437, S. 398–401

[EC07] Ercan, Gonenc ; Cicekli, Ilyas: Using lexical chains for keyword extraction. In: *Information Processing & Management* 43 (2007), November, Nr. 6, 1705–1714. http://dx.doi.org/10.1016/j.ipm.2007.01.015. – DOI 10.1016/j.ipm.2007.01.015. – ISSN 0306–4573

[Goo11] Google: *Google Web Toolkit - Google Code*. http://code.google.com/intl/de-DE/webtoolkit/. Version: Juni 2011

[Got11] Gotter, Lukas: *Evaluation von Text Mining Software Tools - Marktübersicht und formelle Evaluation*. http://wissensexploration.de/textmining-software-markt-uebersicht-evaluation.php. Version: 2011

[Hea99] Hearst, Marti A.: Untangling text data mining. In: *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*. Stroudsburg, PA, USA : Association for Computational Linguistics, 1999 (ACL '99). – ISBN 1–55860–609–3, 3–10. – ACM ID: 1034679

[HNP05] Hotho, Andreas ; Nürnberger, Andreas ; Paass, Gerhard: A brief survey of text mining. In: *LDV FORUM - GLDV JOURNAL FOR COMPUTATIONAL LINGUISTICS AND LANGUAGE TECHNOLOGY* (2005). http://130.203.133.150/viewdoc/summary?doi=10.1.1.153.6679

[KB00] Kosala, R. ; Blockeel, H.: Web mining research: A survey. In: *ACM SIGKDD Explorations Newsletter* 2 (2000), Nr. 1, S. 1–15. – ISSN 1931–0145

[KNI11] KNIME: *KNIME*. http://www.knime.org/knime. Version: Juni 2011

[LOK00] Losiewicz, P. ; Oard, D. W. ; Kostoff, R. N.: Textual data mining to support science and technology management. In: *Journal of Intelligent Information Systems* 15 (2000), Nr. 2, S. 99–119. – ISSN 0925–9902

[Nei09] Neil, Theresa: *12 Standard Screen Patterns*. http://designingwebinterfaces.com/

designing-web-interfaces-12-screen-patterns. Version: Januar 2009

[NS07] Nadeau, David ; Sekine, Satoshi: A survey of named entity recognition and classification. In: *Lingvisticae Investigationes* 30 (2007), Januar, Nr. 1, 3–26. http://www.ingentaconnect.com/content/jbp/li/2007/00000030/00000001/art00002. – ISSN 0378–4169

[ope11] openRDF: *openRDF.org: Home*. http://www.openrdf.org/. Version: Juni 2011

[Orc11] Orchestr8: *AlchemyAPI - Documentation*. http://www.alchemyapi.com/api/. Version: Januar 2011

[Pia11] Piatetsky-Shapiro, Gregory: *Software: Text Analysis, Text Mining and Information Retrieval*. http://www.kdnuggets.com/software/text.html. Version: 2011

[RB97] Rajman, Martin ; Besancon, Romaric: Text Mining: Natural Language techniques and Text Mining applications. In: *IN PROCEEDINGS OF THE 7 TH IFIP WORKING CONFERENCE ON DATABASE SEMANTICS (DS-7). CHAPAM* (1997), 7–10. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.17.4827

[RF10] Richter, Michael ; Flückiger, Markus D.: *Usability Engineering kompakt: Benutzbare Software gezielt entwickeln*. 2. Aufl. Spektrum Akademischer Verlag, 2010. – ISBN 9783827423283

[Rus07] Russom, P.: BI Search and Text Analytics. In: *TDWI Best Practices Report* (2007), S. 9–11

[SAP11] SAP: *SAP R/3 Style Guide*. http://www.sapdesignguild.org/resources/minisg/index.htm. Version: Juni 2011

[Sar07] Sarawagi, Sunita: Information Extraction. In: *Foundations and Trends in Databases* 1 (2007), Nr. 3, 261–377. http://dx.doi.org/10.1561/1900000003. – DOI 10.1561/1900000003. – ISSN 1931–7883

