Unterschrift des Betreuers

MASTERARBEIT

Automatische Zusammenfassung von Textclustern

Ausgeführt am Institut für

Softwaretechnik und Interaktive Systeme

der Technischen Universität Wien

unter der Anleitung von Ao.Univ.Prof. Dipl.-Ing.Dr.techn. Andreas Rauber

durch

Julius Penaranda

Gottschalkgasse 1/3/12, 1110 Wien

Datum

Unterschrift (Student)

Danksagung

Ich möchte mich bei all jenen bedanken, die mir direkt oder indirekt bei der Entstehung dieser Diplomarbeit beigestanden haben.

Besonderer Dank gebührt meinem Betreuer, Herrn Prof. Dipl.-Ing. Dr.techn. Andreas Rauber, der mich von Anfang an motiviert und diese Arbeit durch wertvolle Hinweise und Ideen vorangetrieben hat.

Ich bedanke mich ebenfalls bei Rudolf Mayer für seine Bereitschaft meine Fragen zu beantworten. Ein grosses Dankeschön auch an Robert Neumayer, der mir sein Parallel-Corpus für meine Experimente zur Verfügung gestellt hat.

Diese Arbeit widme ich meinen Eltern, die mir dieses Studium durch ihre finanzielle und persönliche Unterstützung ermöglicht haben.

Das Internet bietet eine ständig wachsende Menge an Informationen zu fast jedem Thema. Allerdings kann das Suchen nach bestimmten Informationen mit großem Zeitaufwand verbunden sein. Um das Finden und Wiederfinden von richtigen Informationen zu erleichtern, tauchten erste Suchmaschinen auf. Allerdings ist das Ergebnis einer Suchanfrage, in dem alle für den Nutzer relevanten Seiten aufgelistet werden, in der heutigen Zeit nicht mehr geeignet. Extraktionsalgorithmen für automatische Zusammenfassungen können hierbei helfen, einen bzw. mehrere Texte zu einem Thema auf den wesentlichen Inhalt zu verkürzen, so dass Leser schnell und hochinformativ mit den Kernpunkten der Texte versorgt werden.

Der Ausgangspunkt dieser Arbeit ist die automatische Zusammenfassung von Text Clustern. Es sollen verschiedene Ansätze, wie auch neue Methoden der automatischen Erstellung von Zusammenfassungen vorgestellt und analysiert werden. Weiters sollen die automatisch erstellten Zusammenfassungen in einer Evaluation mit manuell verfassten Zusammenfassungen verglichen werden.

The internet offers a constantly growing amount of information to almost any desired topic, however, with the setback that it is time-consuming. The first search engines emerged so as to alleviate the problem of searching for relevant information. Nevertheless, it seems as though listings of the search results of relevant sites are no longer suitable. Thus, additional methods such as extraction algorithms for automatic summaries have been established to help reduce several texts on a topic to the essential contents. Consequently, the main points of the texts are rapidly acquired and clearer to the reader.

This work mainly focuses on automatic summarization of text clusters. Here, we present and analyze different approaches, as well as new methods for the generation of automatic summaries. Moreover, these will be evaluated and compared with human written summaries.

Contents

List of Figures

List of Tables

1 Einleitung

Die Zusammenfassung, also die Kunst relevante Inhalte aus einer oder mehreren Informationsquellen zu extrahieren, ist Teil unseres Alltags geworden. Beispielsweise treffen Leute an der Börse wichtige Investitionsentscheidungen aufgrund aktueller Informationen. Viele Leute stützen sich auf Zusammenfassungen, da innerhalb kürzester Zeit wichtige Entscheidungen getroffen werden können. Obwohl bereits Zusammenfassungssysteme existieren, wird es mit dem wachsenden Informationsvolumen und der rasanten Verbreitung des Internets immer schwieriger inhaltsrelevante und zeitlich bedeutende Zusammenfassungen zu generieren. Weiters sind hochwertige Zusammenfassungen ohne sprachliches Wissen schwer zu erstellen. Es gibt zuviele Variationen hinsichtlich Schreibstil, Genre von Dokumenten, lexikalische Elemente, syntaktische Konstruktionen usw., um ein System zu entwickeln, welches in all diesen Fällen gut funktioniert. Eine ideale Textzusammenfassung sollte Informationen beinhalten, die für den Nutzer interessant sind, und gleichzeitig belanglose und redundante Informationen ausschließen. Weiters sollte sie koherent und verständlich sein. Die automatische Erstellung von solchen Zusammenfassungen ist der Kernpunkt der Diplomarbeit.

Eine weitere Aufgabe dieser Arbeit ist die Analyse von Extraktionsalgorithmen, welche für die automatische Textzusammenfassung genutzt werden können. Auf Basis dieser Analyse sollen Algorithmen in Java implementiert werden und die mit diesen Algorithmen erstellten Zusammenfassungen in einer Evaluation verglichen werden.

1.1 Problemstellung

Die automatische Textzusammenfassung, deren Anfänge bis in die 50er und 60er Jahre zurückreichen [Luh58], hat seit den 90er Jahren immer mehr an Bedeutung gewonnen. Die Ursachen dafür werden in diesem Kapitel näher behandelt. Weiters wird im kommenden Abschnitt der Begriff Textzusammenfassung und deren Bedeutung etwas genauer betrachtet. Das Problem der Informationsüberflutung tritt mit der rasanten Verbreitung des Internets immer stärker in den Vordergrund. Je mehr Texte on-line zur

Verfügung stehen, desto schwieriger wird es, das Informationspotential gezielt zu nutzen. Seither steigt die Notwendigkeit von Suchmaschinen, die das Auffinden von relevanten Informationen erheblich erleichtern, ins Unermessliche. Systeme wie Google oder Altavista kamen dann zum Vorschein, die den Zeitaufwand zum Finden und auch zum Wiederfinden der richtigen Informationen minimieren bzw. erst ermöglichen. Doch um einen Überblick zu einem vom Nutzer ausgewählten Thema zu erhalten, ist eine Auflistung aller in Frage kommenden Seiten nicht mehr angemessen. Sehr viele Texte bringen nämlich sehr viel Überflüssiges und Redundantes mit sich. Somit kann das Suchen nach bestimmten Informationen auch mit großem Zeitaufwand verbunden sein. Zusätzliche Methoden der Extraktionsalgorithmen für automatische Zusammenfassungen können dazu beitragen, Suchmaschinen oder Webverzeichnisse zu optimieren, da diese den Zeitaufwand beim Suchen erheblich vermindern und den Nutzer schnell und hochinformativ mit relevanten Fakten versorgen. Die Aufgabe der automatischen Textzusammenfassung besteht darin, eine bzw. mehrere, ähnliche Dokumente auf den wesentlichen Inhalt zu verkürzen.

Heutzutage kann man Zusammenfassungen in jedem Bereich des täglichen Lebens antreffen. Ob es sich dabei um Nachrichten, Protokolle oder Inhaltsangaben von Büchern handelt, jede dieser Zusammenfassungen ist wesentlich kürzer als das Original und bietet somit einen Überblick über den wesentlichen Inhalt eines Textes, möglichst ohne den Inhalt zu interpretieren oder zu bewerten. Zusammenfassungen sind in der heutigen Gesellschaft somit nicht mehr wegzudenken.

1.2 Textzusammenfassung

Grundsätzlich dienen Textzusammenfassungen dazu dem Nutzer eine übersichtlichere Darstellung umfangreicher Dokumente zu liefern.

Um eine genaue Vorstellung zu bekommen, was man unter einer Textzusammenfassung versteht, werden einige Definitionen näher betrachten:

> "'A summary is a text that is produced out of one or more (possibly multimedia) texts, that contains (some of) the same information of the original text(s), and that is no longer than half of the orignal text(s)."' [Hov98]

"'The main goal of a summary is to present the main ideas in a document in less space."' [Rad02]

In Kapitel 1 werden einige Gründe für die automatische Erstellung von Textzusammenfassungen genannt. Der Begriff Textzusammenfassung wird näher beleuchtet und einige Überlegungen zu dieser Aufgabenstellung angestellt.

Kapitel 2 befasst sich mit dem "'State of the Art"' auf dem Gebiet der Automatischen Textzusammenfassung. Dabei werden die unterschiedlichen Ansätze und Methoden bei der Erstellung von Single- und Multi Document-Zusammenfassungen in den Vordergrund gestellt. Des Weiteren wird in diesem Abschnitt die Entwicklung der Automatischen Textzusammenfassung, von den Anfängen der Forschung in den 50er Jahren bis hin zu aktuellen Forschungsprojekten betrachtet. Dabei wird auf die ersten wegweisenden Algorithmen von Luhn und Edmundson eingegangen. Es werden aber auch Weiterentwicklungen und Anwendungsmöglichkeiten für die automatische Textzusammenfassung näher erläutert.

In Kapitel 3 werden die grundlegenden Methoden der Extraktion und des Preprocessings sowie deren Implementierungen in Java vorgestellt. Begriffe wie Abstract und Extract werden näher betrachtet und Parameter für die Erstellung einer hochwertigen Zusammenfassung vorgestellt. Darüber hinaus werden Programme, die im Rahmen dieser Arbeit verwendet werden, wie das TeSeT zur Unterstützung des Preprocessing-Prozesses für die automatische Zusammenfassung und die SOM ToolBox, welche mit den Funktionen der automatischen Erstellung von Zusammenfassungen erweitert wird, näher erläutert.

In Kapitel 4 werden die vom eigenen System erstellten Zusammenfassungen subjektiv und objektiv evaluiert. Für den subjektiven Teil der Evaluierung werden fünf Juroren eingesetzt, die durch die Beantwortung von Fragen die Zusammenfassungen bewerten. Im objektiven Teil werden von den Teilnehmern Extrakte als Vergleichsobjekt erstellt und mit Hilfe von Methoden, wie beispielsweise die Precision&Recall-Methode, die Ergebnisse beider Arten von Extrakten überprüft. Im Anschluss erfolgt dann die statistische Validierung der Ergebnisse durch den Einsatz eines Signfikanztests.

In Kapitel 5 folgt die Zusammenfassung wie auch einige Möglichkeiten der Verbesserung des eigenen Systems.

2 Related Work in Summarization

2.1 Einleitung

Dieses Kapitel soll einen Überblick über die verschiedensten Ansätze und Methoden der Erstellung von Zusammenfassungen liefern, wobei diese grundsätzlich unter zwei wesentliche Kategorien fallen: *Single Document Summarization* und *Multi Document Summarization*. Dazu werden die wesentlichen Unterschiede zwischen einem Single Document- und einem Multi Document System sowie die grundlegenden Methoden und ihre Implementierungen näher erläutert.

2.2 Unterschiede zwischen Single Document und Multi Document Summarization

Heutzutage ist es enorm wichtig geworden Verfahren zu entwickeln, die Texte auf effiziente Weise suchen bzw. darstellen. Nun gibt es Systeme, wie das *single document summarization system*, die die automatische Generierungen von Extrakten, Abstrakten oder auf Anfragen basierte Zusammenfassungen unterstützen (siehe Abschnitt 2.4.1). Single-Document Zusammenfassungen liefern beschränkt Informationen über den Inhalt eines einzelnen Dokumentes, die den Benutzer bei der Entscheidung, ob er das Dokument weiterlesen soll oder nicht, verhelfen könnte. Betracht man eine Situation, in der ein Benutzer eine Anfrage, wie z.B über ein Thema, welches vor kurzem in den Nachrichten behandelt wurde, an das System stellt. Dieses liefert daraufhin Hunderte von Dokumenten zurück. Obwohl sie sich in einigen Bereichen unterscheiden, werden viele dieser Dokumente vom Inhalt her sehr ähnlich sein und höchstwahrscheinlich diesselben Informationen wiedergeben. Eine Zusammenfassung von jedem einzelnen Dokument würde in diesem Fall helfen, jedoch würden auch diese semantisch ähnlich sein. In der heutigen Gesellschaft, in der die Zeit eine wichtige Rolle spielt, sind *multi-document summa-*

rizer, die ganze Kollektionen von Dokumenten oder einzelne Dokumente im Kontext von vorher generierten Zusammenfassungen zusammenfassen können, in solchen Situationen essentiell.

Eine Studie über das Verhalten eines Users bei der Websuche [Spi02] hat ergeben, dass die Anfragen, die vom Benutzer an das System gestellt werden, meist zu allgemein sind, so dass sie kaum das wiedergeben, was für den Nutzer vom eigentlichen Interesse ist. Das Resultat von solchen Anfragen ist in Normalfall eine Liste von Tausenden Dokumenten, von denen die meisten keinen semantischen Zusammenhang mit der Anfrage des Users haben und vielleicht nur einige Dokumente wichtige Themengebiete abdecken. Darüber hinaus ist ein signifikanter Grad an Überlappung unter den Dokumenten aufgrund des enormen Informationsvolumens nicht ausgeschlossen. Die gleiche Studie hat gezeigt, dass mehr als 70% aller User sich nur eine bis zwei Seiten der Ergebnisse anschauen. Dieses Verhalten deutet darauf hin, dass viele Nutzer eine niedrige Toleranzgrenze für das Durchsuchen von großen Dokumentlisten haben.

In diesem Fall würde eine Multidokument Zusammenfassung die Ähnlichkeiten zwischen den Dokumenten mit zusätzlichen Hinweisen, die den Nutzer auf eine oder mehrere Gruppen von Dokumenten, die der Anfrage entsprechen, aufmerksam machen, hervorheben. Multi Document Zusammenfassungen automatisch zu erzeugen ist dennoch viel komplizierter als die Generierung von Single Document Zusammenfassungen. Mehrere Artikel können nämlich von mehreren Autoren geschrieben worden sein, die jeweils verschiedene Schreibstile und Dokumentstrukturen haben sowie ein anderes Vokabular verwenden. Darüber hinaus besteht die Möglichkeit, dass diese Artikel über denselben Themenbereich gegensätzliche Sichten haben. Ein Multi-Document Summarizer sollte in diesem Fall in der Lage sein solche Situationen zu erfassen und bewältigen. Weiters sollte das System folgende Typen von Dokumentkollektionen, wie auch Browsing-Aufgaben bewältigen können [Gol00]:

1. Der Benutzer hat eine Kollektion von unterschiedlichen Dokumenten und möchte auf die "'Informationslandschaft"' dieser Kollektion zugreifen

2. Der Benutzer hat eine Kollektion von themenbezogenen Dokumenten und möchte die Schlüsselpunkte in dieser Kollektion finden

3. Der Benutzer sucht einen Teil der verfügbaren Informationen und verwendet dafür eine Suchmaschine im World Wide Web, die Tausende Webseiten zurückliefert, von

denen nur einige für den Benutzer relevant sind. Ein multi-document summarizer sollte in diesem Fall in der Lage sein die Schlüssel-Features im Informationsraum zu extrahieren bzw. darzustellen.

2.3 State of the Art

2.3.1 Single Document Summarization

Die Automatische Dokumentzusammenfassung geht weit zurück bis in den 50er, wo Luhns Arbeit von 1958 die erste Implementierung eines Satzextraktionsalgorithmus beschreibt [Luh58]. Danach folgten Arbeiten von Paice und Tait in den 80er [Pai81][Tai83]. Eine Reihe von innovativen Ansätzen wurden daraufhin entwickelt: linguistische Ansätze, statistische und informationsbasierte Ansätze, wie auch Kombinationen von beiden.

Single-Document Summarization durch Extraktion

Obwohl bereits Forschungsarbeiten über mögliche Alternativen zu Extraktion existieren, vertrauen die meisten Arbeiten heutzutage noch immer auf die Extraktion von Sätzen aus dem Originaltext. Die meisten früheren Forschungen beschäftigten sich mit der Entwicklung einer relativ einfachen Technik, die relevante Passagen im Ausgangstext ermittelt. Ein Satz von Features wurde für jede einzelne Passage berechnet, normalisiert und aufsummiert. Die Passagen mit der höchsten Gewichtung wurden daraufhin sortiert und als Extrakt zurückgeliefert.

Die erste Implementierung eines Extraktionsalgorithmus wird von Luhns Arbeit "'Automatic Creation of Literature Abstracts"' von 1958 beschrieben [Luh58]. Als sinnvolles Feature sah Luhn die Häufigkeit, mit der ein Wort im Text auftaucht. Diese Annahme wird durch die Tatsache bestätigt, dass ein Autor bestimmte Worte, die mit dem Thema verbunden sind, bei seiner Begründung und der Beschreibung verschiedener Aspekte wiederholt. Weiters geht er davon aus, dass die Position von relevanten Worten innerhalb eines Satzes etwas über die Relevanz dieses Satzes aussagt.

Der Algorithmus von Edmundson (1969) berücksichtigte für die automatische Textzusammenfassung strukturelle und linguistische Textcharakteristika [Edm69]. Er orientierte

sich am Vorgehen der Menschen beim Schreiben einer Zusammenfassung. Um herauszufinden nach welchen Merkmalen sich Menschen beim Erstellen einer Zusammenfassung richten, ging er folgendermassen vor:

1. Auswahl eines Dokumentenkorpus
2. Analyse von Merkmalen traditioneller Zusammenfassungen von Dokumenten hinsichtlich des Inhalts und der Wiedererkennbarkeit durch den Computer
3. Spezifizierung der gewünschten Form und des gewünschten Inhalts an Hand der manuellen Zusammenfassungen
4. Erstellen von Zusammenfassung für Testdokumente, die sich nach der Spezifizierung des Inhalts und der Wiedererkennbarkeit durch den Computer richten
5. Erstellen eines Systems, welches den Merkmalen, die vom Computer wiedererkannt werden, numerische Werte zuweist
6. Einen Computer so programmieren, dass er automatische Zusammenfassungen erstellt
7. Verbesserung des Verfahrens durch den Vergleich der automatischen Zusammenfassung mit der von Hand erstellten Zusammenfassung
8. Evaluierung des Algorithmus mit Hilfe von neuen, noch nicht verwendeten Dokumenten

Edmundson fand so vier verschiedene Textcharakteristika und erstellte daraus vier Methoden (Cue-, Key-, Title- und Location-Methode), mit denen die Merkmale gewichtet werden konnten. Näheres zu den Methoden wird im Abschnitt 3.3.1 erläutert. Heutige Extraktionsansätze verwenden fortgeschrittene Methoden, um relevante Sätze zu extrahieren. Diese beruhen auf den Methoden des *Machine Learning*, *Natural Language Analysis* sowie der Beziehung zwischen den Worten, anstatt der Verwendung vom Bag-of-Words-Ansatz, der Texte in einem Dokument in einzelne Wörter zerlegt und diese als Attribute definiert.

Pionierarbeit hinsichtlich Applikationen, die auf Machine Learning basieren, haben Kupiec, Pedersen und Chen geleistet, die ein System entwickelt haben, das einen Bayesischen Classifier verwendet, um Feature-Merkmale aus einem Korpus, bestehend aus wissenschaftlichen Artikeln und deren Abstracts, zu kombinieren [Kup95]. Aone et al. und Lin experimentierten mit anderen Formen des Machine Learning und seine Effektivität [Aon99] [Lin99].

Machine Learning wurde auch an *learning individual Features* angewendet. Beispielsweise sind Lin und Hovy [Lin97] durch Machine Learning an das Entscheidungsproblem, wie die Position eines Satzes die Auswahl der Sätze beeinflusst, herangegangen, während Witbrock und Mittal statistische Ansätze verwendeten, um wichtige Worte, Sätze und deren syntaktischen Kontext zu selektieren [Wit99].

Einige Forschungen nutzten den Grad der zwischen den Passagen und dem Rest des Textes vorhandene, lexikalische Verbundenheit. Diese konnte durch die Anzahl der gemeinsam genutzten Worte, Synonyme oder Anapher, welche ein Ausdruck ist, der sich auf vorangegangene Ausdrücke im Text bezieht und wieder aufnimmt, gemessen werden [Sal97] [Man97] [Bar99].

Andere Arbeiten ordneten Passagen hohe Werte zu, die *topic words* enthalten. Topic words sind Worte, die gut mit dem Thema des Nutzers (für themen-orientierte Zusammenfassungen) oder dem allgemeinen Thema im Originaltext korrelieren [Buc97] [Str99] [Rad00].

Eine Arbeit von Conroy und O'Leary verwendet das Hidden Markov Modell (HMMs) und die *pivoted QR decomposition*-Methode, auf der Tatsache beruhend, dass die Wahrscheinlichkeit einer Selektion eines Satzes in einem Extrakt davon abhängt, ob der vorherige Satz ebenfalls inkludiert wurde [Con01].

Ein Open-Source Summarization Toolkit, MEAD, wurde im Johns Hopkins Sommer Workshop entwickelt [Rad02a]. MEAD erlaubt Forschern mit unterschiedlichen Features und Methoden für die Kombination zu experimentieren. Genaueres über das MEAD Toolkit, wie auch andere Implementierungen werden im Abschnitt 2.4 behandelt.

Saggion stellte in seiner Arbeit ein System vor, das die Erstellung von generischen und anfrage-basierten Zusammenfassungen unterstützt [Sag03].

Single-Document Summarization durch Abstraktion

In diesem Kapitel werden auf Methoden der *information extraction, ontological information, information fusion* und *compression summarization* näher eingegangen.

Die *information extraction*-Methode kann auch als "'top-down"' Methode charakterisiert werden, da diese nach einem Satz vordefinierter Informationstypen sucht, welche in die Zusammenfassung inkludiert werden. Für jedes Thema definiert der Nutzer sogenannte *frames*, zusammen mit Informationstypen und Kriterien zur Wiedererkennung. Beispielsweise kann ein Frame über Erdbeben Informationslücken aufweisen, was Ort, Anzahl der Unfälle oder das Ausmass des Erdbebens betreffen. Das Zusammenfassungssystem lokalisiert die gewünschten Informationen, setzt sie ein, und generiert daraus eine Zusammenfassung mit den Ergebnissen [Dej78] [Rau91].

Eine *compressive summarization* erhält man, indem man das Problem aus der Sicht der Sprachgenerierung betrachtet. In der Arbeit von Witbrock und Mittal wird ein Satz von Worten aus dem Eingabedokument extrahiert und entsprechend eines *bigram language* Modells in Sätze umgewandelt [Wit99]. *Bigrams* sind Gruppen von zwei Buchstaben, zwei Silben oder zwei Worte, die als Grundlage für die statistische Analyse von Texten verwendet werden.

Jing und McKeown gehen von der Annahme aus, dass eine von Menschen erstellte Zusammenfassung oft einem Prozess, in dem Dokumentfragmente aus dem Originaldokument kopiert und eingefügt werden, welche dann in der Zusammenfassung zu Sätzen kombiniert und generiert werden, entspricht [Jin99]. So kann ein Summarizer entwickelt werden, der diese Sätze durch Auslassen unwichtiger Fragmente reduziert, und die übrigen Fragmente durch Anwendung von *information fusion* und *information generation* kombiniert. *Information fusion* wendet Methoden an, um bestimmte Informationen aus verschiedenen Quellen zu vereinigen, auch wenn diese sich in ihren konzeptuellen, kontextellen oder typographischen Darstellungen unterscheiden. *Information generation* ist hingegen die maschinelle Ableitung natürlichsprachlicher Texte aus formalen Repräsentationen.

Andere Forschungsarbeiten beschäftigten sich mit dem Reduktionsprozess. In einem Versuch einem System die Reduktionsregeln beizubringen, verwendeten Knight und Marcu die *expectation maximization*-Methode, um ein System zu trainieren, welches einen Syntaxbaum eines Satzes komprimieren soll, mit dem Ziel eine kürzere, aber grammatikalisch korrekte Version zu erzeugen [Kni00]. Man fand aber heraus, dass

diese Methode nicht einer automatischen Zusammenfassung entspricht, sondern eher ein Ansatz für das Kürzen zweier Sätze in einen, drei in zwei (oder einen) usw., ist.

Echte Abstraktion geht einen Schritt weiter. Der Prozess der echten Abstraktion besteht darin zu erkennen, dass ein Satz extrahierter Textpassagen zusammen etwas neues bildet, etwas, das nicht explizit im Originaltext erwähnt werden muss, und diesen mit neuen (idealerweise neueren) Konzepten in die Zusammenfassung zu integrieren. Dass sich neues Material nicht explizit im Text befindet, setzt voraus, dass das System Zugriff zu externen Informationen, einer Art Wissensbasis, haben und auch in der Lage sein muss kombinatorische Inferenz durchzuführen [Hah97].

Im Bereich *Language Modeling* werden Methoden angewendet, um Dokumente zu charakterisieren und durch den Einsatz von graph-theoretischen Algorithmen relevante Themenbereiche für eine hierarchische Zusammenfassung zu bestimmen [Law03].

2.3.2 Multidocument Summarization

Die Multidokumentzusammenfassung, also der Prozess eine einzelne Zusammenfassung aus einer Menge von ähnlichen Dokumenten zu erstellen, ist eine relativ neue Methode der Zusammenfassung. Drei Probleme, die dabei auftauchen, sind:

1. Wiedererkennen und Bewältigung der Redundanz
2. Bestimmung wichtiger Unterschiede unter den Dokumenten
3. Sicherstellung der Kohärenz

In ihrer früheren Entwicklungsphase wurde die Methode der Informationsextraktion verwendet, um eine Möglichkeit der Identifikation von Ähnlichkeiten bzw. Unterschiede unter den Dokumenten zu erzielen [Mck95].

Spätere Arbeiten mischten diesen Ansatz mit der Methode der Regenerierung von extrahiertem Text, um die Erstellung einer Zusammenfassung zu verbessern [Rad98].

White und Cardie folgten diesem Ansatz durch Anwendung einer verbesserten Extraktionsmethode und einer randomisierten, lokalen Suchprozedur, die nach der besten Kombination von Sätzen in einer Zusammenfassung sucht [Whi02].

Um die Redundanz in den Textdokumenten bestimmen zu können, wurde eine Reihe von Methoden verwendet. Eine davon misst die Ähnlichkeit zwischen Satzpaaren und

wendet Clustering an, um die Themen, die in den Dokumenten behandelt werden, zu bestimmen [Mck99] [Rad00] [Mar01].

Ein anderes System verwendet "'relevant novelty"' als potentielles Kriterium zur Minimierung der Redundanz von Dokumenten. Die Relevanz und die Neuheit werden unabhängig voneinander gemessen und deren Linearkombination wird als Metrik geliefert. Der größte Wert dieser Linearkombination wird auch *maximal marginal relevance (MMR)* genannt [Car97] [Car98]. Werden ähnliche Textpassagen in den Dokumenten bestimmt, müssen deren Informationen in die Zusammenfassung einbezogen werden. Anstatt ähnliche Sätze aufzulisten, werden Passagen selektiert, um wichtige Informationen in jedem Cluster zu übermitteln [Rad00].

Andere Ansätze verwenden Informationsfusionstechniken, um sich wiederholende Sätze im Cluster zu bestimmen und diese in der Zusammenfassung zu kombinieren [Bar99a]. Die Arbeit von Mani, Gates und Bloedorn beschreibt den Gebrauch von menschlich-generierten Kompressions- und Formulierungsregeln [Man99].

Kohärenz sicherzustellen ist meist schwierig, da es Verständnis über den Inhalt in jeder Passage und Wissen über die Struktur erfordert. Die meisten Systeme folgen der Ordnung der Zeit und des Textes (Passagen aus älteren Dokumenten erscheinen zuerst, in der Reihenfolge, in der sie in der Eingabe erscheinen). Um den Leser nicht irrezuführen im Falle, dass manche Passagen von unterschiedlichen Daten das Wort "'yesterday"' enthalten, unterstützen einige Systeme explizite Timestamps [Lin02]. Andere Systeme verwenden eine Kombination von Zeit- und Kohärenzeinschränkungen [Bar01].

Einige Forschungen schlagen einen ganz anderen Weg ein und arbeiten an der Generierung von Informationsupdates, die es Nutzern erlaubt über die aktuellsten Informationen über ein bestimmtes Thema informiert zu werden [All01].

Andere wiederum beschäftigen sich mit der Generierung von Titeln, entweder für ein einzelnes Dokument oder ein Satz von Dokumenten. Dieses Problem wurde in einer Evaluierung, die von NIST im Jahr 2003 durchgeführt wurde, näher untersucht [Far03].

2.4 Implementierungen

In diesem Kapitel werden nun einige Implementierungen vorgestellt. Abschnitt 2.4.1 beschreibt die Implementierung eines Single-Document Zusammenfassungssystems. Die weiteren Implementierungen im Abschnitt 2.4.2 und Abschnitt 2.4.3 beruhen auf den Methoden der Multi-Document Summarization.

2.4.1 SUMMARIST

Im Jahre 1995 entwickelten Eduard Hovy und Chin-Yew Lin ein Textzusammenfassungssystem namens SUMMARIST an der Universität von Southern California [Hov99]. Die Zusammenfassung setzt sich aus verschiedenen Komponenten zusammen, die in verschiedenen Modulen implementiert sind. Dabei erfolgte der Prozess der Zusammenfassung in einem dreistufigen Verfahren: Themenidentifikation (*Topic Identification*), Themeninterpretation (*Topic Interpretation*) und die Generierung von Zusammenfassungen (*Summary Generation*). Abbildung 2.1 zeigt die Architektur von SUMMARIST.

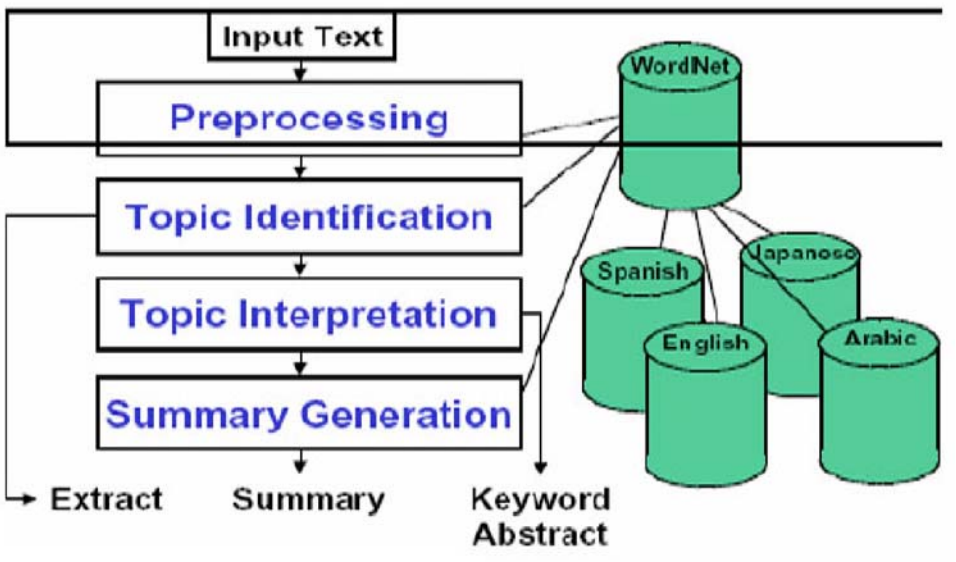


Figure 2.1: Architektur von SUMMARIST [Hov99]

SUMMARIST - Themenidentifikation

Die Berechnung der Relevanz der Sätze erfolgt im SUMMARIST-System anhand einer Reihe von Modulen, die jeweils verschiedene Methoden verwenden. Am Ende werden die Sätze bezüglich ihrer Gewichte, die sich durch die Kombination der Resultate aus den Modulen ergeben, geordnet. Dabei unterscheidet man drei Module:

1. *Postitionsmodul:* dieses Modul geht von der Tatsache aus, dass an bestimmten Postionen einiger Textgenres relevante Sätze vorkommen. Lin und Hovy definierten dafür die Optimal Position Policy (OPP), welche eine Liste mit Positionen von relevanten Sätzen enthält [Lin97].

2. *Cue-Phrase-Modul:* Hinweisphrasen wie "'in summary"', "'in conclusion"', "'the best"' oder "'the most important"' können gute Indikatoren für die Relevanz eines Satzes sein [Edm69]. Dieses Modul sucht nach diesen Hinweisphrasen und ordnet Sätze, die diese enthalten, einem konstanten Wert zu.

3. *Topic-Id-Signature-Modul:* man bezeichnet eine Topic Signatur als ein Themenwort, zusammen mit einer Liste von Wort-Gewicht-Paaren. Hierbei geht man davon aus, dass Worte mit hoher Wahrscheinlichkeit auftreten, sobald ein Schlüsselwort auftaucht. Jedes Vorkommen eines solchen Schlüsselwortes erhält das entsprechende Gewicht. Das Satzgewicht setzt sich dann aus der Summe der Gewichte der Schlüsselworte zusammen, die sich im Satz befinden, normalisiert durch die Länge des Satzes [Hov99].

SUMMARIST - Themeninterpretation

In diesem Schritt werden zwei oder mehrere Themen in ein oder mehrere Konzepte vereinigt. Diese Vorgehensweise erweist sich als der schwierigste Teil der automatischen Textzusammenfassung und wird nur für Zusammenfassungen in Form von Abstrakten verwendet. Das SUMMARIST-System bewältigt diese Aufgabe unter Verwendung zweier verschiedener Methoden:

Interpretation durch Zählung der Inhaltskonzepte

Bei dieser Methode werden nicht Worte, sondern Inhaltskonzepte gezählt. Die Häufigkeiten werden in WordNet, eine lexikalischen Datenbank [Mil95], die semantische und lexikalische Beziehungen zwischen den Wörtern enthält, hinterlegt und aufwärts propgagiert, so dass jeder Knoten als Gewicht die Summe der Gewichte seiner Kinder erhält. Daraufhin überprüft ein top-down Algorithmus bei jedem Knoten, ob dieser eine gute Verallgemeinerung seiner Kinder ist. Dies ist dann der Fall, wenn jeder Kindknoten soviel zum Gewicht des Elternknotens beiträgt wie seine Geschwisterknoten [Kru06].

Interpretation mit Topic Signaturen

Das Topic-Signature-Interpretationsmodul erkennt anhand einer Sammlung von Signaturen ein oder mehrere Themenworte, die die relevanten, vom Themenidentifikationsmodul zurückgelieferten Themengebiete am besten beschreiben. Diese Themenworte werden dann als ein vereinigtes Zusammenfassungskonzept definiert [Kru06].

Generierung von Zusammenfassungen

Demnächst soll das SUMMARIST-System insgesamt drei Module zur Generierung von Zusammenfassungen implementieren. Momentan ist nur das Modul für die Erstellung von Extrakten implementiert, deren Inhalte von der Themenidentifikation bestimmt werden. Bald soll SUMMARIST einen Generator enthalten, der anhand von Wortgruppen oder Teilsätzen einfache Sätze erzeugt. Weiters soll ein zusätzliches Modul wohlgeformte, fließende Zusammenfassungen generieren. Ein Screenshot der Alpha Version von SUMMARIST ist in Abbildung 2.2 abgebildet.

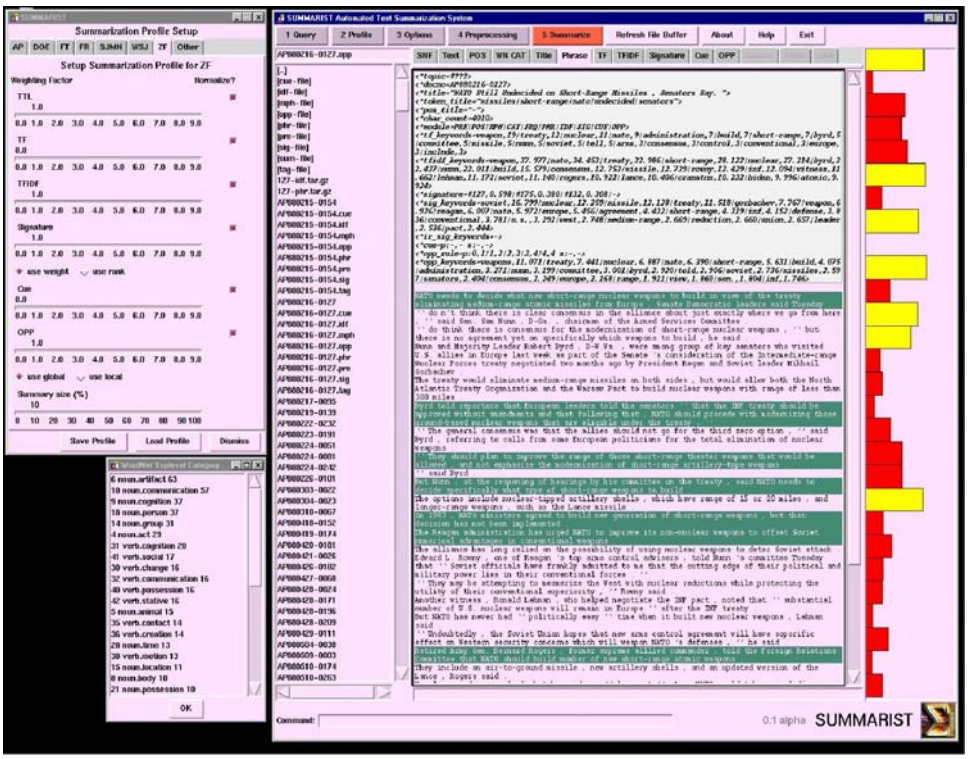


Figure 2.2: Alpha Version von SUMMARIST [Kru06]

2.4.2 SUMMONS

SUMMONS ist ein wissensbasiertes multi document summarization-System, das Zusammenfassungen anhand einer Reihe von sich in einer Domain befindlichen Artikel über Terrorismus produziert. Als Input wird eine Menge semantischer Templates, welche von einem "'message understanding system"' extrahiert wird, an das System geliefert. Dieses erkennt spezielle Muster in diesen Templates wie Änderungen der Perspektive, Widersprüche, Verfeinerungen, Festlegungen und Ausarbeitungen [Mck95].

Die Techniken, die in SUMMONS verwendet werden, erfordern ein großes Maß an Wissen im Bereich Knowledge Engineering, auch für eine relativ kleine Textdomäne und ist für domänenunabhängige Textanalyse nicht geeignet.

SUMMONS-Systemarchitektur

SUMMONS basiert auf einer herkömmlichen Sprachgenerierungssystem-Architektur [Mck85] [Mcd85] [Hov88]. Der Sprachgenerator in diesem System besteht hauptsächlich aus zwei Komponenten, einem *content planner*, der Informationen selektiert, um sie dem Text hinzuzufügen, und einer linguistischen Komponente (*linguistic component*), welche Worte auswählt, um auf in den selektierten Informationen enthaltende Konzepte hinzuweisen.

Der *content planner* erzeugt eine konzeptuelle Repräsentation der Textbedeutungen (z.B. Rahmen, logische Form oder interne Textrepräsentation) und schliesst keine linguistische Informationen ein. Er entscheidet welche Informationen, die aus den Templates stammen, in die Zusammenfassung eingebunden werden. Die linguistische Komponente hingegen verwendet Grammatiken und ein Lexikon, um den Ausdruck und die syntaktische Form der Zusammenfassung zu bestimmen.

Die Ausgabe des Systems ist ein Absatz bestehend aus ein oder mehrere Sätze. Die Länge der Zusammenfassung wird von einem Eingabeparameter bestimmt. Informationen werden nach Wichtigkeit bewertet, wobei Informationen, die nur in einem Artikel auftauchen, niedrige Werte erhalten während Informationen, die von mehreren Artikeln stammen, höher bewertet werden. Abbildung 2.3 zeigt die gesamte Systemarchitektur von SUMMONS.