[Seb02] Sebastiani, Fabrizio: Machine learning in automated text categorization. In: *ACM Computing Surveys (CSUR)* 34 (2002), März, S. 1–47. http://dx.doi.org/10.1145/505282.505283. – DOI 10.1145/505282.505283. – ISSN 0360–0300. – ACM ID: 505283

[Sek10] Sekine, Satoshi: *Sekine's Extended Named Entity Hierarchy*. http://nlp.cs.nyu.edu/ene/. Version: Oktober 2010

[Sen11] Sencha: *Java UI Component Library for Google Web Toolkit | Ext GWT | Products | Sencha*. http://www.sencha.com/products/extgwt/. Version: Juni 2011

[SLVL09] Starlinger, Johannes ; Leitner, Florian ; Valencia, Alfonso ; Leser, Ulf: SOA-Based Integration of Text Mining Services. In: *SERVICES '09: Proceedings of the 2009 Congress on Services - I*. Los Angeles, CA, USA : IEEE Computer Society, Juli 2009, 99–106

[SN09] Scott, Bill ; Neil, Theresa: *Designing Web Interfaces: Principles and Patterns for Rich Interactions*. 1. O'Reilly Media, 2009. – ISBN 0596516258

[SP04] Shneiderman, Ben ; Plaisant, Catherine: *Designing the User Interface: Strategies for Effective Human-Computer Interaction*. 4. Addison Wesley, 2004. – ISBN 0321197860

[SS11] Seidler, K. ; Schil, A.: Service-oriented information extraction. In: *Proceedings of the 2011 Joint EDBT/ICDT Ph. D. Workshop*, 2011, S. 25–31

[Tan99] Tan, A. H.: Text mining: The state of the art and the challenges. In: *Proceedings of the PAKDD 1999 Workshop on Knowledge Disocovery from Advanced Databases*, 1999, S. 65–70

[Tec11] Technologies, Endeca: *Endeca User Interface Design Pattern Library*. `http://patterns.endeca.com/content/library/en/home.html`. Version: April 2011

[Tid11] Tidwell, Jenifer: *Patterns for Effective Interaction Design*. `http://designinginterfaces.com/firstedition`. Version: April 2011

[Tox11] Toxboe, Anders: *User Interface Design Patterns*. `http://ui-patterns.com/`. Version: April 2011

[VW01] Van Welie, M.: Task-based user interface design. In: *SIKS Disserta* (2001)

[VW11] Van Welie, Martijn: *Interaction Design Pattern Library*. `http://www.welie.com/patterns/`. Version: April 2011

[Wan11] Wandora: *Wandora - WandoraWiki*. `http://www.wandora.org`. Version: Januar 2011

[WGL04] Wickens, C. D. ; Gordon, S. E. ; Liu, Y.: *An introduction to human factors engineering*. Pearson Prentice Hall Upper Saddle River, NJ, 2004. – ISBN 0131837362