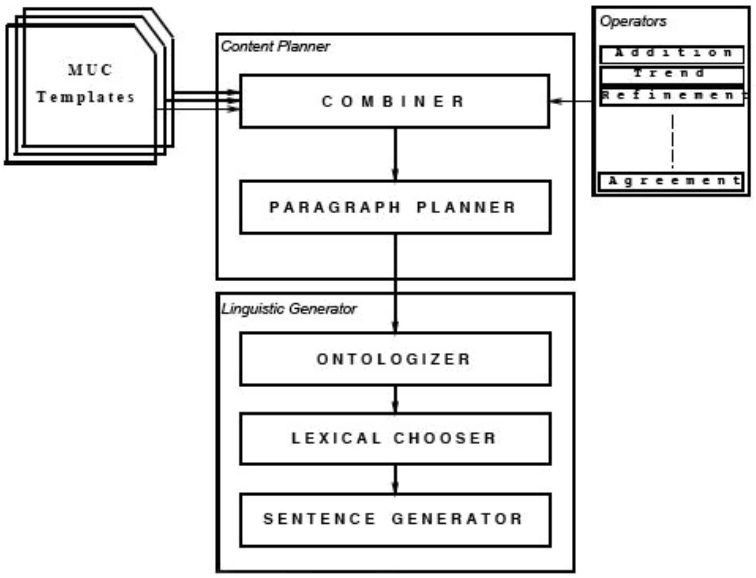


Figure 2.3: Systemarchitektur von SUMMONS [Mck95]

2.4.3 MEAD - ein centroid-basierter Summarizer

MEAD ist ein Toolkit, das mehrere Algorithmen für die Generierung von Zusammenfassungen implementiert, wie *position-based, TFIDF, largest common subsequence* und *keywords* [Rad00]. Es liefert eine Form von Zusammenfassungen zurück, die centroid-basiert sind. Ein *centroid* ist ein Satz von Worten, der für einen Cluster statistisch relevant ist. Dabei können Centroids sowohl der Klassifikation relevanter Dokumente als auch der Identifikation hervorstechender Sätze in einem Cluster dienen.

Centroid-basiertes Clustering

Ähnliche Dokumente werden mit Hilfe eines Algorithmus, der von Radev in Detail beschrieben wird, in ein Cluster gruppiert [Rad99]. Jedes Dokument wird als ein gewichteter Vektor von TFIDF-Werten dargestellt. Zunächst generiert CIDR, ein System für die automatische Unterbringung von Textdokumenten in ein Cluster [Rad99], einen Centroid unter Verwendung des ersten Dokuments im Cluster. Sobald neue Dokumente verarbeitet werden, werden ihre TFIDF-Werte mit dem Centroid anhand der Formel 2.1 verglichen. Befindet sich der Wert $sim(D, C)$ innerhalb eines Grenzbereiches, wird das neue Dokument in den Cluster aufgenommen.

$$sim(D, C) = \frac{\sum_k (d_k * c_k * idf(k))}{\sqrt{\sum_k (d_k)^2}\sqrt{\sum_k (c_k)^2}} \tag{2.1}$$

MEAD Komponenten

Der MEAD Summarizer besteht hauptsächlich aus drei Komponenten: *feature extractor, sentence scorer* und *sentence reranker*.

In jedem Satz berechnet der *feature extractor* Werte für die Features, die vom Benutzer definiert werden. Daraufhin weist der *sentence scorer* jedem Satz einem Wert zu, der eine Linearkombination von den Features ist. Anschliessend werden die Sätze hinsichtlich ihrer Gewichtung sortiert. Die Aufgabe des *sentence rerankers* besteht nun darin diese Sätze, beginnend mit der höchsten Gewichtung, in die resultierende Zusammenfassung einzufügen. Weiters überprüft er die in Frage kommenden Sätze mit den sich in der Zusammenfassung bereits befindlichen Sätzen auf Ähnlichkeit. Ist der Grad

der Übereinstimmung über einem bestimmten Grenzwert, so wird der Satz vom *reranker* ignoriert und der nächste Satz wird betrachtet. Sätze werden solange in die Zusammenfassung hinzugefügt bis die Gesamtanzahl der Sätze dem Wert der Kompressionsrate entspricht.

2.5 Andere Anwendungsbereiche

Da Zusammenfassungen in so vielen Bereichen des täglichen Lebens zu finden sind, sind auch die Einsatzgebiete für Automatische Textzusammenfassungssysteme relativ vielseitig. Ein Einsatzgebiet, nämlich die Multidokument Zusammenfassung, hat man bereits kennengelernt. Im Folgenden sollen einige andere Bereiche behandelt werden [Hah97].

2.5.1 Multiple Languages

Hochwertige, von Maschinen erstellte Übersetzungen sind im Vergleich zu manuell erstellten Zusammenfassungen noch weit unterentwickelt. Was jedoch möglich und durchaus brauchbar für diese Art der Zusammenfassung wäre, ist der Gebrauch eines Filterungsmechanismus. Nutzer könnten solche Filter verwenden, um monolinguale Zusammenfassungen zu erstellen, deren Inhalte ursprünglich aus multilingualen Quellen kommen. Sie könnten dann entscheiden, ob sie weitere Übersetzungen benötigen.

2.5.2 Hybrid Sources

Diese Art der Zusammenfassung kombiniert formatierte Informationen mit unformatiertem, freien Text. Ein Beispiel ist jene Zusammenfassung, welche die Statistik eines Baseball-Spielers aus einer Datenbank mit den Nachrichtenartikeln über diesen Spieler verbindet.

2.5.3 Multimedia

Vor allem wegen ihrer wachsenden Verbreitung, wird Multimedia womöglich eine der wichtigsten Anwendungen der automatischen Zusammenfassung sein. Heutige Methoden

werten Informationen aus dem Audio oder Video aus, um Eigenschaften, die besonders hervorstechen, zu bestimmen.

Ein Projekt aus dem Jahr 1997 beschäftigte sich damit den Inhalt eines Videos mit Hilfe einer Applikation zu untersuchen, die auf Mustererkennung spezialisiert ist und zur Ermittlung von Bereichen, die bestimmte Ereignisse zeigen (Unfälle, Kämpfe, Auftreten des Hauptdarstellers usw), verwendet wird [Lie97].

Abbildung 2.4 zeigt eine Zusammenfassung, erstellt vom Broadcast News Navigator System, ein Tool, welches Search-, Browsing- und Summarize-Funktionen von TV Nachrichtensendungen unterstützt. BNN benutzt dafür eine Reihe unterschiedlicher Präsentationsstrategien, bei denen Keyframes, die automatisch aus dem Video extrahiert werden, mit Zusammenfassungen kombiniert werden. Fortschritte in automatischer Spracherkennung von Audioquellen könnte diese Methode der Zusammenfassung verbessern.

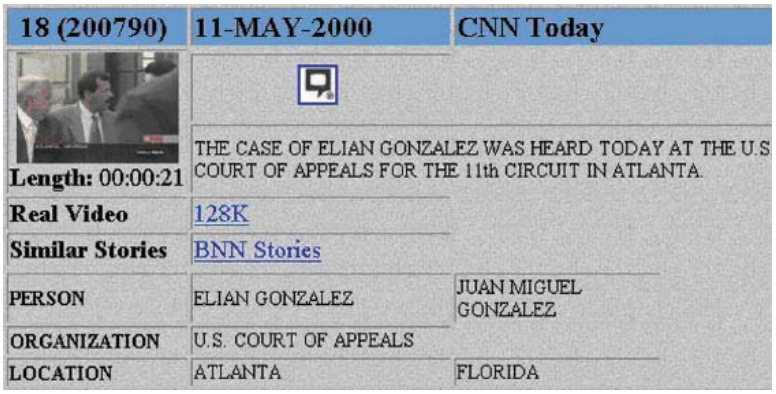


Figure 2.4: Zusammenfassung von BNN-System [Hah97]

2.6 Document Understanding Conference

Das große Interesse und die vielen Aktivitäten im Bereich der Entwicklung von Informationssystemen führten zu einer großen Nachfrage nach unterschiedlichen Methoden der Evaluierung für unterschiedliche Aufgabenbereiche. Ein Grundbaustein legte das TIDES-Programm von DARPA, einer Pentagon-Agentur, die Hightech-Projekte für das

amerikanische Militär durchführt. Bei diesem Programm handelt es sich dabei um eine Sprachverarbeitungstechnologie, die es Menschen ermöglicht, in anderssprachigen Texten relevante Informationen zu interpretieren ohne Wissen über die jeweilige Sprache. Daraufhin bildeten einige Forscher aus dem Programm eine Gruppe, die sich mit der automatischen Textzusammenfassung und deren Evaluierung beschäftigte.

Im Herbst 2000 fand ein zweitätiger Workshop über verschiedene Methoden der Evaluierung von Zusammenfassung statt. Aus diesem Workshop enstand die jährlich stattfindende Document Understanding Conference[1] (DUC). Diese Konferenz soll Forschern die Möglichkeit geben an Experimenten im großen Rahmen teilzunehmen, um somit weitere Entwicklungen auf dem Gebiet der Zusammenfassung zu erzielen [Kru06].

DUC 2001

Im September 2001 wurden generische, Single- und Multidokument Zusammenfassungen von wissenschaftlichen Veröffentlichungen durch Anwendung von intrinsischen Methoden evaluiert. Die Länge der Zusammenfassungen war auf 50, 100, 200 und 400 Worte festgelegt. Insgesamt wurden 60 Sätze von jeweils 10 Dokumenten verwendet. Dabei wurden Schwächen in den Evaluierungsmethoden entdeckt, sodass kein aussagekräftiges Ergebnis hergeleitet werden konnte [Mar01b].

DUC 2002

Im Juli 2002 wurden aus Zeitungsartikeln automatische Zusammenfassungen mit vorgegebener Länge generiert sowie mehrere Projekte für extrinsische Evaluation durchgeführt [Ove03].

DUC 2003

2003 mussten die Zusammenfassungssysteme folgende Aufgaben lösen:

1. die Erstellung von sehr kurzen Zusammenfassungen, die nur bis zu 10 Worte enthalten durften
2. die Erstellung von kurzen Zusammenfassungen, die 100 Worte beinhalten
3. die bis zu 100 Worte enthaltene Zusammenfassung soll basierend auf Dokumentclustern generiert werden

[1] http://duc.nist.gov/

4. kurze, bis zu 100 Worte enthaltene Zusammenfassungen sollen eine bestimmte Frage beantworten

Die Evaluierung erfolgte intrinsisch [Kru06].

DUC 2004

2004 mussten insgesamt fünf Aufgaben für Zeitungsartikel der TDT- und der TREC-Kollektion bewältigt werden, von denen zwei den Aufgaben aus der DUC 2003 entsprachen. Eine der neuen Aufgaben war es kurze Zusammenfassungen von mehreren Dokumenten unter Berücksichtigung von bestimmten Ereignissen zu generieren. Es gab aber auch fragenbezogene Aufgaben. Die Evaluierung der ersten vier Aufgaben erfolgte durch Anwendung von Recall-Oriented Understudy for Gisting Evaluation (ROUGE) [Kru06].

DUC 2005

2005 stand die Entwicklung von neuen Evaluierungsmethoden, die Variationen in manuell verfasste Zusammenfassungen berücksichtigten, im Vordergrund [Dan05]. Die Aufgabe des Systems war es aus einem Satz von 25-50 Dokumenten eine kurze, fließende Antwort zu einer komplexen Frage zu erzeugen. Dieses Experiment zeigte deutlich die Schwierigkeit der automatischen Erstellung von fließenden Multi-Dokument Zusammenfassungen. Als Evaluierungsmethode wurde die von Ani Nenkova und Rebecca Passonneau entwickelte Pyramiden-Evaluation [Nen04] verwendet.

DUC 2006

Im Juni 2006 fand in New York die DUC 2006 statt. Dieselbe Aufgabe aus der DUC 2005 wurde von NIST vorbereitet mit ähnlich großen Dokumentclustern und derselben Längenvoraussetzung für die Zusammenfassungen. Die aus dem Jahr 2005 vorgestellte Pyramid-Evaluation wurde leicht verändert und als Evaluierungsmethode verwendet. Verbesserungen in den Systemen, wie auch in den Evaluierungsprozessen führten zu mehreren signifikanten Unterschieden zwischen den Sytemen als im Vorjahr [Pas06].

2.7 Zusammenfassung

In diesem Kapitel hat man den *State of the Art* der automatischen Textzusammenfassung kennengelernt. Erste Implementierungen und Entwicklungen sowie die wesentlichen Unterschiede zwischen einer Single- und Multi-Document Zusammenfassung wurden näher erläutert. Man hat aber auch eine Reihe von Zusammenfassungssystemen kennengelernt, deren Methoden der Extraktion noch immer auf die grundlegenden Extraktionsverfahren aus den 60er Jahren zurückgreifen.

Einige dieser Methoden, wie beispielsweise die TFIDF-Methode, die die Gewichtung von Wörtern in einem Textdokument berechnet, wie auch die Location-Methode vom SUMMARIST-System, wird in das eigene System integriert und im folgenden Kapitel erläutert.

3 Eigene Implementierung verschiedener Extraktionsalgorithmen

3.1 Einleitung

In diesem Kapitel soll das eigene System wie auch die zu implementierenden Methoden der Extraktion vorgestellt werden. Zuvor wird im Abschnitt 3.2 auf zwei Begriffe, *extract* und *abstract*, näher eingegangen. Weiters wird die Frage "'Was ist bei der Generierung einer Zusammenfassung zu berücksichtigen?"' in den Vordergrund gestellt.

3.2 Parameter für die Erstellung einer Zusammenfassung

Eine Zusammenfassung, die von Menschen erstellt wird, ist meistens eine neuformulierte Wiedergabe der für wesentlich gehaltenen Inhalte eines Textes. Dies ist sehr schwierig zu automatisieren, da dazu unter anderem tiefes Textverständnis, breites Weltwissen und eine gute Spracherzeugung nötig sind. Vorher müssen aber zunächst je nach Verwendungszweck einige Parameter für die Erstellung von Zusammenfassungen berücksichtigt werden.

Um also genau festlegen zu können, welchem Typ der Zusammensetzung das Resultat entsprechen soll bzw. wie zusammengefasst werden soll, sollte man sich folgende Fragen stellen [Kru06]:

- Soll die Zusammenfassung eher einem Extract oder einem Abstract entsprechen?
- Ist eine generische oder nutzer-orientierte Zusammenfassung gewünscht?
- Soll die Zusammenfassung informativ sein oder indikativ?
- Ist ein kohärenter Text oder eine Exzerptsammlung gewünscht?
- Sollen ein oder mehrere Texte als Grundlage für die Zusammenfassung dienen?

Grundsätzlich gibt es zwei Arten von Textzusammenfassungen: ein *Extract* oder ein *Abstract*.

Ein *Extract* ist ein Auszug aus dem Original. Hierbei werden wichtige Textsegmente unverändert aus dem Quelltext extrahiert und je nach Wichtigkeit und Reihenfolge in die Zusammenfassung übernommen. Diese Textsegmente können Absätze, Sätze oder einzelne Wörter sein.

Ein *Abstract* ist im Gegensatz zum Extract eine echte Zusammenfassung. Hier ist es wesentlich, dass die Kernaussage aus dem Text entnommen und umformuliert oder sogar in einer vordefinierten Form ausgegeben wird. Es ist sehr schwierig Abstracts zu automatisieren, da das System über ein gewisses Maß an Verständnis und Hintergrundwissen verfügen muss.

Ob eine allgemeine oder nutzer-orientierte Zusammenfassung gewünscht wird, hängt davon ab, ob alle thematischen Schwerpunkte des Originals wiedergegeben werden sollen, oder ob eher eine Zusammenfassung benötigt wird, die sich nach einer aktuellen Frage oder einer bestimmten Zielgruppe richtet.

Ein wichtiger Aspekt, den man berücksichtigen muss, ist die Festlegung, ob eine informative oder indikative Zusammenfassung generiert werden soll. Bei einer informativen Zusammenfassung wird das Original auf wesentliche Aussagen oder bestimmte Informationen gekürzt. Indikative Zusammenfassungen können für das Herausfiltern der Hauptinformationen im Quelltext und die Erstellung sehr kurzer Texte genutzt werden.

Oft ist ein kohärenter Text das Standardergebnis einer Zusammenfassung, obwohl auch Exzerptsammlungen das Ergebnis einer vom Menschen erstellten Zusammenfassung sein können. Meist liefert das Resultat einer maschinellen Zusammenfassung Extrakte, welche eine Sequenz von Exzerpten aus dem Ausgangstext sind.

Abhängig von der Eingabe, kann eine *single-* oder *multiple-document* Zusammenfassung erzeugt werden. Will man wissen, was in einem bestimmten Text behandelt wird, so wird nur dieser Text für die Zusammenfassung verwendet. Möchte man jedoch Informationen über ein bestimmtes Thema, so können mehrere Texte für den Prozess der Zusammenfassung herangezogen werden.

Alles in allem gibt es eine Reihe von Parametern für ein Zusammenfassungssystem, von denen einige bereits oben diskutiert wurden [Man01]:

1. compression rate (summary length / source length)
2. audience (user focused vs. generic)

3. relation to source (extract vs. abstract)
4. function (indicative vs. informative)
5. coherence (coherent vs. incoherent)
6. span (single- vs. multi- document summarization)
7. language (monolongual, multilingual oder cross-lingual)
8. genre (spezielle Strategien für verschiedene Textvarianten)
9. media (Typmedien oder deren Kombination)

Je nach Anwendung variiert die Wichtigkeit der Parameter. Es ist daher unwahrscheinlich, dass ein Zusammenfassungssystem all diese Parameter berücksichtigt.

Zur Erstellung von Textextrakten gehen die meisten Verfahren satzweise vor und entscheiden dann, ob ein bestimmter Satz zum Extrakt gehören soll oder nicht [Gol99]. Diese Entscheidung kann aufgrund der Position im Dokument, bestimmter Redewendungen (*cue phrases*), nach statistischen Maßen wie Worthäufigkeiten, oder nach semantischer Verwandschaft von Wörtern (*lexical cohesion*) getroffen werden.

Die Lesbarkeit der Extrakte leidet allerdings unter dem fehlenden Zusammenhang der einzelnen Sätze. In der Regel soll daher der Extrakt nicht den Ausgangstext komplett ersetzen, sondern nur verdeutlichen, ob es sich lohnt, diesen zu lesen.

3.3 Methoden zur Erstellung von Textzusammenfassungen

Es gibt mittlerweile eine Reihe von Methoden, wie man an das Problem der automatischen Erstellung von Zusammenfassungen herangeht. Luhn war es, der den klassischen Ansatz der Extraction vorstellte. Nach ihm folgten Arbeiten von Edmundson, Wyllis und Kollegen, die den Grundbaustein für die Modellierung des Zusammenfassens legten (siehe Kapitel 2.3.1). Seither existieren vielseitige Arbeiten zur automatischen Textzusammenfassung, die grundsätzlich unter zwei wesentliche Überschriften fallen: *content-based scoring method, context-based scoring method.*

3.3.1 Content-based scoring method

Diese Methode ist ein offener Ansatz, das heisst es gibt keine vorherigen Annahmen über die Art bzw. die Wichtigkeit des Inhalts. Hierbei wird die Wichtigkeit der Sätze im Quelltext basierend auf dem Inhalt, ohne dessen weitere Interpretation, bestimmt. Das Resultat ist ein durch eine reduzierte Transformation des Quelltexts generierter Text.

Bei diesem Ansatz geht es um das Erkennen und Selektieren der wichtigen Sätze. So werden beispielsweise Textphrasen an Hand ihrer Position (location method) ausgewählt oder an Hand ihrer Gewichtung (key method).