A Anhang

A.1 Screenshots des Prototyps

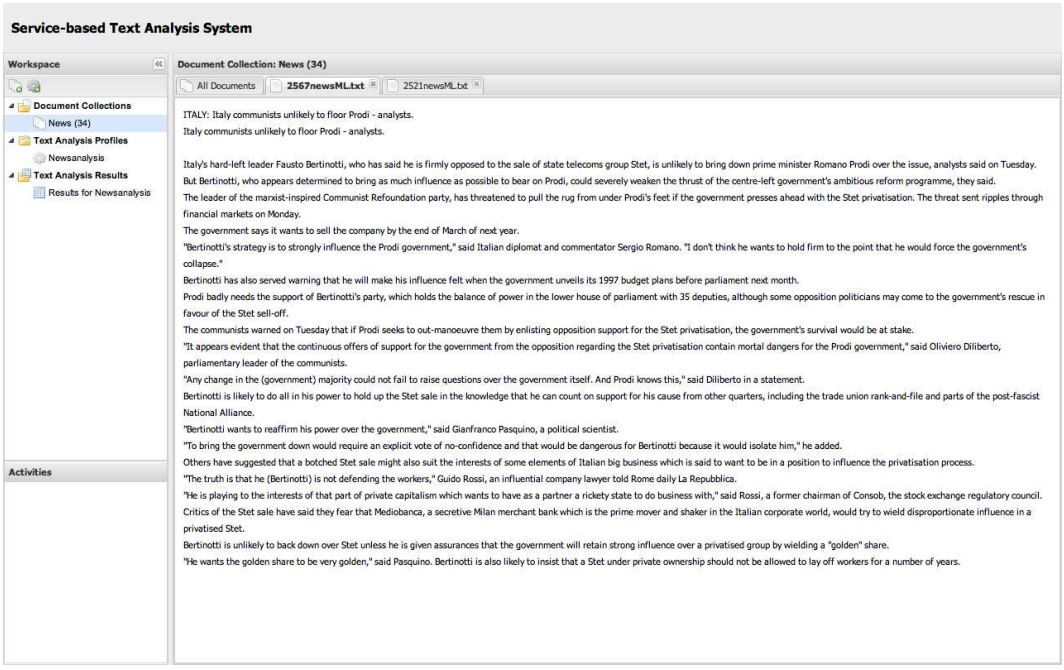


Abbildung A.1: Dokumentenansicht

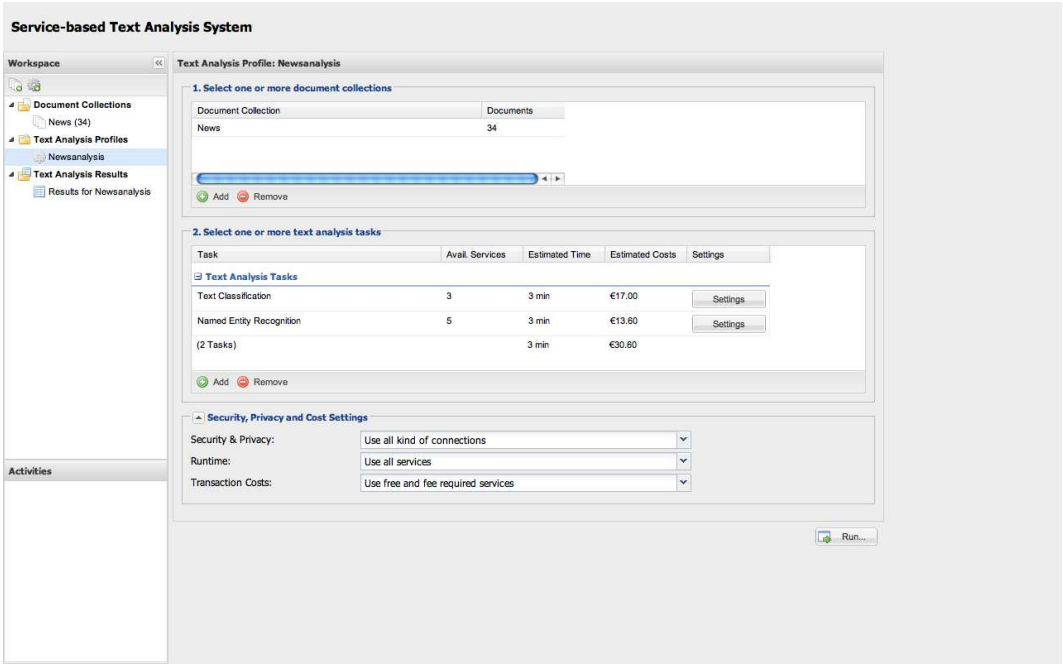


Abbildung A.2: Konfiguration des Textanalyseprofils

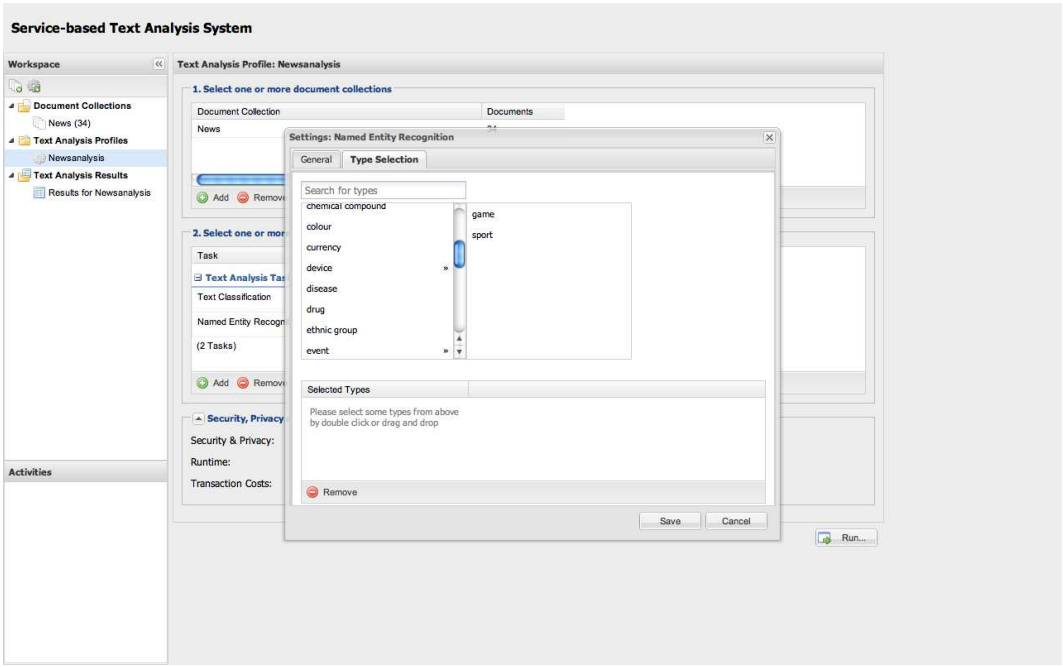


Abbildung A.3: Typauswahl für Entitätserkennung

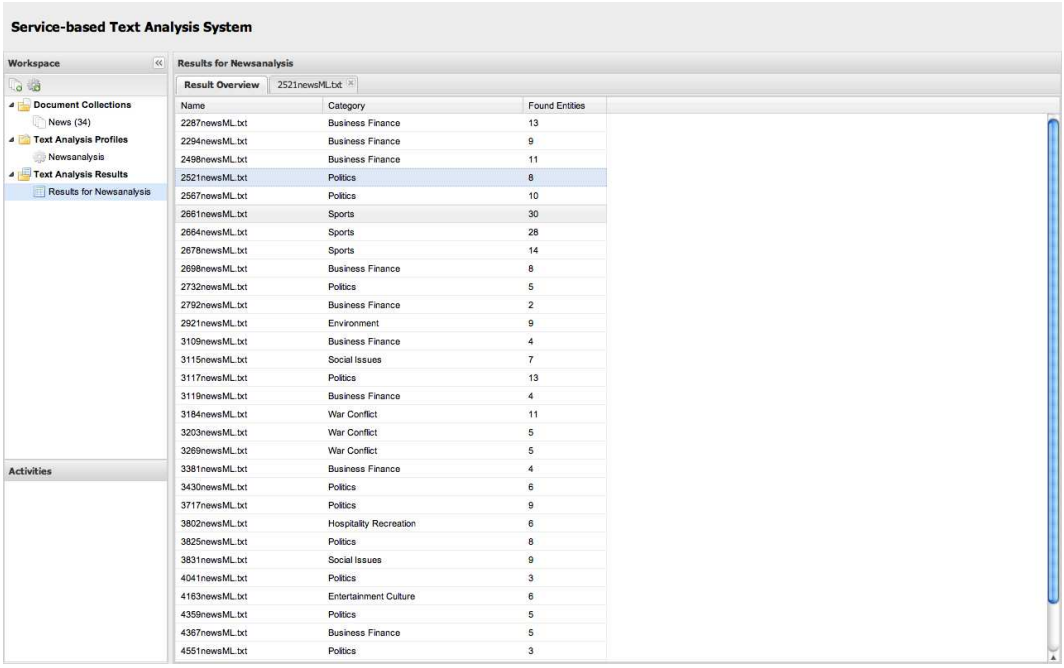


Abbildung A.4: Ergebnisübersicht

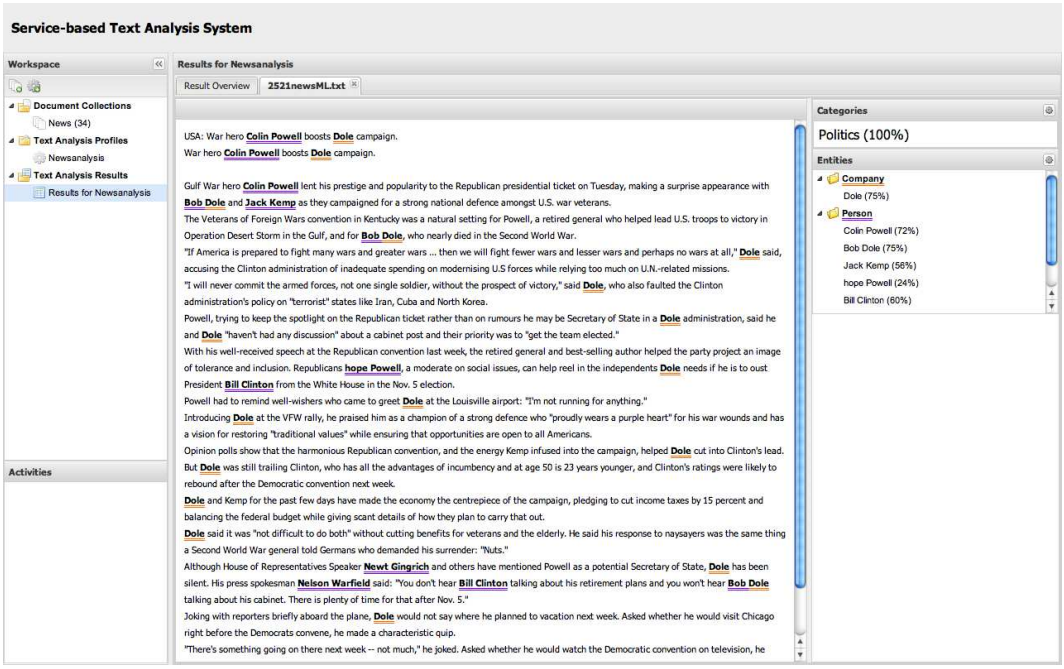


Abbildung A.5: Ergebnisansicht für ein Dokument

A.2 Evaluationsaufgaben

Szenario