Ein wesentlicher Nachteil dieser Methode ist die begrenzte Kohärenz des erzeugten Textes, da sowohl beim inhaltlichen Zusammenhang als auch bei der Verknüpfung der Sätze große Mängel auftreten können. Im Abschnitt 3.6.3 wird auf folgende Methoden eingegangen: Methode der Worthäufigkeit, Schlüsselwort-Methode, Location-Methode und Titel-Methode.

3.3.2 Context-based scoring method

Diese Methode liefert im Gegensatz zur content-based scoring Methode eine Zusammenfassung, die allein vom Thema abhängig ist. Die Art der Zusammenfassung wird also vom Benutzer bzw. System bestimmt. Man bezeichnet diesen Ansatz auch als einen geschlossenen Ansatz, da der Ausgangstext einer Spezialisierung für zuvor feststehende generische Inhaltsforderungen entspricht. So gibt es Verfahren, die berücksichtigen, dass die Texte, die zusammengefasst werden sollen, stets eine ähnliche Struktur haben wie das Original [Ami00]. Beispielsweise kann dann bei wissenschaftlichen Texten angenommen werden, dass zu Beginn ein Abstrakt, der einen kurzen Überblick über den Text liefert, und am Ende meist eine Zusammenfassung bzw. Schlussfolgerung steht. Mit diesem Wissen kann das System Sätze aus diesen Textteilen extrahieren mit der Gewissheit, die wichtigsten Sätze ausgewählt zu haben.

Eine wesentliche Eigenschaft dieses Ansatzes ist, dass er nur beschränkt einsetzbar ist, da auf Grund der vorherigen Festlegung einer Zusammenfassungsschablone für die Ausgabe immer nur ein Zusammfassungsgebiet besteht.

3.4 Text Preprocessing

Ein großer Nachteil in Text Mining ist die große Anzahl an Worten, die in einem Dokument enthalten sind. Wie man bereits weiß, werden für jedes einzelne Dokument die relevantesten Sätze extrahiert. Die Relevanz der Sätze wird durch die Berechnung der durchschnittlichen Relevanz aller Wörter in dem Satz ermittelt. Wenn nun jedes dieser Worte als Vektorkoordinate dargestellt wird, würde die Anzahl der Dimensionen für den Algorithmus zu hoch werden. Daher ist es ausschlaggebend Preprocessing-Methoden auf den Dokumenten durchzuführen, die die Anzahl der Dimensionen reduzieren.

Dabei werden verschieden Methoden wie z.B. case folding, stemming, Entfernen von Stop Words und N-Grams verwendet. Sobald die Dokumente vorverarbeitet worden sind, kann die eigentliche Textzusammenfassung erfolgen. Im Folgenden werden einige Preprocessing-Methoden vorgestellt:

3.4.1 Case folding

Bei dieser Methode werden alle Zeichen im Dokument in dasselbe Format umgewandelt, sei es in Kleinschreibung oder in großschreibung. Beispielsweise würden die Wörter "‘the"’, "‘The"’, "‘tHe"’, "‘THe"’, "‘thE"’, "‘ThE"’, "‘tHE"’, "‘THE"’ in die als Standard gewählte Kleinschreibung "‘the"’ umgewandelt werden.

3.4.2 Stemming

Stemming bezeichnet ein Verfahren, mit dem verschiedene Varianten eines Wortes auf ihren gemeinsamen Wortstamm zurückgeführt werden. Dieses Verfahren erfordert viel sprachliches Wissen, so dass Algorithmen meistens auf einer Sprache arbeiten. Um nun das Wortstamm eines Wortes zu erhalten, ist es notwendig alle Endsilben zu entfernen. So werden beispielsweise "‘development"’, "‘developed"’ und "‘developing"’ als das Wort "‘develop"’ behandelt.

Zum Stemming gibt es verschiedene Algorithmen für verschiedene Sprachen. Dabei ist die Entwicklung eines Stemmers eine experimentelle Wissenschaft, da Algorithmen nicht verifiziert werden können, sondern erst an Textkorpora, und in der Praxis getestet werden müssen.

3.4.3 Stop word removal

Stop words sind Worte, die sehr oft in einem Dokument vorkommen. Dadurch, dass sie in vielen Dokumenten auftauchen, enthalten diese nur sehr wenig Information über den Inhalt eines Dokuments. Daher werden beispielsweise Stop Wort-Listen erstellt, in denen Wörter wie "'can"', "'will"', "'do"', "'does"', usw. vom Text entfernt werden.

3.4.4 N-Gram

Eine N-Gram Darstellung ist eine Alternative zu Stemming oder Stop word removal. Ein N-Gram kann als Teil eines längeren Strings betrachtet werden. Zum Beispiel wird das Wort DATA als Tri-Grams *_DA, DAT, ATA, TA_*, oder Quad-Grams grams *_DAT, DATA, ATA_* dargestellt werden, wobei das Unterzeichen ein führendes oder zurückhängendes Leerzeichen sein kann.

Im Vergleich zu Stemming und Stop word removal, ist die N-gram Darstellung robuster, da sie gegenüber grammatische und typographsiche Fehler weniger empfindlich ist. Weiters bedarf es bei dieser Methode keiner linguistischen Preprocessing-Methode, wodurch sie sprachunabhängiger ist. Sie hat jedoch den Nachteil, dass sie bei der Reduzierung der Dimensionen nicht effizient ist wie bei der Anwendung von Stemming oder Stop word removal [Net00].

3.5 Berechnung der Satzgewichtung

Die relevanten Sätze, aus denen sich eine Zusammenfassung zusammensetzt, zeichnen sich durch hohe Gewichte aus, da diese darauf hinweisen, dass viele Merkmale, die auf die Wichtigkeit eines Satzes deuten, in dem betreffenden Satz gefunden wurden. Die Gewichtung eines Satzes lässt sich an Hand der Gewichtung für jeden einzelnen Term in dem Satz ermitteln. Die Berechnung dieser Werte und deren Erläuterung erfolgt durch die folgenden Ansätze:

3.5.1 TFIDF

$$w_{ij} = tf_{ij} \cdot log_2 \frac{N}{df_i} \tag{3.1}$$

Der *term frequency/inverse document frequency* Ansatz ist eine allgemein verwendete Methode, mit der man die Gewichtung von Wörtern in Bezug auf ihre Einzigartigkeit in einem Textdokument berechnet. In der Formel 3.1 ist w_{ij} das berechnete Gewicht des Terms i im Dokument j. tf_{ij} entspricht der Häufigkeit des Terms i im Dokument j. N ist die Gesamtanzahl an Dokumenten, während df_i als die Anzahl der Dokumente definiert ist, die den Term i enthalten.

Die Termfhäufigkeit in einem gegebenen Dokument gibt also einen Hinweis auf die Bedeutung dieses Terms für das Dokument. Die inverse Dokumenthäufigkeit hingegen misst die allgemeine Bedeutung des Terms. Die Grundidee hierbei ist, dass Dokumente als Vektoren dargestellt werden, wobei jedes Attribut in diesen Vektoren einem Wort entspricht. Damit die Dimensionaltät des Vektors nicht ins Unermessliche steigt, werden Preprocessing-Methoden verwendet, die im vorherigen Kapitel behandelt werden. Die Gewichtung kann nach unterschiedlichen lokalen und globalen Prinzipien erfolgen [Gue05]:

Lokale Einflüsse

- Die Gewichtung eines Wortes entspricht entweder dessen Häufigkeit im Dokument oder dessen Anzahl des Auftretens in Relation zu dem am häufigsten auftretenden Terms im Dokument.
- Die Gewichtung eines Terms wird an Hand seiner Position beeinflusst. z.B wird ein Wort im Titel als wichtiger eingestuft.

Globale Einflüsse:

- Die Häufigkeit eines Terms wird nicht in Bezug auf ein Dokument, sondern auf mehrere Dokumente betrachtet. Im Falle, dass ein Term in vielen Dokumenten vorkommt, wird dieser als Indexterm für ein bestimmtes Dokument als schwach gewertet.

Berücksichtigt man sowohl lokale als auch globale Einflüsse für einen bestimmten Term, so kann seine Gewichtung bestimmt werden, woraus dann die Gleichung TFIDF resultiert.

Die Grundlage dieser Gleichung basiert auf folgende Annahmen: wenn in einem Dokument ein Term häufiger auftritt als ein anderes, muss dieser Term von größerer Bedeutung sein. Sollte der Term in der Gesamtkollektion seltener auftreten als andere, so kann man davon ausgehen, dass dieser höher bewertet werden sollte.

Ein Problem bei dieser Methode ist, dass in längeren Sätzen einige Terme häufiger auftreten könnten, als in kürzeren Texten. Somit besteht die Gefahr, dass ein Satz eine hohe Gewichtung erhält ohne zwangsläufig eine große Relevanz aufzuweisen. Dieses Problem kann jedoch durch Normalisierung der Sätze gelöst werden, indem das Gewicht noch zusätzlich durch die Satzlänge geteilt wird.

3.5.2 TFISF

Der *term frequency/inverse sentence frequency* Ansatz berechnet die Gewichtung eines Terms nach dessen Auftrittswahrscheinlichkeit innerhalb eines Satzes [Net00b]. Sätze, die also einen hohen TFISF-Wert aufweisen, werden für die Zusammenfassung des Ausgangstextes ausgewählt. Bis auf ein paar Unterschiede ist die Berechnung von TFISF für jedes einzelne Wort der Berechnung von TFIDF ähnlich. Die Ermittlung der Termgewichtung erfolgt anhand folgender Formel:

$$TFISF(W,S) = TF(W,S) \cdot ISF(W) \tag{3.2}$$

wobei die Termhäufigkeit $TF(W,S)$ die Anzahl des Vorkommens eines Wortes W in einem Satz S ist. Die inverse Satzfrequenz $ISF(W)$ hat folgende Form:

$$ISF(W) = \log \frac{|S|}{SF(W)} \tag{3.3}$$

$|S|$ entspricht der Gesamtanzahl von Sätzen während SF(W) die Anzahl von Sätzen, in denen das Wort W vorkommt, beschreibt. Nachdem das durchschnittliche TFISF-Gewicht, auch Avg - TFISFT(S) genannt, für jeden Satz S berechnet wird, werden die relevantesten Sätze, d.h. die Sätze mit den grössten Avg - TFISF Werten über S für die Generierung der Zusammenfassung ausgewählt.

Ein Nachteil hierbei ist, dass Dokumente meistens sehr viele Wörter beinhalten. Die Darstellung jedes Wort als Vektorkoordinate würde somit eine große Anzahl der Dimensionen zu Folge haben. Durch die Anwendung von Text Preprocessing-Methoden kann jedoch eine Reduktion der Anzahl von Dimensionen erzielt werden (siehe Kapitel 3.4).

3.6 Das eigene System

Im Rahmen der Diplomarbeit soll ein Java-Programm entwickelt werden, welches die in vorherigem Kapitel vorgestellten Extraktionsalgorithmen implementiert. Darüber hinaus soll es eine Methode der Multidokument Zusammenfassung unterstützen, welche es erlaubt Gemeinsamkeiten bzw. Unterschiede anhand einer Reihe von Dokumenten zu identifizieren. Dieses Tool wird dann als Erweiterung in ein bereits existierendes Clustering-Programm integriert, um eine automatische Zusammenfassung von Textclustern zu ermöglichen.

Im Folgenden sollen nun zwei Programme vorstellt werden, die zum Zweck des Preprocessings und des Clusterings verwendet werden: TeSeT, SOMToolBox.

3.6.1 TeSeT

Das TeSeT (Term Selection Tool) ist ein Tool zur Unterstützung des Preprocessing-Prozesses für die automatische Erstellung einer Zusammenfassung. Die Funktion von TeSeT besteht darin, Merkmale aus den Daten zu extrahieren und in Vektordateien zur weiteren Verarbeitung zu speichern. Dies erspart eine Menge Zeit und Arbeitsaufwand, da schnell auf Terme, die eine niedrige Termfrequenz aufweisen, zugegriffen und aus der Wortliste entfernt werden können, sodass nur die Terme gewichtet werden, die für den Nutzer von Interesse sind. Darüber hinaus unterstützt das Tool Preprocessing-Methoden

wie Case-Folding, Stemming und das Entfernen von Stop-Worten. In Abbildung 3.1 ist ein Screenshot vom Tool abgebildet.

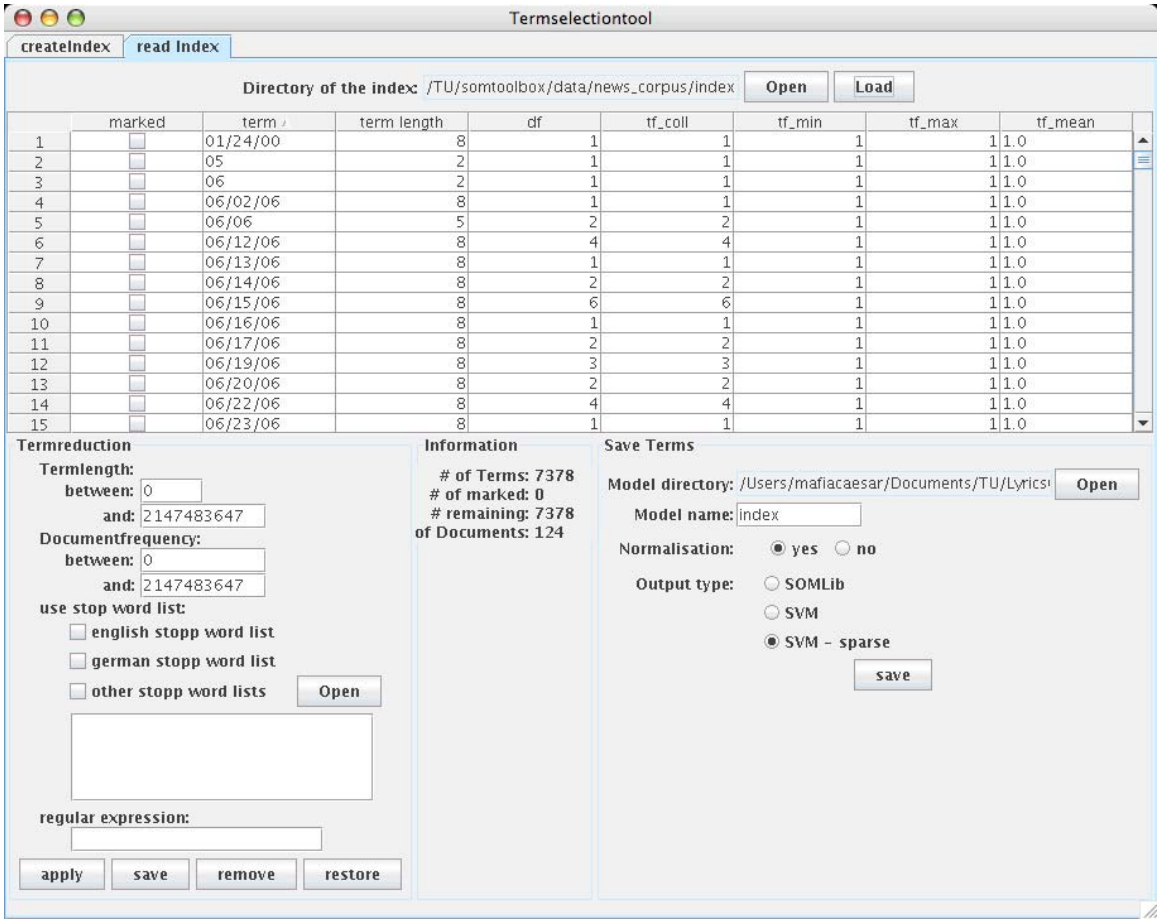


Figure 3.1: Benutzeroberfläche von TeSet

Einlesen und Indizieren von Dokumenten

Das Einlesen bzw. Indizieren von Dokumenten ist aufgrund der einfachen Benutzerobefläche leicht zu bewerkstelligen. Dazu sind zwei einfache Schritte notwendig:

1. Ordner angeben, in dem sich die Dokumente befinden
2. Zielordner angeben, in dem die Indexdateien gespeichert werden

Erstellung einer Vektorrepräsentation der Dokumente

TeSeT hat zwei Listen von Stop-Worten definiert, eine englische und eine deutsche. Durch die Auswahl dieser Listen werden die entsprechenden Stop-Worte selektiert und können aus der Wortliste entfernt werden. Es besteht auch die Möglichkeit eigene stop word-Listen zu definieren. Weiters unterstützt TeSeT *case Folding*.

Um Informationen aus den Daten zu extrahieren, wird eine Vektorrepräsentation aller Dokumente ermittelt. Dies erfolgt durch die Erstellung eines Individual Vector File und eines Template Vector File. Im Individual Vector File wird für jedes Dokument der Datensammlung festgehalten, wie oft ein bestimmtes Merkmal im jeweiligen Dokument vorkommt. Das Template Vector File listet für alle Attribute auf, wie oft diese in wie vielen Dokumenten auftreten. Diese Informationen sind notwendig, da diese als Inputdaten für das Trainieren des SomToolBox verwendet werden.

Abbildung 3.2 zeigt ein Beispiel eines Template Vectors. In $XDIM ist die Gesamtanzahl der Attribute im Template Vector File eingetragen. $YDIM entspricht der Anzahl der eingelesenen Dokumente der Datensammlung. In $VEC_DIM ist die Dimension des Vektors angegeben. Anschliessend werden für jedes Attribut dessen Dokument- und Termfrequenzen aufgelistet.

```
$TYPE template
$XDIM 7
$YDIM 4
$VEC_DIM 8
0 system 71 157 1 11 2.211
1 software 74 144 1 9 1.945
2 research 77 215 1 11 2.792
3 science 77 155 1 12 2.012
4 information 79 145 1 9 1.835
5 university 89 165 1 13 1.853
6 computer 111 273 1 14 2.459
7 technology 115 266 1 14 2.313
```

Figure 3.2: Template Vector File

Das entsprechende Individual Vector File wird in der Abbildung 3.3 dargestellt. Hier werden die TFIDF-Werte der Attribute, gefolgt vom Dateinamen, in der Reihenfolge des Template Vector Files angegeben.

```
$TYPE vec_tfxidf
$XDIM 4
$YDIM 1
$VEC_DIM 8
0 0 3.038 0 0 0 0.647 6.116 data/text_1.txt
2.187 1.052 0 0 0.987 0.867 5.176 0.611 data/text_2.txt
4.375 0 0 0 0 1.735 0.647 0 data/text_3.txt
0 1.052 3.038 8.102 2.961 8.679 2.588 3.669 data/text_4.txt
```

Figure 3.3: Input Vector File

Um eine Normalisierung der Werte auf das Intervall [0,..,1] durchzuführen, braucht man lediglich die Option "'Normalisation"' auswählen. Als Ergebnis erhält man ein durch die Länge normalisiertes Individual Vector File, welches in der Abbildung 3.4 abgebildet ist.

```
$TYPE vec_tfxidf
$XDIM 4
$YDIM 1
$VEC_DIM 8
0 0 0.442 0 0 0 0.0943 0.891 data/text_1.txt
0.370 0.178 0 0 0.167 0.147 0.877 0.103 data/text_2.txt
0.920 0 0 0 0 0.365 0.136 0 data/text_3.txt
0 0.078 0.226 0.603 0.220 0.646 0.192 0.273 data/text_4.txt
```

Figure 3.4: Normalisiertes Input Vector File

3.6.2 Self-Organizing Map(SOM)

Die Self-Organizing Map (SOM) ist eines der bekanntesten neuronalen Netzwerkmodelle des unüberwachten Lernparadigmas. Mit ihrer Hilfe lassen sich Daten auf einer 2-dimensionalen Karte nach deren Ähnlichkeit anordnen, sodass ähnliche Daten nahe beisammen abgebildet werden.