Sie sind Journalist und haben eine größere Menge älterer Nachrichtentexte erhalten. Sie möchten herausfinden, von welchen Themen die Texte handeln und welche Firmen und Personen darin genannt werden. Ihre Aufgabe besteht darin, die Nachrichtentexte mit Hilfe des folgenden Textanalyse-Programms auf die genannten Fakten zu untersuchen sowie die Ergebnisse dieser Analyse anschließend zu kontrollieren. Das Programm verwendet zur Textanalyse externe Dienste.

Aufgabe 1

Im Browser vor sich sehen Sie die Benutzeroberfläche der Textanalyse-Anwendung, ohne dass bisher irgendwelche Daten geladen wurden. Nutzen Sie bitte noch nicht die Maus, sondern beschreiben Sie nur:

1. Was sehen Sie?
2. Was glauben Sie, können Sie hier tun?

Aufgabe 2

Sie möchten die besagten Nachrichtentexte in das Programm laden, nochmals anschauen und wenn möglich auch einzelne Texte löschen.

1. Laden Sie alle Dateien aus dem Ordner “Nachrichtentexte” in das Programm.
2. Schauen Sie sich im Programm den Inhalt einzelner Nachrichtentexte an, um sich zu vergewissern, dass die Dateien korrekt hochgeladen wurden.
3. Löschen Sie den längsten Nachrichtentext.

Aufgabe 3

Die Nachrichtentexte wurden geladen und sie möchten jetzt festlegen, was analysiert werden soll.