Grundsätzlich besteht das Modell aus einer Reihe von neuronalen Prozesselementen, die als Knoten bezeichnet werden. Jedem dieser Knoten ist ein n-dimensionaler Gewichtsvektor zugeordnet. Zu beachten ist, dass die Dimension der Gewichtsvektoren der Dimension der Inputvektoren, also die Eingabedaten in Vektorform, entspricht.

Der Trainingsprozess des Modells beschreibt ein abwechselndes Präsentieren von Inputvektoren und Adaptieren der Gewichtsvektoren. Dabei wird bei jeder Trainingsiteration versucht die Gewichtsvektoren der adaptierenden Knoten dem Inputvektor ähnlicher zu machen. Diese Adaptierung hat zur Folge, dass Knoten, die eine Ähnlichkeit aufweisen, immer näher zusammenrücken. Das Resultat ist eine räumliche Clusterung von ähnlichen Inputdaten der Self-Organizing Map. Somit erhält man eine topologieerhaltende Abbildung von einem hochdimensionalen Input Space in einen zweidimensionalen Output Space, in dem Muster, die im Input Space ähnlich sind, auch im Output Space auf geografisch nahegelegenen Knoten abgebildet werden.

SOMToolBox

Die SOMToolBox ist eine von vielen Implementierungen des SOM Algorithmus. Sie erlaubt die Darstellung und Interaktion mit Dokumenten und Audiodateien. Als Eingabedaten verlangt die SOM ToolBox das Individual Vector File und den Templatevektor, welche vom TeSeT erstellt werden. Anschliessend werden als Ergebnis des Trainingsprozesses Dateien, wie das Weight Vector, erzeugt. In der Abbildung 3.5 ist ein Screenshot von der SOM ToolBox abgebildet.

Aufbauend auf diesem Programm soll nun ein Tool entwickelt werden, das es erlaubt ein oder mehrere Dokumente mit den vorgestellten Extraktionsmethoden zusammenzufassen.

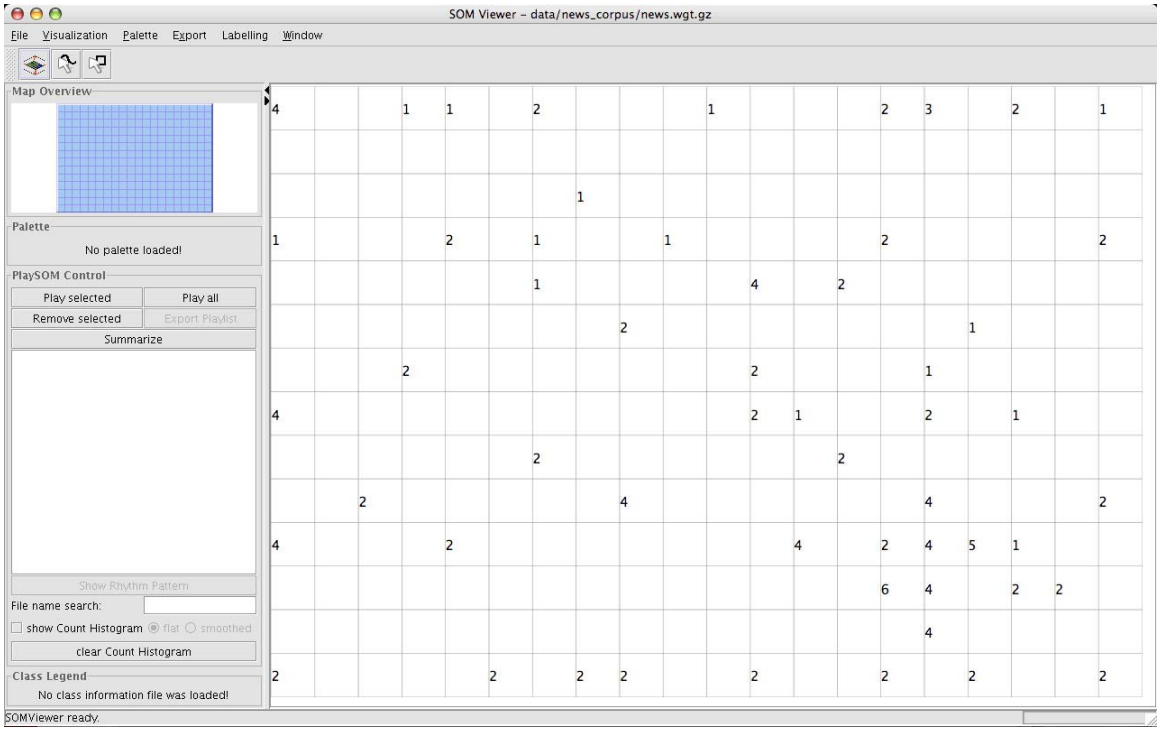


Figure 3.5: Screenshot von SOMToolBox

3.6.3 Implementierung verschiedener Extraktionsalgorithmen

Die Bestimmung relevanter Sätze erfolgt hierbei anhand ihrer Gewichtungen, die, je nach Extraktionsmethode, unterschiedlich errechnet werden.

Zunächst werden die TFIDF-Werte von allen relevanten Worten im Korpus (Sammlung von Texten) mittels TeSeT berechnet und in eine Vektordatei gespeichert. Diese Werte werden dann, je nach Algorithmus, unterschiedlich behandelt. Die Gewichtung der Sätze setzt sich aus der Summe der Gewichtung der sich im Satz befindlichen Worte zusammen. Durch die vom Benutzer ausgewählte Kompressionsrate wird anschliessend die Länge der Zusammenfassung bestimmt. Die Zusammenfassung entsteht dann letztendlich durch die Selektion der am höchsten gewichteten Sätze. Abbildung 3.6 zeigt die grafische Benutzerobefläche des Tools sowie die Selektion der relevanten Sätze je nach Extraktionsmethode. Der Grad der Relevanz der Sätze wird durch Farben hervorgehoben. Es besteht aber auch die Möglichkeit die Gewichte der Worte in den relevanten Sätzen durch diese Farben zu repräsentieren (Abbildung 3.7).

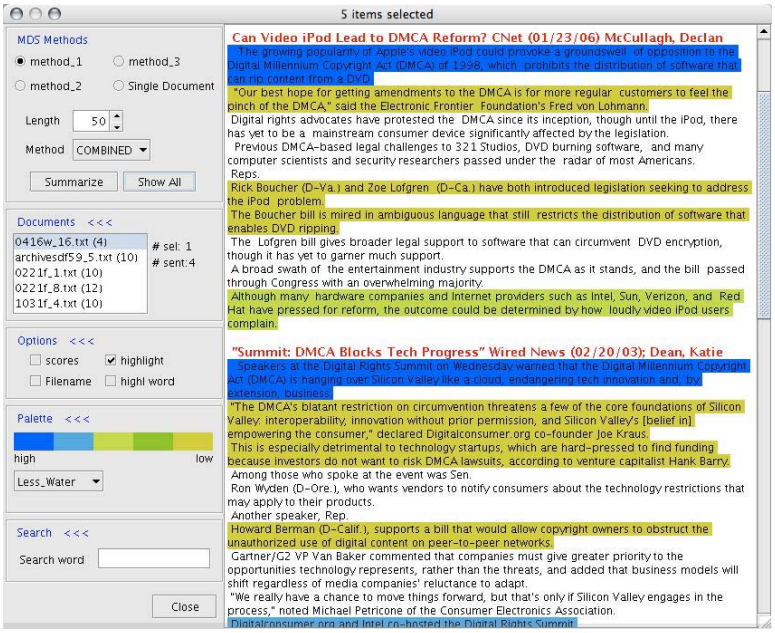


Figure 3.6: Grafische Benutzeroberfläche: Selektion der relevanten Sätze

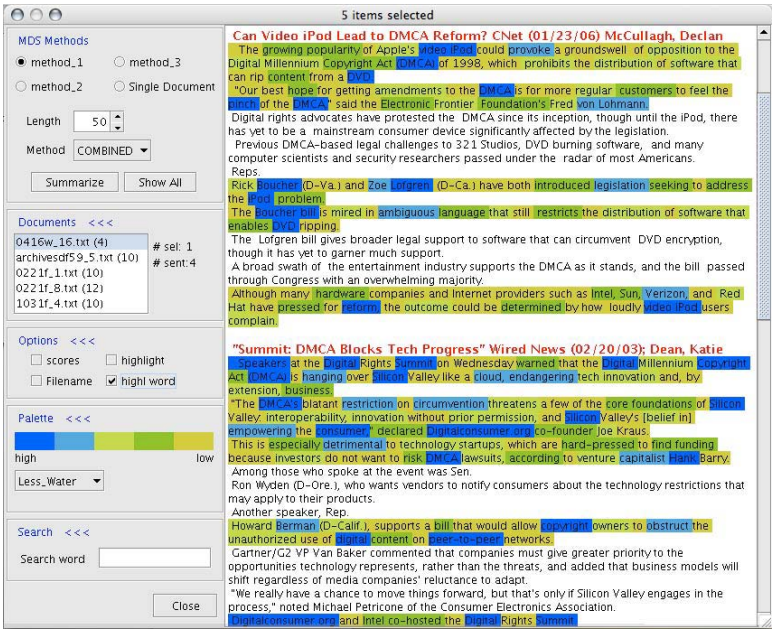


Figure 3.7: Grafische Benutzeroberfläche: Hervorhebung der Gewichte der Wore in den relevanten Sätzen durch Farben

Methode der Worthäufigkeit

Diese Methode geht von der Annahme aus, dass wichtige Worte auf die Worthäufigkeit zurückzuführen sind. Das bedeutet, dass Sätze, die von inhaltlicher Relevanz sind, Wörter enthalten, die im gesamten Text am häufigsten auftreten. Wörtern mit Häufigkeiten, die zwischen einem oberen und unteren Wert liegen, werden Werte zugeordnet, aus denen sich der Gesamtwert des Satzes und somit dessen Wichtigkeit bildet (siehe Abbildung 3.8). Die restlichen Worte, die nicht in die Gewichtung eingehen, werden ignoriert. Die Trennung der zu gewichteten Worte von den restlichen Worten erfolgt durch Anwendung von Preprocessing-Methoden (siehe Abschnitt 3.4).

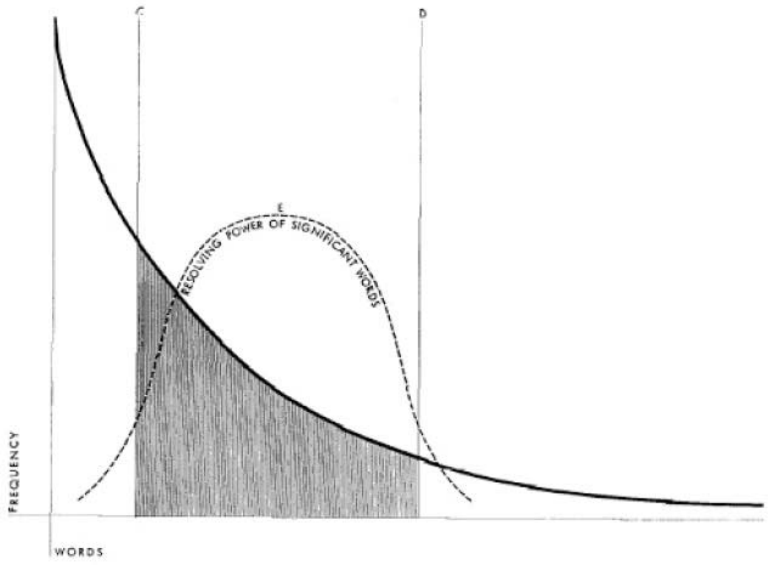


Figure 3.8: Word-Frequency-Diagramm [Luh58]

Die Abbildung zeigt eine Wortliste, geordnet nach der Häufigkeit. Die untere Grenze trennt die relevanten Worte von den selten vorkommenden Worte, die obere von den am häufigsten vorkommenden Worte, so genannte 'stop words'. Diese sind meist Artikel (der, die, das,..), Fragewörter (wer, wie, wann, was,..), Modalverben (wollen, haben,...) usw.

Diese Methode ist durch diese Herangehensweise sehr schwach und daher nur in speziellen Fällen einzusetzen. Ein kurzes Beispiel soll dies verdeutlichen [Hov99]:

John and Bill wanted money. They bought ski-masks and guns and stole an old car from a neighbor. Wearing their skimasks and waving their guns, the two entered the bank, and within minutes left the bank with several bags of 100 $ bills. They drove away happy, throwing away the ski-masks and guns in a sidewalk trash can. They were nevercaught.

Eine Methode für einfache Wortzählung würde nun anzeigen, dass es sich in dieser Geschichte um Skimasken und Waffen handelt, da diese am häufigsten vorkommen. Die Geschichte handelt jedoch eindeutig über einen Raub, und jede Zusammenfassung müsste dies erwähnen.

Zur Berechnung des Satzgewichtes wird nun für jedes Wort im Satz überprüft, ob es im Vektor enthalten ist oder nicht. Ist es im Vektor enthalten, so wird der TFIDF-Wert jenes Wortes, der vom TeSeT errechnet wurde, in die Berechnung des Satzgewichtes miteinbezogen. Es muss sichergestellt werden, dass zwei Worte, die verglichen werden, trotz großschreibung als identisch angesehen werden. Eine Methode convertUpper2LowerCase() löst dieses Problem und wird vor jeder Berechnung der Satzgewichte ausgeführt. Da das TeSeT das Stemming-Verfahren unterstützt, werden Worte wie "'development"', "'developed"' und "'developing"' ebenfalls als identisch angesehen.

Schlüsselwort-Methode

Hier wird davon ausgegangen, dass der Autor des Textes bestimmte Schlüsselworte verwendet, um seine Meinungen zum Ausdruck zu bringen. Diese Methode sucht nach diesen Schlüsselworten und ordnet Sätzen, die diese enthalten, Werte zu. Es gibt eine Reihe von Varianten, die diese Methode implementieren [Kru06]:

- die Schlüsselwörter werden in jedem Satz gezählt und gespeichert; die Zusammenfassung entsteht dann aus der Selektion der Sätze mit den grössten Werten
- die in der Überschrift enthaltende Wörter werden als Schlüsselwörter definiert; es werden Sätze ausgewählt, die diese Schlüsselwörter enthalten
- Zusammenfassungen, die auf Anfragen basieren oder benutzerorientiert sind, benutzen Worte, die Anfrageworte oder vom Interesse des Benutzers sind, als Schlüsselworte

- nur Substantive oder Verben werden als Schlüsselworte verwendet, da diese relevant sind

In unserem Fall werden nur Substantive oder Verben als Schlüsselworte behandelt. Der Benutzer hat die Möglichkeit Substantive, Verben oder beides als Schlüsselworte zu definieren. Die Berechnung des Satzgewichtes erfolgt im Prinzip genauso wie die Berechnung der Worthäufigkeiten, nur dass Worte, die nicht als Substantive oder Verben erkannt werden, aus der Berechnung ausgeschlossen werden. Um Substantive und Verben in einem Satz identifizieren zu können, wird ein Programm namens LingPipe[2] verwendet, welches Part-of-Speech Tagging unterstützt.

Wird das Wort als Substantiv oder Verb erkannt, wird der TFIDF-Wert des Wortes als Gewicht zum bisherigen Keygewicht addiert. Die Satzgewichtung errechnet sich durch die Summe der Gewichte geteilt durch die Gesamtanzahl der relevanten, sich im Satz befindlichen Worte.

Location-Methode

Edmundson [Edm69] geht bei dieser Methode davon aus, dass in einigen Bereichen eine Regelmässigkeit der Struktur des Dokuments aufweist, so dass aufgrund der Platzierung bestimmter Sätze, diese dazu neigen, mehr über den Inhalt auszusagen als andere. Dabei basiert die Methode auf folgende Annahmen:

1. die ersten Sätze eines Abschnitts sind relevant
2. Sätze von inhaltlicher Relevanz sind meist in der Nähe des Beginns oder des Endes des Dokumentes

Da der erste Satz eines Dokuments, wie auch der letzte, zumeist relevant ist, wird diesen zusätzlich ein höheres Gewicht zugeordnet. Diese Locationgewichte werden dann zum Satzgewicht addiert, das sich vorher durch die Methode der Worthäufigkeit errechnen lässt.

[2] http://www.alias-i.com/lingpipe/index.html

Titel-Methode

Die Überschrift eines Dokuments verrät vieles über das Thema, das im Dokument hauptsächlich behandelt wird. Sie übermittelt dem Leser eine generelle Vorstellung vom Inhalt. Daher wird bei dieser Methode die Relevanz eines Satzes durch die durchschnittliche Häufigkeit der Titelworte (ausgenommen Stop-list Worte) in diesem Satz bestimmt. Dabei wird für jeden Satz folgender Wert berechnet [Lam01]:

$$gew(S) = \frac{tts(S)}{ttt} \tag{3.4}$$

Die Gewichtung eines Satzes $gew(S)$ entspricht also der Gesamtanzahl von Titelbegriffen $tts(S)$, die im Satz vorkommen, geteilt durch die Gesamtanzahl von Titelbegriffen ttt (Formel 3.4). Dieses Gewicht wird anschliessend zum Satzgewicht addiert, welches man vorher durch die Methode der Worthäufigkeit erhält.

Kombinierte Gewichtung

Das Berechnen der kombinierten Gewichtung erfolgt durch einfaches Addieren aller vier Methodengewichte. Dabei steht s_i für den zu gewichteten Satz, $W(s_i)$ für das Ergebnis der Methode der Worthäufigkeit, $K(s_i)$ für das der Schlüsselwort-Methode, $L(s_i)$ für das Resultat der Location-Methode und $T(s_i)$ für das Ergebnis der Titel-Methode (Formel 3.5).

$$SCORE(s_i) = W(s_i) + K(s_i) + L(s_i) + T(s_i) \tag{3.5}$$

3.6.4 Implementierung der Multidocument Summarization

Nachdem die grundlegenden Methoden der Single-Document Zusammenfassung implementiert wurden, sollen nun einige Ansätze der Multi-Document Zusammenfassung real-

isiert werden. Vorher wird noch ein Algorithmus vorgestellt, der Redundanzen ermittelt.

Redundanz-basierter Algorithmus

Der von [Rad00] hergenommene Ansatz wurde implementiert, um zu verhindern, dass Sätze wiederholt Informationen liefern, die bereits aus einem anderen, bereits in der Zusammenfassung aufgenommenen Satz stammen.

Dabei spricht man von einer *subsumption*, wenn der Informationsgehalt eines Satzes a im Satz b enthalten ist (b *subsumes* a, was soviel bedeutet wie: b subsumiert a). Im Beispiel unten subsumiert Satz 2 Satz 1, da die Information in Satz 1 auch in Satz 2 enthalten ist, der zusätzlich noch folgende Informationen enthält: *the court, last August*, und *sentenced him to life*.

1. John Doe was found guilty of the murder.
2. The court found John Doe guilty of the murder of Jane Doe last August and sentenced him to life.

Für jedes Paar von Sätzen, das in die Zusammenfassung aufgenommen werden soll, wird folgender Wert errechnet:

$$R_i = 2 * \frac{ow}{w1 + w2} \tag{3.6}$$

wobei ow die Anzahl der überlappenden Worte (*overlapping words*), $w1$ die Anzahl der Worte im Satz 1, und $w2$ die Anzahl der Worte im Satz 2 ist.

Wird ein bestimmter Wert erreicht, so werden die zwei Sätze als ähnlich erkannt, woraufhin der Satz mit der grösseren Gewichtung in die Zusammenfassung übernommen wird. Dieser Algorithmus dient nicht nur dazu, den Grad der Redundanz einer Zusammenfassung zu reduzieren, man wird auch in den folgenden Abschnitten sehen, dass er gut dafür geeignet ist Gemeinsamkeiten aus einem Cluster von Dokumenten zu ermitteln.