1. Beschreiben Sie, wie Sie vorgehen würden!
2. Legen Sie fest, dass die so eben hochgeladenen Nachrichtentexte analysiert werden sollen.
3. Sie sind daran interessiert, den Themenbereich jedes Textes zu ermitteln. Versuchen Sie diesen Wunsch einzustellen.
4. Außerdem interessieren Sie die in den Texten auftauchenden Personen- und Firmennamen. Versuchen Sie diesen Wunsch für die Textanalyse einzustellen.
5. Vergewissern Sie sich, dass wirklich nur Personen- und Firmennamen gefunden werden.

6. Nachdem Sie nun ihre Textanalyse-Aufgabe definiert haben, können sie die vorgenommen Einstellungen noch einmal überprüfen. Finden Sie zudem heraus, wie viel die Textanalyse kosten wird und wieviele Dienste für die Ausführung der einzelnen Aufgaben zur Verfügung stehen.
7. Da Sie das System erst einmal nur ausprobieren wollen, möchten Sie nur kostenlose Dienste verwenden. Stellen Sie dies ein!
8. Nachdem Sie die Einstellung vorgenommen haben. Was stellen Sie fest?
9. Damit all ihre gewünschten Analyseaufgaben ausgeführt werden können, stellen Sie ein, dass sowohl kostenfreie als auch kostenpflichtige Dienste verwendet werden. Starten Sie anschließend die Textanalyse und bestätigen die anfallenden Kosten.

Aufgabe 4

Sie haben die Textanalyse durchgeführt und möchten nun die Ergebnisse auswerten.

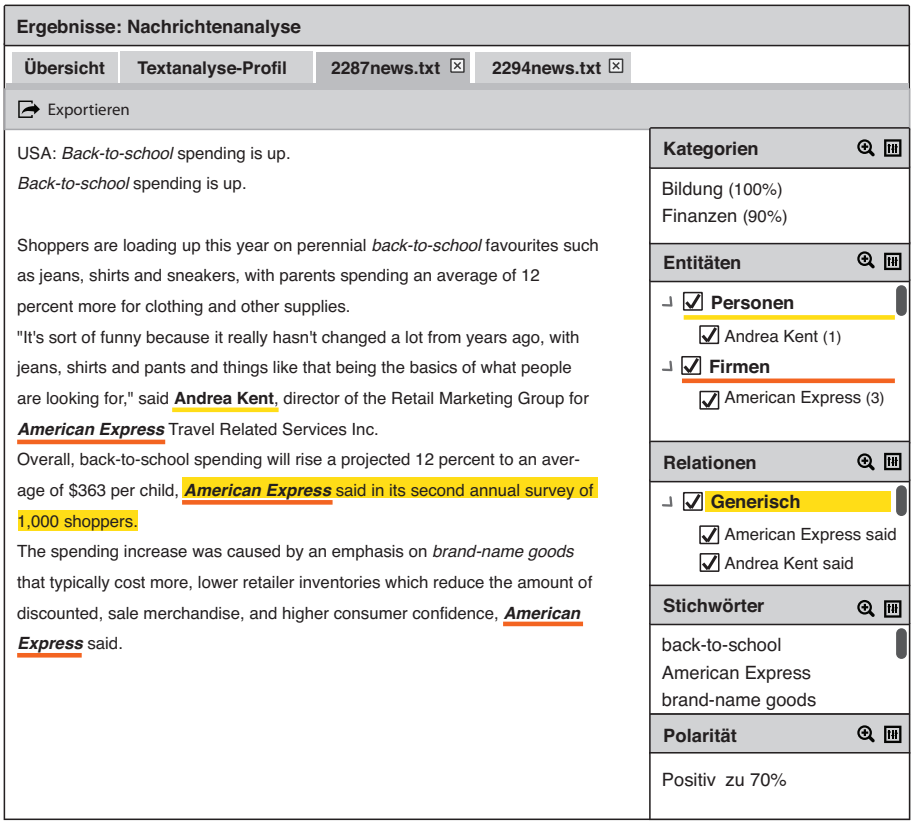
1. Lassen Sie sich die Resultate der Textanalyse anzeigen. Beschreiben Sie, was Sie sehen!
2. Sie wollen die genauen Ergebnisse zu einem Dokument wissen. Versuchen Sie diese aufzurufen und beschreiben sie, was Sie sehen!
3. Zu jedem Ergebnis der Textanalyse stehen verschiedene Zusatzinformationen bereit. Wie würden Sie vorgehen, um diese Informationen herauszufinden?
4. Ihnen fällt auf, dass eine Person im Text nicht erkannt wurde. Vorausgesetzt das System ist lernfähig, wie würden Sie dies dem System beibringen?
5. Sie stellen Fest, dass eine im Text erkannte Person gar keine Person ist. Wie würden Sie vorgehen, um dies dem System mitzuteilen?