Gleichzeitige Erstellung mehrerer Zusammenfassungen (1.Implementierung)

Diese Methode der Multi-Document Zusammenfassung arbeitet im Prinzip wie die Single-Document Zusammenfassung mit dem Unterschied, dass statt ein einzelnes Dokument mehrere Dokumente gleichzeitig bearbeitet werden können. Das verschafft dem Leser die Möglichkeit schnell über die wichtigsten Punkte aller Dokumente in einem Cluster informiert zu werden. Wie bei der Single-Document Zusammenfassung können auch hier die Extraktionsmethoden und die Länge der Extrakte vom Nutzer bestimmt werden, wobei die Änderungen dann auf alle selektierten Dokumente wirksam werden. Als Ergebnis wird eine Liste von Zusammenfassungen, die jeweils aus einem Dokument stammen, ausgegeben.

Erstellung einer redundanz-armen Zusammenfassung (2.Implementierung)

Wodurch sich diese Methode von der 1.Implementierung unterscheidet, ist die Betrachtung der Dokumente als Ganzes. Die Zusammenfassung entsteht hierbei durch die Selektion der am höchsten gewichteten Sätze, die aus allen selektierten Dokumenten, und nicht, wie bei der 1.Implementierung, aus einem Dokument, stammen. Weiters werden die Sätze, noch bevor sie in die Zusammenfassung aufgenommen werden, paarweise auf Redundanz untersucht. Da diese Dokumente aus verschiedenen Quellen stammen und mindestens ein Thema gemeinsam behandeln, ist es unvermeidlich, dass redundante Informationen enthalten sind. Um diese zu identifizieren, wird der redundanz-basierte Algorithmus verwendet. Hat das Ergebnis des Algorithmus, das durch Vergleich eines Satzpaares entsteht, einen Mindestwert, so wird der Satz mit der grösseren Gewichtung in die Zusammenfassung aufgenommen. Der andere Satz wird ignoriert. Als Resultät liefert das System eine Zusammenfassung, deren Inhalt aus mehreren Dokumenten stammt, wobei eventuelle Gemeinsamkeiten unter den Dokumenten ausgeschlossen werden.

Im Folgenden werden zwei ähnliche Dokumente gezeigt, die mit Hilfe der soeben genannten Methode zusammengefasst werden.

[1] When time is the enemy, Baylor students meet the challenge.
[2] Stephenville sophomore Nick Soltau won the first round of the International Collegiate Programming competition this year, held Tuesday at Rogers Engineering and Computer Science Building.
[3] Assistant Professor of Computer Science Dr. Greg Hamerly and Associate Professor Dr. David Sturgill organized and judged the programming contest, which challenged competitors to solve real-life problems using their programming skills.
[4] Students were asked to write programs – to sort and track student records, to determine the optimal operating hours for a store, to cut printing costs and to handle secure communications.
[5] Points are earned for solving the most problems the fastest with the least amount of penalties.
[6] "The difficulty is how to utilize a variety of competencies in Computer Science and get it right with the pressure of time against them," Hamerly said.
[7] "The competition was fun but the time constraints intensified the level of difficulty.
[8] The hardest part was deciding whether you will be able to solve a problem before investing time to work on it" Soltau said.
[9] This year's prizes included gift certificates from Altex Electronics, computer equipment and valuable programming books.
[10] The local programming contest is an individual competition, but the regional and international contests are team competitions.
[11] Winners from Baylor will travel to the regional competition in November where they'll compete against other teams to solve real life problems in a five-hour time limit.
[12] Regional winners will then advance to the international competition.
[13] The international contest is funded through IBM and put on by the Association of Computing Machinery (ACM) International.
[14] Baylor recently hosted the 30th annual ACM International Collegiate Programming Contest World Finals in San Antonio in April.
[15] Baylor organized the World Finals for the 83 teams of students that were narrowed down from the 5,606 teams selected from 1,737 universities in 84 countries around the world.
[16] The competition is open to undergraduates and first-year graduates from all majors.

[1] The regional competition of the International Collegiate Programming contest in November is up next for Baylor University programming students.
[2] Earlier this week, Baylor held the first round of the International Collegiate Programming competition, which was won by Nick Soltau, a sophomore from Stephenville, Texas.
[3] Soltau and other participating students faced the challenge of writing programs that monitor student records, identify the best operating hours for a store, reduce printing costs, and manage secure communications.
[4] Soltau was able to solve the most real-world problems in the least amount of time and with the fewest penalties.
[5] "The difficulty is how to utilize a variety of competencies in computer science and get it right with the pressure of time against them," says Dr. Greg Hamerly, an assistant professor of computer science who helped organize and judge the competition.
[6] Soltau will team up with other top Baylor competitors for the regional competition, with hopes of advancing to the international contest, which, under the auspices of ACM, is sponsored by IBM.
[7] Last April, the 30th annual ACM International Collegiate Programming Contest World Finals drew 83 teams to San Antonio.

Das System ermittelt Ähnlichkeiten zwischen den Dokumenten und gibt eine redundanzarme Zusammenfassung zurück:

[1] The regional competition of the International Collegiate Programming contest in November is up next for Baylor University programming students.
[2] Earlier this week, Baylor held the first round of the International Collegiate Programming competition, which was won by Nick Soltau, a sophomore from Stephenville, Texas.
[3] When time is the enemy, Baylor students meet the challenge.
[4] Assistant Professor of Computer Science Dr.Greg Hamerly and Associate Professor Dr.David Sturgill organized and judged the programming contest, which challenged competitors to solve real-life problems using their programming skills.
[5] Points are earned for solving the most problems the fastest with the least amount of penalties.
[6] "The difficulty is how to utilize a variety of competencies in Computer Science and get it right with the pressure of time against them," Hamerly said.
[7] "The competition was fun but the time constraints intensified the level of difficulty.
[8] This year's prizes included gift certificates from Altex Electronics, computer equipment and valuable programming books.
[9] The local programming contest is an individual competition, but the regional and international contests are team competitions.
[10] Winners from Baylor will travel to the regional competition in November where they'll compete against other teams to solve real life problems in a five-hour limit.
[11] Regional winners will then advance to the international competition.
[12] The international contest is funded through IBM and put on by the Association of Computing Machinery (ACM) International.
[13] The competition is open to undergraduates and first-year graduates from all majors.

Identifzierung von Ähnlichkeiten unter den Dokumenten (3.Implementierung)

Diese Art der Multi-Document Zusammenfassung ermittelt Ähnlichkeiten von mehreren Dokumenten in einem Cluster und stellt diese dar. Die Ermittlung von Ähnlichkeiten erfolgt ebenfalls unter Verwendung der vorgestellten, redundanz-basierten Methode. Werden zwei Sätze inhaltlich als ähnlich erkannt, so wird der Satz mit der grösseren Gewichtung in die Zusammenfassung aufgenommen. Somit erhält man eine Zusammenfassung, die nicht, wie bei der 2.Implementierung, Gemeinsamkeiten ausschliesst, sondern diese ausgibt.

Diese Vorgehensweise soll anhand eines Beispiels verdeutlicht werden. Nimmt man die zwei ähnlichen Dokumente aus dem letzten Beispiel, so sieht man im folgenden Text die Sätze, die mit Hilfe der redundanz-basierten Methode als ähnlich erkannt werden:

[2a] Stephenville sophomore Nick Soltau won the first round of the International Collegiate Programming competition this year, held Tuesday at Rogers Engineering and Computer Science Building. (0.29)
[2b] Earlier this week, Baylor held the first round of the International Collegiate Programming competition, which was won by Nick Soltau, a sophomore from Stephenville, Texas. (0.51)

[4a] Students were asked to write programs – to sort and track student records, to determine the optimal operating hours for a store, to cut printing costs and to handle secure communications. (0.26)
[3b] Soltau and other participating students faced the challenge of writing programs that monitor student records, identify the best operating hours for a store, reduce printing costs, and manage secure communications. (0.28)

[6a] "The difficulty is how to utilize a variety of competencies in Computer Science and get it right with the pressure of time against them," Hamerly said. (0.42)
[5b] "The difficulty is how to utilize a variety of competencies in computer science and get it right with the pressure of time against them," says Dr. Greg Hamerly, an assistant professor of computer science who helped organize and judge the competition. (0.26)

[14a] Baylor recently hosted the 30th annual ACM International Collegiate Programming Contest World Finals in San Antonio in April. (0.27)
[7b] Last April, the 30th annual ACM International Collegiate Programming Contest World Finals drew 83 teams to San Antonio. (1.33)

Dabei werden die Sätze, die aus dem einen Dokument stammen, mit "‘a"’ und die aus dem anderen Dokument mit "‘b"’ versehen. Die Zahlen in den eckigen Klammern entsprechen den Positionen der jeweiligen Sätze in den ursprünglichen Dokumenten. Das Gewicht steht am Ende jedes Satzes. Das System nimmt die Sätze mit der grösseren Gewichtung und liefert als Ergebnis eine Zusammenfassung in folgender Form zurück:

[1] Earlier this week, Baylor held the first round of the International Collegiate Programming competition, which was won by Nick Soltau, a sophomore from Stephenville, Texas.
[2] Soltau and other participating students faced the challenge of writing programs that monitor student records, identify the best operating hours for a store, reduce printing costs, and manage secure communications.
[3] "The difficulty is how to utilize a variety of competencies in Computer Science and get it right with the pressure of time against them," Hamerly said.
[4] Last April, the 30th annual ACM International Collegiate Programming Contest World Finals drew 83 teams to San Antonio.

3.7 Zusammenfassung

Wie man gesehen hat, gibt es beim automatischen Erzeugen von Zusammenfassungen eine Vielzahl unterschiedlicher Ansätze. In diesem Kapitel wurde das eigene Textzusammenfassungssystem vorgestellt. Einige Methoden, die man aus dem vorherigen Kapitel kennengelernt hat, wurden in das System übernommen, es wurden aber auch Extraktionsmethoden zum Zwecke der Multi-Document Zusammenfassung implementiert.

Im folgenden Kapitel 4 werden die automatisch erzeugten Zusammenfassungen nun in einer Evaluation mit manuell verfassten Zusammenfassen verglichen und bewertet.

4 Evaluierung

4.1 Einleitung

In diesem Kapitel werden zunächst auf zwei Arten von Evaluierungsmethoden, nämlich die intrinsische und extrinsische, etwas näher eingegangen. Es folgt dann die Durchführung der subjektiven und objektiven Evaluierung unter Verwendung von Juroren. Dabei wird die automatische Zusammenfassung mit denen der Juroren verglichen und ausgewertet. Zum Schluss werden die Ergebnisse beider Arten mit Hilfe eines Signifikanztests auf Unterschiede überprüft.

4.2 Evaluierungsmethoden

Man unterscheidet grundsätzlich zwei Klassen von Evaluierungsmethoden, die intrinsische und extrinsische Evaluierung. Während die intrinsische Evaluierung die Qualität der Zusammenfassung auf Grundlage von Analysen des Inhalts der Zusammenfassung bezüglich eines Satzes von Regeln bewertet, überprüft die extrinsische Evaluierungsmethode wie gut die automatisch erstellten Zusammenfassungen Menschen bei der Erfüllung einer bestimmten Aufgabe unterstützen. Diese Aufgaben können Leseverständnis-, Kategorisierungs-, Retrievalaufgaben usw. sein.

Beide Evaluierungsmethoden haben ihre Vor- und Nachteile. So ist ein Vorteil von intrinsischen Evaluationen, dass sie bereits in frühen und mittleren Entwicklungsstufen des Zusammenfassungssystems eingesetzt werden können, während die extrinsischen Methoden erst angewendet werden können, wenn das System komplett entwickelt ist. Allerdings sind die Nachteile intrinsischer Evaluationen die Beschränkung auf wenige Referenzzusammenfassungen. Darüber hinaus ist die geringe Aussagekraft über die Qualität einer Zusammenfassung bei Verwendung einer intrinsischen Evaluierungsmethode als ein großer Nachteil zu betrachten. Weiters muss erwähnt werden, dass die intrinsische Methode wesentlich kostenspieliger ist, als die extrinsische Methode.

Um nicht nur die Qualität, sondern auch die Nützlichkeit der automatisch erzeugten Zusammenfassungen bewerten zu können, sollten sowohl intrinsische als auch extrinsische Evaluationsverfahren eingesetzt werden. Dies würde aber zu einer viel zu umfangreichen Evaluierung mit zu hohem Zeitaufwand, sowohl für die Evaluierungsteilnehmer, als auch für die Auswertung führen. Aus diesem Grund wird für eine rein intrinsische Evaluierung entschieden. Das sind jene Evaluierungsmethoden, die auf Vergleichen mit einer "'idealen"' Zusammenfassung beruhen. Die Erstellung der "'idealen"' Zusammenfassungen erfolgt durch den Einsatz von Juroren, die jeweils eigene, für sich als "'ideal"' angesehene Zusammenfassungen erstellen. Dabei sollten sie in verschiedenen Texten die für sie wichtigsten Sätze markieren. An dieser Stelle sollte berücksichtigt werden, dass jeder Mensch eine eigene Vorstellung von einer guten Zusammenfassung hat. Von daher gibt es keine richtige Zusammenfassung und auch keine endliche Menge richtiger Zusammenfassungen. Es besteht durchaus die Möglichkeit, dass ein System eine völlig andere Zusammenfassung erstellt, die von der Aussagefähigkeit genau so gut ist.

4.3 Durchführung der Evaluierung

4.3.1 Auswahl der Texte für die Evaluierung

Für die Evaluierung werden drei verschiedene Text-Corpora verwendet: der ACM-, der Banksearch-, und der Lyrics-Corpus.

Das ACM (Association for Computing Machinery) Data Set umfasst 10000 Artikeln der letzten sieben Jahre. Der gesamte Corpus ist frei verfügbar unter

http://technews.acm.org/archives.cfm.

Die Banksearch Kollektion besteht ebenfalls aus ca. 10000 Dokumenten, unterteilt in 10 gleich großen Kategorien mit jeweils 1000 Dokumenten. Jeder Kategorie wird eines von vier unterschiedlichen Themen, nämlich *Banking and Finance, Programming Languages, Science*, und *Sport* zugeordnet.

Der Lyrics-Corpus ist eine Ansammlung von Liedtexten aus über 20 Genres bestehend aus ca. 8000 Dokumenten.

Obwohl sich Banksearch- und Lyrics-Corpus nicht zum Zusammenfassen eignen, da aufgrund mangelnder Formatierung keine sinnvolle, automatisch erstellte Zusammenfassung entstehen würde, werden diese trotzdem für die Evaluierung verwendet, um die

Flexibilität des Programms zu prüfen. Die dabei auftretenden Probleme werden im Abschnitt 4.3.3 geschildert.

4.3.2 Auswahl der Juroren

Es werden insgesamt fünf Juroren für die subjektive und objektive Evaluierung eingesetzt. Um die Verallgemeinerungsfähigkeit der Bewertungen abzusichern, sind Juroren aus verschiedenen Altersgruppen anzuwerben. Zwei von ihnen sind unter 27, eine über 30, und die übrigen über 40 Jahre alt. Drei von ihnen sind bereits berufstätig, zwei Juroren sind Studenten an der Technischen Universität. Zwei von ihnen stufen sich im Bereich Computer als Anfänger ein. Die restlichen besitzen eine hohe Kompetenz im Bereich Computernutzung und Wirtschaft. Im Zuge der Evaluierung sind die zwei Studenten als Juror 3 bzw. 4, die über 30 jährige Testperson als Juror 1, und die über 40 jährigen Teilnehmer als Juror 2 bzw. 5 zu bezeichnen.

Eines der Ziele dieser Evaluierung ist es nun festzustellen, ob die Profile der Juroren die Beurteilung der Zusammenfassungen ausschlaggebend beeinflussen, oder ob die Qualität der Zusammenfassungen einen grösseren Beitrag für das Ergebnis darstellt.

4.3.3 Anpassungen vor der Evaluation

Vor der Durchführung der Evaluation sind folgende Schritte zu durchlaufen. An erster Stelle erfolgt der Schritt des Preprocessings. Da die vorgestellten Corpora aus 10000 Dokumenten bestehen, würde die Anzahl der Dimensionen aufgrund der Vektorkoordinatendarstellung der Worte zu hoch werden. Mit Hilfe des im vorherigen Kapitel vorgestellten Term Selection Tools werden Terme mit niedriger Termfrequenz aus der Wortliste entfernt. Weiters werden Preprocessing-Methoden wie Case-Folding, Stemming und das Entfernen von Stop-Worten angewendet. Für den Lyrics- und Banksearch-Corpus wurden noch zusätzliche Anpassungen gemacht:

Da der Lyrics Corpus aus Texten besteht, die kaum Satztrennzeichen, wie Punkt oder Beistrich, beihnhalten, wurde das Programm dementsprechend modifiziert, sodass jede Zeile im Text als ein Satz betrachtet wird. Der Grund für diese Anpassung liegt insbesondere darin, dass das Zusammenfassungssystem aufgrund der fehlenden Trennze-

ichen den gesamten Text in die Zusammenfassung aufnimmt, welches eine Verfälschung der Evaluierungsergebnisse nach sich zieht.

Auch die Banksearch Kollektion forderte leichte Veränderungen. Da viele Dokumente Statistiken beinhalten, wurde versucht in der Preprocessing-Phase durch Ausschluss von Zahlen und Elemente, wie Tags, Links oder Sonderzeichen, aus der Wortliste die Möglichkeit einer Bewertung von unbrauchbaren Informationen zu reduzieren.

Die automatischen Zusammenfassungen, die mit den manuell erstellten Zusammenfassung verglichen bzw. von den Juroren bewertet werden, werden vom eigenen Programm erstellt. Für die Grösse der Zusammenfassung wird 50% des Originaltextes gewählt. Weiters wird die "'Kombinierte Gewichtung"' als Extraktionsverfahren verwendet.

Die für die Evaluierung verwendeten Dokumente stammen aus Textclustern, die Eckpunkte, Spitze und Bereiche um die Spitze auf der SOM-Karte darstellen. Abbildung 4.1 illustriert das Selektieren der Cluster. Die mit "'X"' versehenen Bereiche entsprechen den Clustern, die für die Evaluierung hergenommen werden. Dabei beinhaltet ein Cluster ca. 20-30 Textdokumente. Je nach Auswahl der Corpora sind die Inhalte der Dokumente verschieden. Im Falle vom ACM-Corpus werden Themen aus dem Bereich "'IT"' behandelt. Der Banksearch-Corpus enthält Texte aus verschiedenen Themenbereichen (siehe Abschnitt 4.3.1). Das Lyrics Data Set ist eine Kollektion von Liedtexten aus mehreren Genres. Die folgenden drei Texte zeigen Inhalte aus den Dokumenten der drei Corpora. Das erste Dokument stammt aus dem ACM-Corpus, das zweite aus dem Banksearch-Corpus, und das dritte aus dem Lyrics-Corpus.

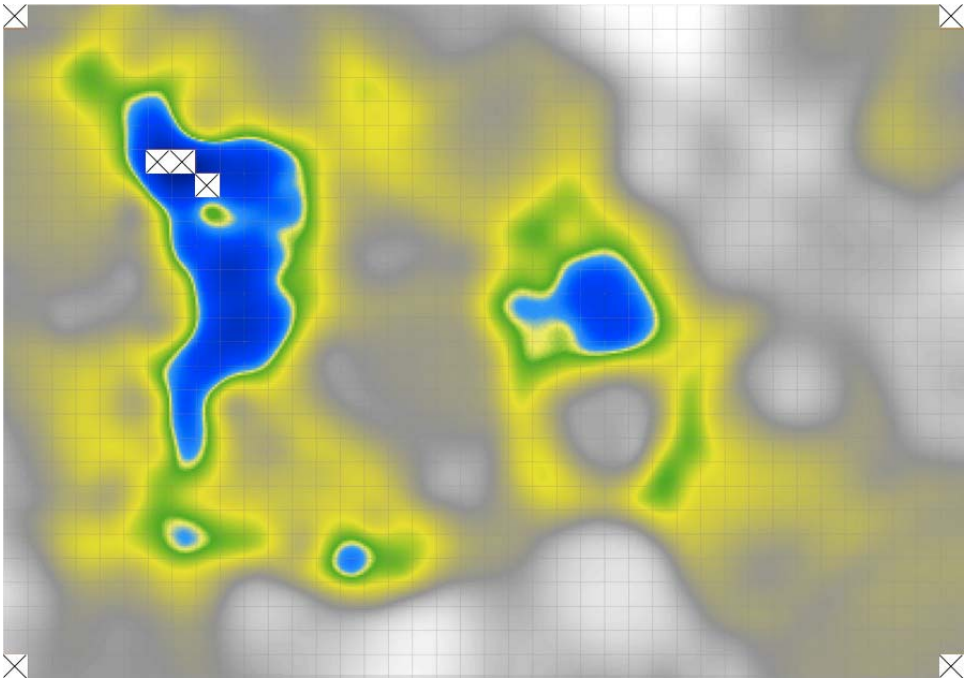


Figure 4.1: SOM Karte: ausgewählte Cluster

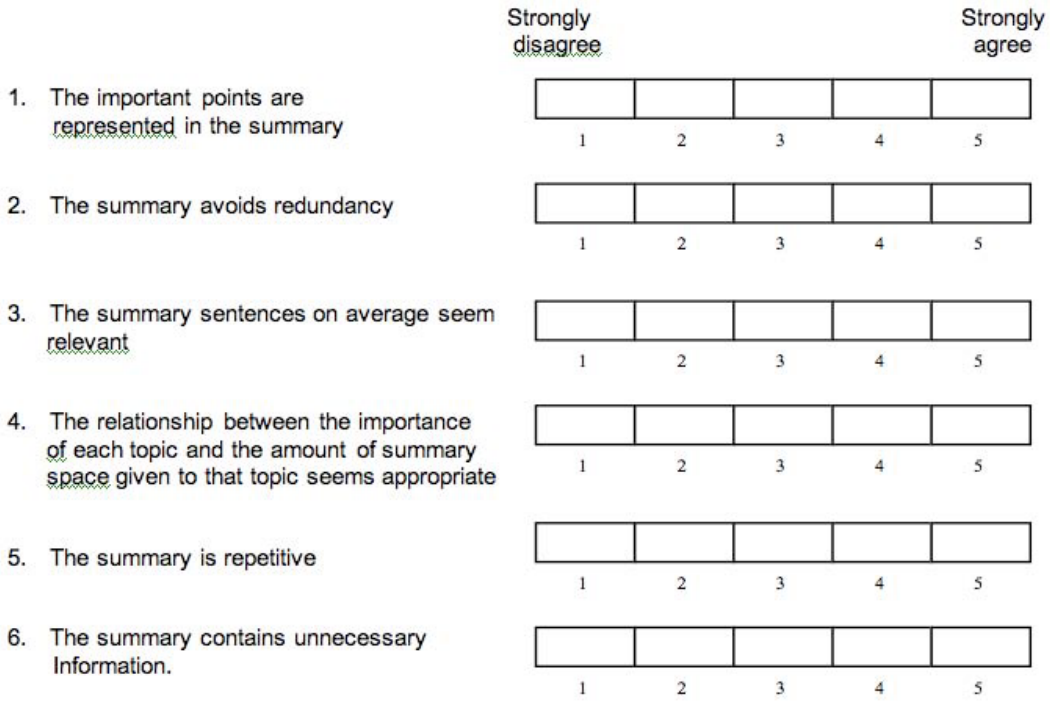
Solar Power May Soon Bring the Web to Remote Areas Christian Science Monitor (08/31/06) P. 15; Islam, Ranty [1] While the lofty promise of the One Laptop Per Child (OLPC) initiative to build and distribute hundreds of millions of inexpensive laptops for the world's poorest children has garnered headlines, it remained unclear how the devices would connect to the Internet in some of the remotest areas in the world. [2] But a new solar-powered wireless device could change that. [3] Many developing nations already have a thriving Internet presence, with their urban centers increasingly wired and Internet cafes springing up even in small towns. [4] But that accessibility fails to span the so-called "last mile," topping short of the millions of people who live in geographically removed villages and towns–the places where the Web might have its greatest impact. [5] The absence of reliable electricity had dimmed the prospect of building wireless networks, but the cofounders of the Green Wi-Fi project, partially funded by OLPC, have built the prototype of a solar-powered wireless router–essentially a commodity router hooked up to a battery that can be recharged by a solar panel. [6] Green Wi-Fi cofounders Marc Pomerleau and Bruce Baikie added an "intelligent charge controller" that governs the router's power consumption. [7] In preliminary testing, the wireless node appears to be able to run for four weeks even if the sky is overcast for sustained periods of time. [8] Theoretically, just one node connected to the Internet could provide Web access for a wireless network between villages. [9]To create a backbone network linking the hundreds of major access points across a region, engineers could use existing Wi-Fi, WiMax, or third generation mobile network technologies. [10] "If I had to design a backbone network from scratch, I would use all three," said Daniel Aghion, executive director of the Wireless Internet Institute.

Standard Life Bank - FAQ About Us Mortgages Personal Savings Business Savings Insurance Online Banking Contact Us Rates Brochure Library Added Extras Site Map
Search
HomeOnline BankingOnline Banking HelpFAQ Account List - How do I open a new account? - How do I close one of my accounts? - What if my personal details are incorrect? - What if I can't bring up details of an account? - Why aren't all of my funds available for withdrawal? Deposits - My external bank details have changed. Who do I contact? - Why can't I cancel a scheduled deposit? - Why does the date I have entered change? Internal Transfers - Why can't I see the account I want to transfer funds into? - Why won't it accept the account number I've entered? - Why does the date I have entered change? Withdrawals - My external bank details have changed. Who do I contact? - Why can't I cancel a scheduled withdrawal? - Why does the date I have entered change? Account Statements - Why doesn't anything happen when I press the Withdrawals button? - Why doesn't anything happen when I press the Deposits button? - Why doesn't anything happen when I press Internal Transfers button? - What if my designated external details are wrong? - Why can't I see details of one of my transactions? - What if one of my transactions is incorrect? Future Transactions Screen - How do I change the amount or date of a scheduled deposit, withdrawal or transfer? Hardware and Software - How can I get access to the online banking service? - What equipment do I need to be able to use the online banking service? - Why do I need to use Internet Explorer v 4.01 or later or Netscape navigator v 4.06 or later? - What if my computer internet session ends unexpectedly? - Is the online banking service available to Apple Macintosh / Windows 3.1 / Unix / OS/2 users? - Who do I contact if I still have a technical query? Security - Is the internet safe and secure for online banking? - Can anyone else see my account information? How do you keep my banking details private and secure? - Can more than one person use the same PC and still have their details kept private? - What encryption are you using? - I have entered my password incorrectly three times and my access has now been revoked. How can I be reinstated? - I have forgotten my Internet UserID and password. What should I do? - What can I do to make my Online banking 'safer'? - Can Standard Life Bank's online service operate alongside my company's firewall? - What measures are being taken to ensure that no viruses are passed over the Internet? - How secure is the on-line banking logon?
Account List - How do I open a new account? If you're new to Standard Life Bank, all you have to do is apply online. You can also call us on 08457 555657 for personal savings, 08457 555659 for business savings and 0845 8458450 for mortgages. Back to top
- How do I close one of my accounts? Call us on 0845 609 0256 and we'll close the account for you. Back to top
- What if my personal details are incorrect? You can change some of your personal details online. To register for online banking click on 'register' at the top of the screen. Alternatively, you can call us on 0845 609 0256 for personal savings, 0845 609 0257 for business savings and 0845 609 0262 for mortgages. Back to top
- What if I can't bring up details of an account? Only active accounts appear on the Account List. If you have an account which you think should be displayed, but isn't, call us on 0845 609 0256 for personal savings, 0845 609 0257 for business savings and 0845 609 0262 for mortgages. Back to top
- Why aren't all of my funds available for withdrawal? Only cleared funds can be withdrawn. It takes 8 working days for a cheque to clear from the date of receipt. Deposits by Direct Debit take 5 working days to clear. The Account Statement screen shows deposits made but not yet cleared. Back to top
Deposits - My external bank details have changed. Who do I contact? Call us on 08457 555657 for personal savings and 08457 555659 for business savings and we'll make the changes for you. Back to top
- Why can't I cancel a scheduled deposit? You can't cancel a scheduled deposit within 3 days of the effective date. We allow this time for your money to pass through our payments system. Back to top
- Why does the date I have entered change? You can't make a deposit on a bank holiday or at the weekend so our system recommends the next most suitable date. You can change this date if you want. Back to top
Internal Transfers - Why can't I see the account I want to transfer funds into? You can only transfer funds into active accounts in the same plan. If you're not the holder of an account you want to transfer funds into, you must input the account number manually. Back to top
- Why won't it accept the account number I've entered? You can't transfer funds online into bond or ISA accounts. You can only transfer funds into active accounts in the same plan. If you think you have a valid account number and you're having problems call us on 0845 609 0256. Back to top
- Why does the date I have entered change? You can't make a deposit on a bank holiday or at the weekend so our system recommends the next most suitable date. You can change this date if you want. Back to top
Withdrawals - My external bank details have changed. Who do I contact? Call us on 08457 555657 for personal savings and 08457 555659 for business savings and we'll make the changes. Back to top
- Why can't I cancel a scheduled withdrawal? You can't cancel a scheduled withdrawal within 3 days of the effective date. We allow this time for your money to pass through our payments system. Your money will still earn interest right up to the day it's transferred to your external account. Back to top
- Why does the date I have entered change? You can't make a deposit on a bank holiday or at the weekend so our system recommends the next most suitable date. You can change this date if you want. Back to top
Account Statement - Why doesn't anything happen when I press the Withdrawals button? Have you selected a bond or ISA account where no withdrawals can be made? Have you yet to return the letter accepting our Terms & Conditions? Perhaps we haven't got your Personal Identification documentation? Call us on 0845 609 0256 for personal savings, 0845 609 0257 for business savings and 0845 609 0262 for mortgages and we'll try our best to help. Back to top
- Why doesn't anything happen when I press the Deposits button? Have you selected a bond or ISA account where you can't make any further deposits? Have you yet to return your Direct Debit Mandate or the letter accepting our Terms & Conditions? Perhaps we haven't got your Personal Identification documentation? Call us on 0845 609 0256 for personal savings, 0845 609 0257 for business savings and 0845 609 0262 for mortgages and we'll try our best to help. Back to top
- Why doesn't anything happen when I press Internal Transfers button? Have you selected a bond or ISA account where no withdrawals or deposits can be made? Have you yet to return the letter accepting our Terms & Conditions? Perhaps we haven't got your Personal Identification documentation? Call us on 0845 609 0256 for personal savings, 0845 609 0257 for business savings and 0845 609 0262 for mortgages and we'll try our best to help. Back to top
- What if my designated external details are wrong? Call us on 08457 555657 for personal savings, 08457 555659 for business savings and 0845 8458450 for mortgages and we'll change them for you. Back to top
- Why can't I see details of one of my transactions? If a transaction appears to be missing, call us on 0845 609 0256 for personal savings, 0845 609 0257 for business savings and 0845 609 0262 for mortgages and we'll investigate it for you. Back to top

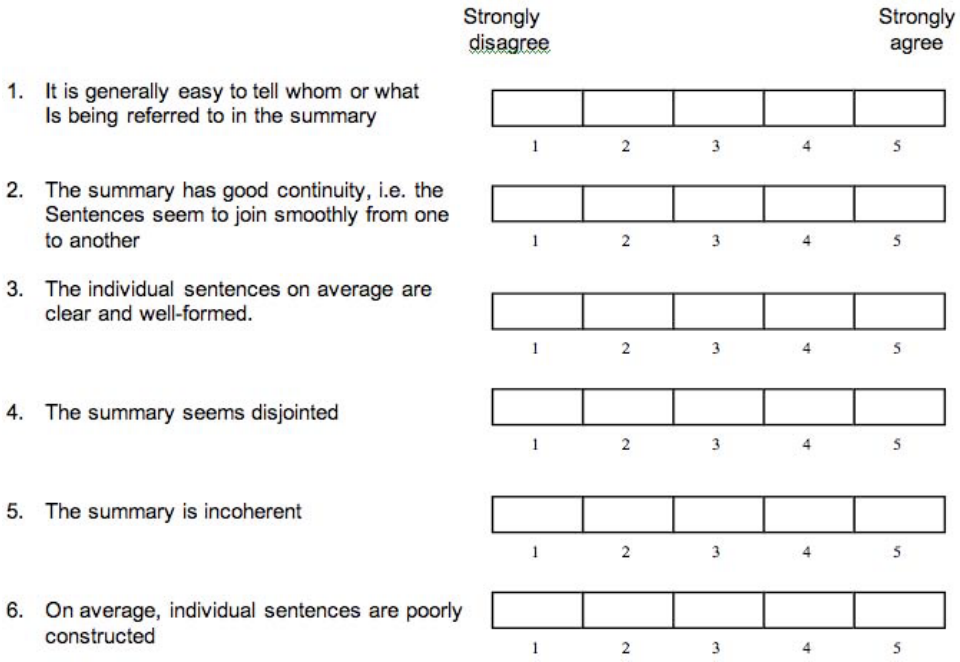
TLC - The Hits
yes it's me again
and i'm back
oh i oh i oh i yeah
oh i oh i oh i baby
oh i oh i oh i yeah
oh i oh i oh i baby
the 22nd lonliness and we've
been through so many thangs
i love my man all honesty
but i know he's cheating me
i look him his eyes but all he
tells me lies keep me near
i'll never leave him down though
i might mess around it's only
'cause i need some affection oh
chorus
so i creep yeah
just keep down low
said no one else supposed know
so i creep yeah
'cause he doesn't know
what i do and no attention
goes show oh so i creep
the 23rd lonliness
and we don't talk
like we used do
now seems pretty
strange but i'm not
buggin' 'cause i still feel
the same yeah yeah
i'll keep loving
till day he pushes me away
never go astray
if he new things i did he couldn't
handle it
and i choose keep him protected
chorus
i think us baby all time
but you know
that i'm gonna need some atention
yeah yeah can you dig it
love you forever baby soul and mind
and you gotta know if
you don't give i'ma
get mine
oh i oh i oh i yeah
oh i oh i oh i baby
oh i oh i oh i yeah
yeah yeah yeah
chorus
i creep around because i need some attention
don't mess around my affection
oh i oh i oh i yeah
chorus yes it's me again
and i'm back
oh i oh i oh i yeah
oh i oh i oh i baby
oh i oh i oh i yeah
oh i oh i oh i baby

4.3.4 Subjektive Evaluierung

Alle fünf Juroren bekommen die gleichen, vom System erstellten Zusammenfassungen, wobei keiner von ihnen mit dem Inhalt der Texte zuvor vertraut ist. In Anlehnung an die im [Mur05] vorgestellte, subjektive Evaluierung werden alle Testpersonen mit 12 Fragen konfrontiert, die, ohne dass die Teilnehmer erneut die Zusammenfassung lesen, am Ende jeder Zusammenfassung beantworten werden. Sechs dieser Fragen beziehen sich auf die Aussagefähigkeit der Zusammenfassungen, die anderen sechs auf die Lesbarkeit und Kohärenz. Die Evaluierung verwendet eine fünffach gestufte Likert Skala basierend auf agreement or disagreement mit folgenden Aussagen über die Aussagefähigkeit (oder *informativeness*):



Der andere Teil über die Lesbarkeit (*readability*) und Kohärenz (*coherence*)besteht aus folgenden Aussagen:



Ergebnisse der subjektiven Evaluierung

Nachdem die Teilnehmer die Zusammenfassungen gelesen und die Fragen beantwortet haben, werden alle Ergebnisse durch Ermittlung der Mittelwerte und Standardabweichungen ausgewertet und in eine Tabelle eingetragen. Die folgenden Tabellen 4.1, 4.2 und 4.3 zeigen die Ergebnisse aller drei Corpora. Im linken Teil der Tabellen sind die 12 Fragen angeführt.

	Juror 1		Juror 2		Juror 3		Juror 4		Juror 5	
	Mean	Std.Dev	Mean	Std.Dev	Mean	Std.Dev	Mean	Std.Dev	Mean	Std.Dev
Qi1	4,427	0,632	4,339	0,676	4,301	0,651	4,388	0,642	3,602	0,863
Qi2	4,854	0,352	4,738	0,439	4,476	0,518	4,214	0,717	3,825	0,715
Qi3	3,924	0,605	4,447	0,664	4,359	0,651	4,534	0,604	3,689	0,849
Qi4	4,146	0,659	4,184	0,679	4,117	0,643	4,126	0,667	3,748	0,867
Qi5	2,049	0,215	1,214	0,409	1,398	0,509	1,679	0,544	2,009	0,764
Qi6	2,244	0,911	1,718	0,918	1,699	0,811	2,049	0,826	2,127	0,797
Qrc1	4,699	0,479	4,515	0,605	4,495	0,605	4,427	0,568	3,388	0,686
Qrc2	3,952	0,734	4,194	0,738	4,068	0,727	4,184	0,693	2,971	0,782
Qrc3	3,602	0,546	4,388	0,578	4,495	0,572	4,339	0,584	3,485	0,787
Qrc4	1,446	0,772	1,631	0,800	1,650	0,844	1,757	0,689	2,379	0,886
Qrc5	1,718	0,829	1,825	0,756	1,883	0,792	1,854	0,756	2,272	0,883
Qrc6	2,165	0,464	1,495	0,621	1,447	0,619	1,718	0,565	2,320	0,791

Table 4.1: ACM-Corpus: subjektive Evaluierung

	Juror 1		Juror 2		Juror 3		Juror 4		Juror 5	
	Mean	Std.Dev	Mean	Std.Dev	Mean	Std.Dev	Mean	Std.Dev	Mean	Std.Dev
Qi1	2,232	0,773	3,568	0,807	3,581	0,678	3,689	0,821	3,068	0,777
Qi2	2,773	0,682	3,527	0,774	3,635	0,746	3,554	0,824	3,135	0,797
Qi3	1,849	0,796	3,446	0,756	3,689	0,734	3,554	0,824	2,081	0,748
Qi4	1,311	0,837	3,054	0,695	3,311	0,914	2,797	0,877	2,797	0,874
Qi5	2,851	0,671	2,297	0,766	2,405	0,804	2,432	0,522	2,432	0,754
Qi6	4,195	0,957	3,405	0,971	3,392	0,835	3,905	1,115	3,905	0,826
Qrc1	2,838	0,884	3,284	0,847	3,257	0,678	2,297	0,668	1,986	0,779
Qrc2	1,608	0,589	2,743	0,699	2,959	0,703	1,986	0,688	1,500	0,663
Qrc3	2,365	0,815	3,392	0,867	3,014	0,707	2,108	0,831	2,973	0,854
Qrc4	3,770	0,794	3,081	0,673	3,054	0,733	3,462	0,658	3,724	0,790
Qrc5	3,570	0,741	3,203	0,735	3,041	0,687	3,149	0,671	3,743	0,749
Qrc6	4,027	1,551	3,514	0,874	3,297	0,801	3,351	0,743	3,662	0,874