Aufgabe 5

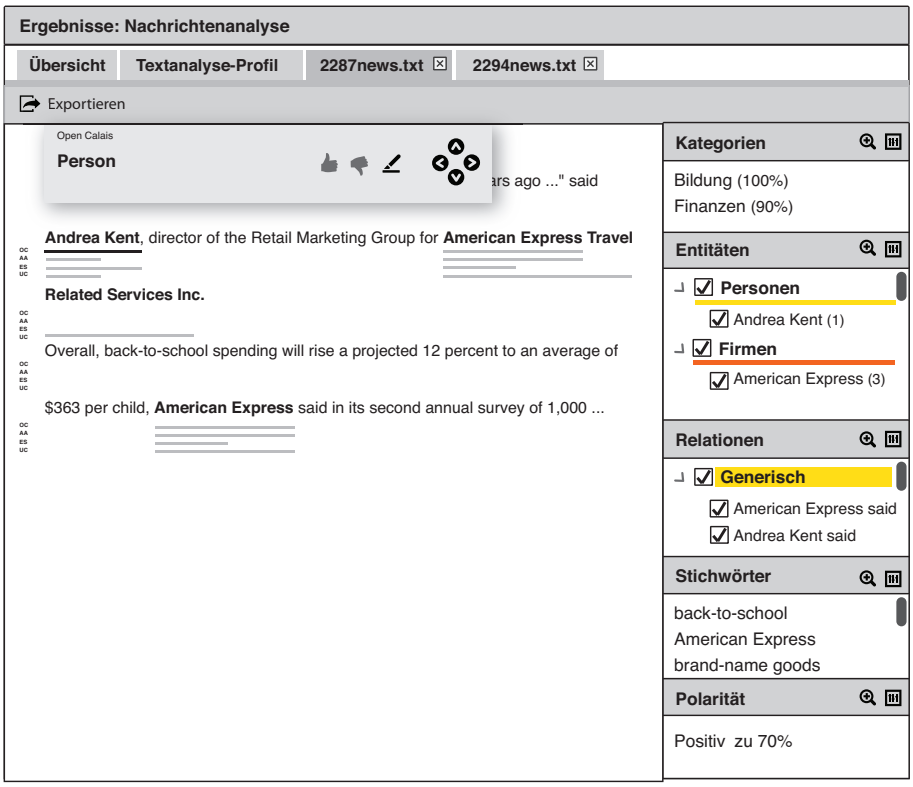
Für die Erkennung der Personen und Firmen wurden verschiedene Textanalyse-Dienste eingesetzt. Zur Zeit sehen Sie eine Zusammenfassung dieser Ergebnisse.

1. Sie möchten die Ergebnisse der verschiedenen Dienste vergleichen. Beschreiben Sie anhand des ersten Mockups, wie Sie vorgehen würden!
2. Auf dem zweiten Mockup ist eine Variante aufgeführt, wie die verschiedenen Ergebnisse der Entitätserkennung verglichen werden können. Beschreiben Sie, was Sie sehen!

Mockup 1



Mockup 2



A.3 Fragebogen

1. Wie fanden Sie die Eingabe der Textdokumente?

1	2	3	4	5	6
Sehr kompliziert					Sehr intuitiv

2. Wie fanden Sie die Konfiguration der Textanalyse?

1	2	3	4	5	6
Sehr kompliziert					Sehr intuitiv

3. Wie fanden Sie die Auswahl der Entitätstypen?

1	2	3	4	5	6
Sehr kompliziert					Sehr intuitiv

4. Welche Stellen haben Sie besonders verwirrt?

5. Was hätten Sie noch von der Benutzeroberfläche erwartet?