Table 4.2: Banksearch-Corpus: subjektive Evaluierung

	Juror 1		Juror 2		Juror 3		Juror 4		Juror 5	
	Mean	Std.Dev	Mean	Std.Dev	Mean	Std.Dev	Mean	Std.Dev	Mean	Std.Dev
Qi1	3,564	0,590	3,295	0,736	3,603	0,686	3,653	0,617	2,987	0,776
Qi2	3,192	0,621	3,397	0,713	3,397	0,667	3,231	0,553	2,987	0,742
Qi3	3,513	0,635	3,474	0,763	3,603	0,562	3,141	0,614	2,936	0,722
Qi4	2,462	0,634	2,410	0,629	2,718	0,677	2,500	0,655	2,244	0,429
Qi5	3,410	0,815	3,026	0,679	3,103	0,841	3,077	0,703	3,564	0,652
Qi6	3,397	0,765	3,179	0,655	3,167	0,912	3,244	0,804	3,936	0,677
Qrc1	2,415	0,487	3,436	0,545	3,795	0,585	3,679	0,650	3,077	0,694
Qrc2	2,482	0,597	3,154	0,533	3,192	0,680	3,064	0,667	2,059	0,674
Qrc3	2,389	0,719	2,808	0,680	3,077	0,635	3,103	0,794	2,021	0,675
Qrc4	2,872	0,667	3,000	0,599	3,115	0,640	2,974	0,816	3,214	0,693
Qrc5	3,026	0,760	3,128	0,723	3,205	0,774	2,897	0,727	3,469	0,673
Qrc6	3,549	0,732	3,577	0,631	3,423	0,743	2,936	0,757	3,821	0,780

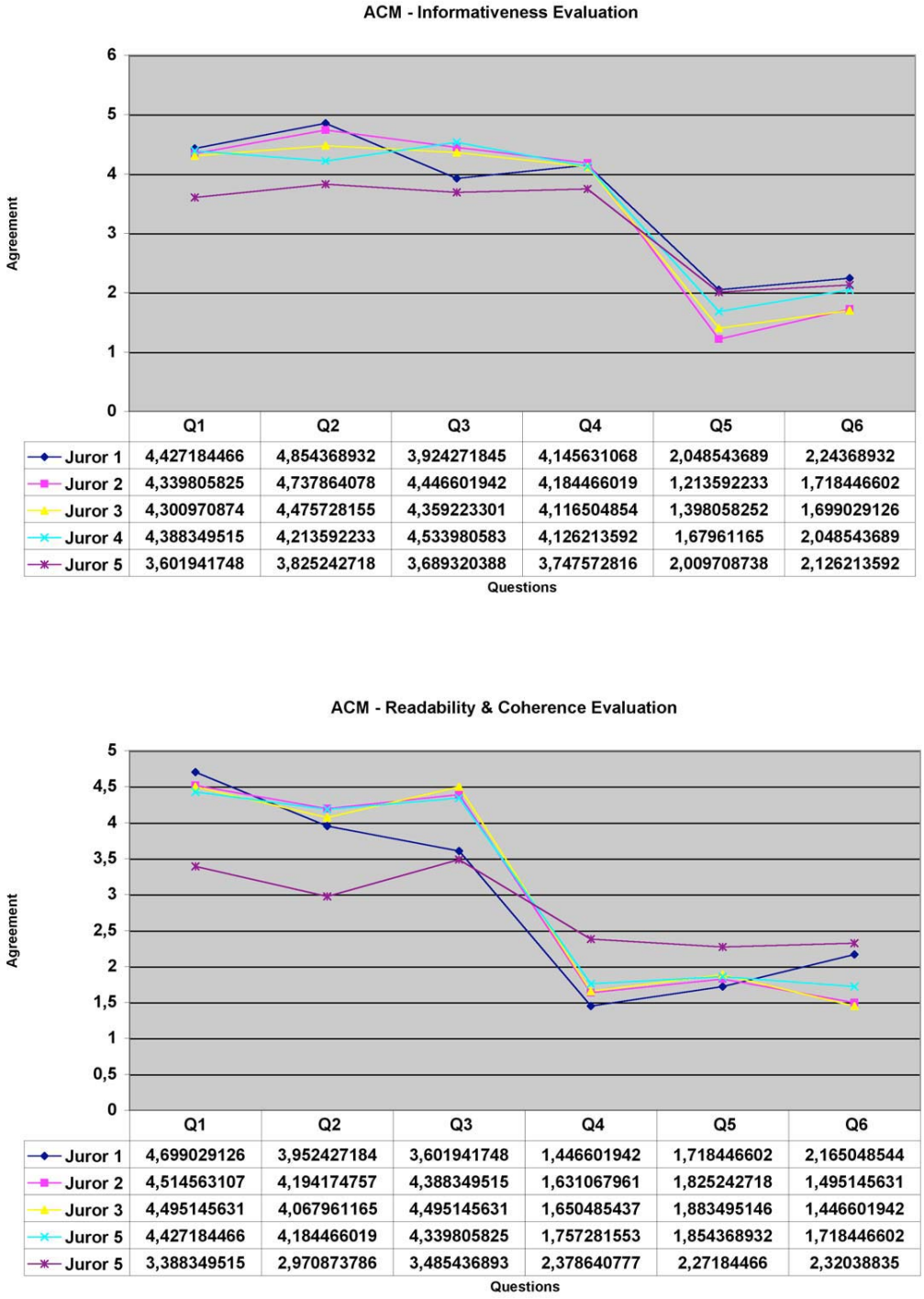
Table 4.3: Lyrics-Corpus: subjektive Evaluierung

Stellt man nun diese Mittelwerte grafisch dar, so erhält man die Abbildung 4.2, 4.3 und 4.4.

Es hat sich herausgestellt, dass der ACM Corpus die besten Ergebnisse erzielte. Die Mehrheit der Juroren haben fast alle automatisch erstellten Zusammenfassungen, die aus dem ACM Corpus stammen, gut bewertet. Obwohl sich einige Teilnehmer im Bereich Computer als Anfänger einstufen, sind sie der Ansicht, dass die Zusammenfassungen überwiegend verständlich und informativ seien.

Der Banksearch Corpus hingegen erzielte die schlechtesten Ergebnisse. Das Diagramm über die Aussagekraft (Abbildung 4.6) zeigt, dass viele Juroren der Meinung sind, dass die Zusammenfassungen unnötige Informationen enthalten. Einige merken an, dass die Sätze in den Zusammenfassungen überwiegend inkohärent und schwach strukturiert sind. Ein möglicher Grund für die schlechten Ergebnisse liegt in der mangelnden Struktur der Textdokumente. Da der Corpus ohne jegliche Formatierung direkt aus dem Internet übernommen wurde, und die manuelle Bearbeitung von 10000 Dokumenten den Rahmen dieser Arbeit sprengen würde, ist es für das Programm schwierig sinnvolle Sätze zu extrahieren und zu bewerten. Im Abschnitt 4.3.3 wurde versucht dieses Problem durch einige Anpassungen zu lösen.

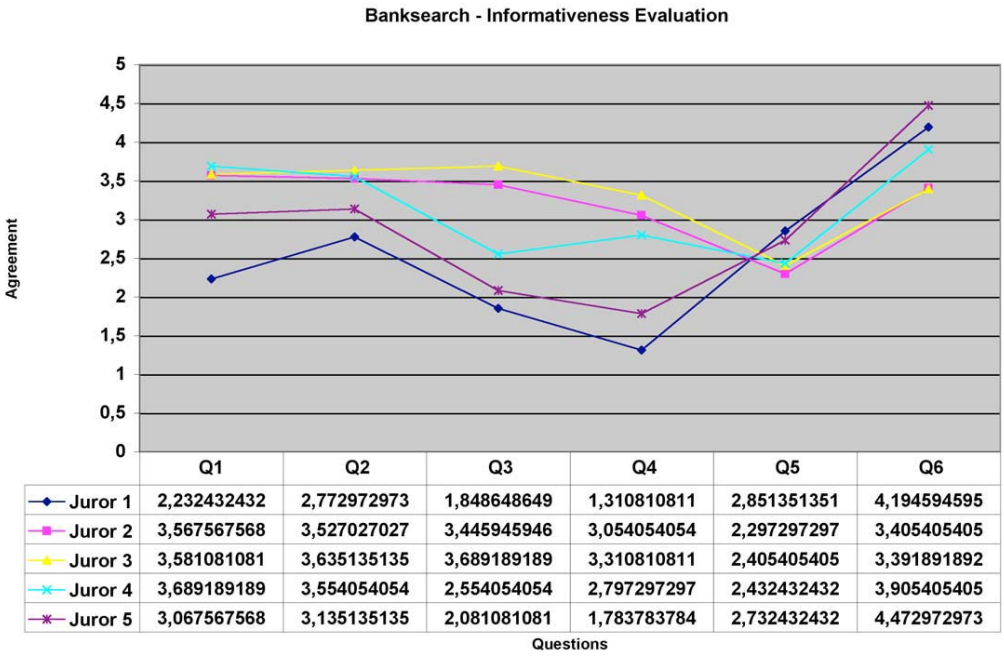
Die Zusammenfassungen, die aus dem Lyrics Corpus stammen, wurden von den Teilnehmern eher neutral bewertet.



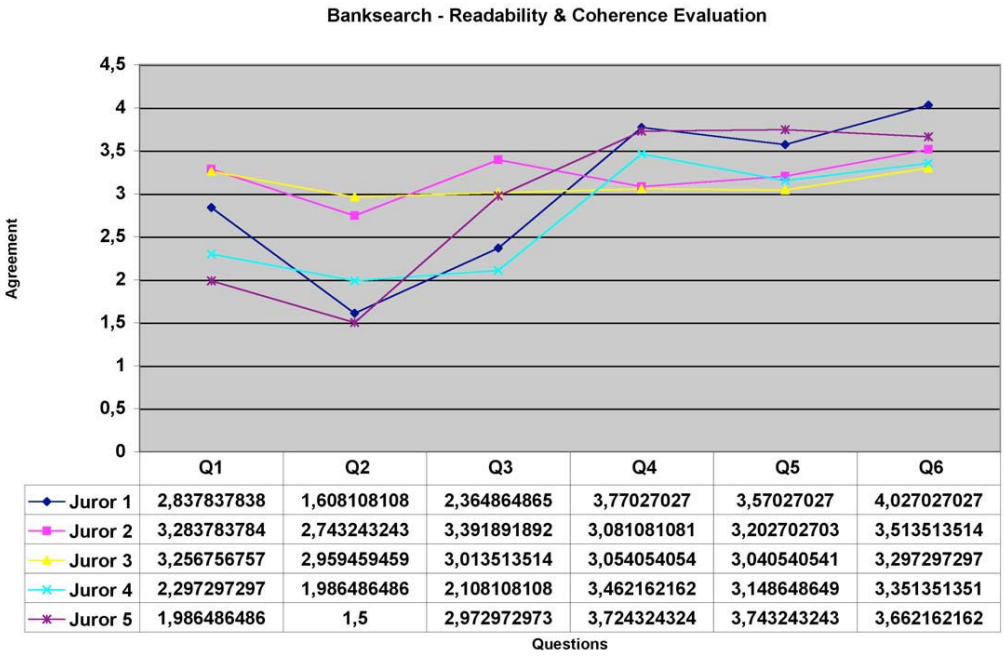
	Q1	Q2	Q3	Q4	Q5	Q6
Juror 1	4,427184466	4,854368932	3,924271845	4,145631068	2,048543689	2,24368932
Juror 2	4,339805825	4,737864078	4,446601942	4,184466019	1,213592233	1,718446602
Juror 3	4,300970874	4,475728155	4,359223301	4,116504854	1,398058252	1,699029126
Juror 4	4,388349515	4,213592233	4,533980583	4,126213592	1,67961165	2,048543689
Juror 5	3,601941748	3,825242718	3,689320388	3,747572816	2,009708738	2,126213592

	Q1	Q2	Q3	Q4	Q5	Q6
Juror 1	4,699029126	3,952427184	3,601941748	1,446601942	1,718446602	2,165048544
Juror 2	4,514563107	4,194174757	4,388349515	1,631067961	1,825242718	1,495145631
Juror 3	4,495145631	4,067961165	4,495145631	1,650485437	1,883495146	1,446601942
Juror 5	4,427184466	4,184466019	4,339805825	1,757281553	1,854368932	1,718446602
Juror 5	3,388349515	2,970873786	3,485436893	2,378640777	2,27184466	2,32038835

Figure 4.2: ACM Corpus - subjektive Evaluierung

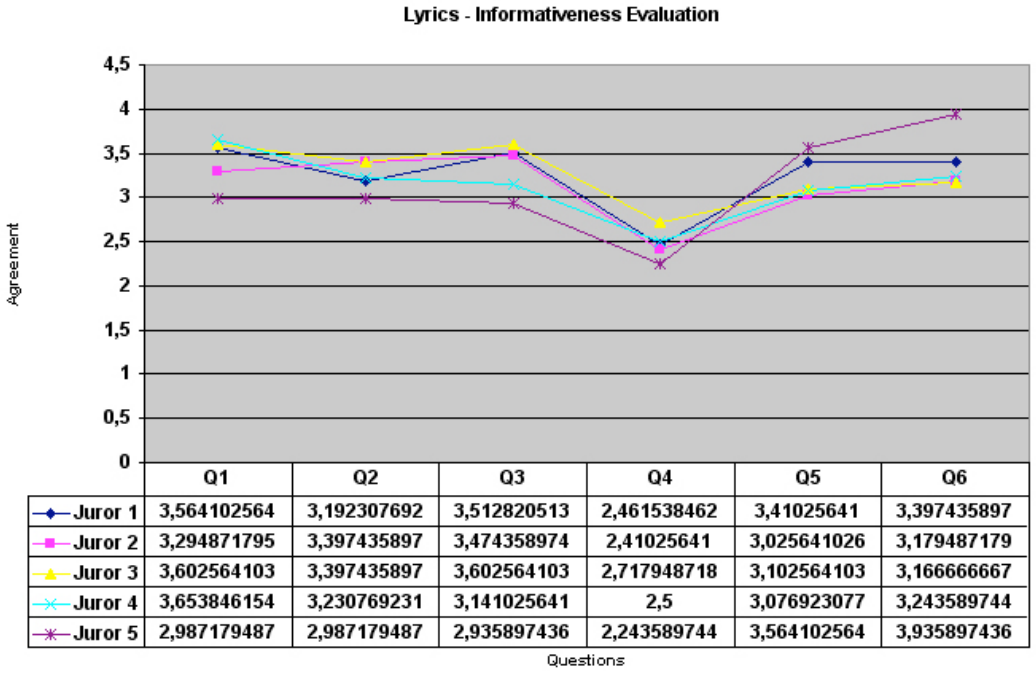


	Q1	Q2	Q3	Q4	Q5	Q6
Juror 1	2,232432432	2,772972973	1,848648649	1,310810811	2,851351351	4,194594595
Juror 2	3,567567568	3,527027027	3,445945946	3,054054054	2,297297297	3,405405405
Juror 3	3,581081081	3,635135135	3,689189189	3,310810811	2,405405405	3,391891892
Juror 4	3,689189189	3,554054054	2,554054054	2,797297297	2,432432432	3,905405405
Juror 5	3,067567568	3,135135135	2,081081081	1,783783784	2,732432432	4,472972973

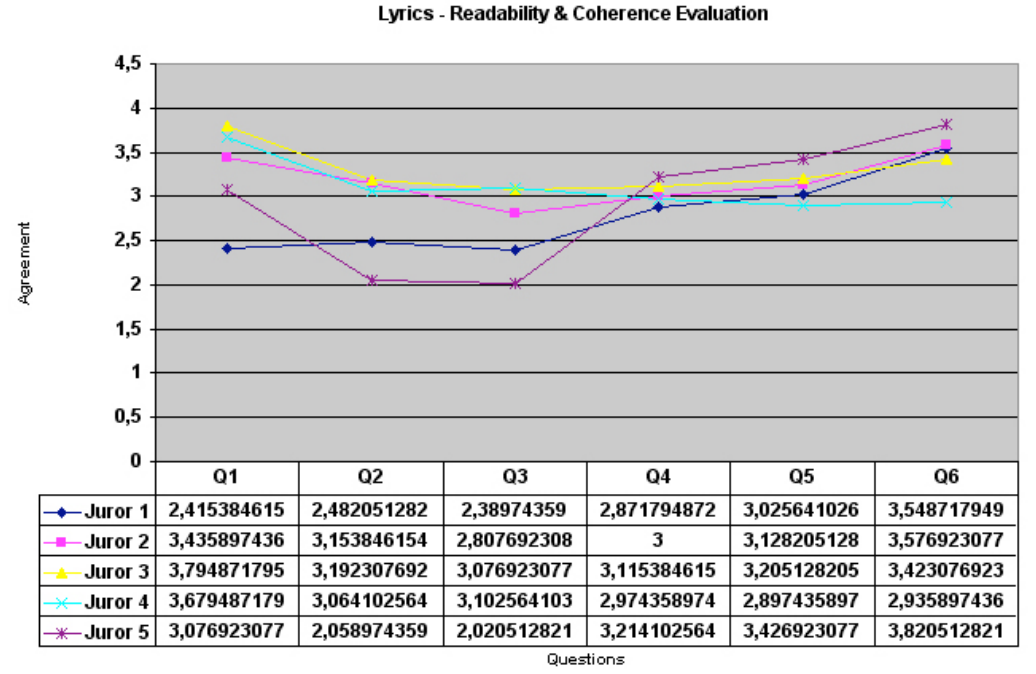


	Q1	Q2	Q3	Q4	Q5	Q6
Juror 1	2,837837838	1,608108108	2,364864865	3,77027027	3,57027027	4,027027027
Juror 2	3,283783784	2,743243243	3,391891892	3,081081081	3,202702703	3,513513514
Juror 3	3,256756757	2,959459459	3,013513514	3,054054054	3,040540541	3,297297297
Juror 4	2,297297297	1,986486486	2,108108108	3,462162162	3,148648649	3,351351351
Juror 5	1,986486486	1,5	2,972972973	3,724324324	3,743243243	3,662162162

Figure 4.3: Banksearch Corpus - subjektive Evaluierung



	Q1	Q2	Q3	Q4	Q5	Q6
Juror 1	3,564102564	3,192307692	3,512820513	2,461538462	3,41025641	3,397435897
Juror 2	3,294871795	3,397435897	3,474358974	2,41025641	3,025641026	3,179487179
Juror 3	3,602564103	3,397435897	3,602564103	2,717948718	3,102564103	3,166666667
Juror 4	3,653846154	3,230769231	3,141025641	2,5	3,076923077	3,243589744
Juror 5	2,987179487	2,987179487	2,935897436	2,243589744	3,564102564	3,935897436



	Q1	Q2	Q3	Q4	Q5	Q6
Juror 1	2,415384615	2,482051282	2,38974359	2,871794872	3,025641026	3,548717949
Juror 2	3,435897436	3,153846154	2,807692308	3	3,128205128	3,576923077
Juror 3	3,794871795	3,192307692	3,076923077	3,115384615	3,205128205	3,423076923
Juror 4	3,679487179	3,064102564	3,102564103	2,974358974	2,897435897	2,935897436
Juror 5	3,076923077	2,058974359	2,020512821	3,214102564	3,426923077	3,820512821

Figure 4.4: Lyrics Corpus - subjektive Evaluierung

Vergleich mit manuell verfassten Zusammenfassungen

In einem weiteren Schritt der subjektiven Evaluation will man nun erfahren, ob die automatisch erstellten Zusammenfassungen besser sind als die menschlich verfassten Zusammenfassungen. Dazu wird jeder Juror beauftragt, eine eigene, für sich ideale Zusammenfassung, nach dem Originaltext gerichtet, zu erstellen. Es ist anzumerken, dass hierbei nur Dokumente aus dem ACM-Corpus hergenommen werden, da im Vornherein klar ist, dass bei Verwendung des Banksearch- bzw. Lyrics-Corpus die manuell verfasste Zusammenfassung deutlich besser ist. Weiters werden zehn Dokumente, die in der Beurteilung der Juroren sehr unterschiedlich sind, aus einem der für die Evaluierung verwendeten Clustern (siehe Abschnitt 4.3.3), in diesem Fall der linke obere Eckpunkt, genommen.

Nachdem die Teilnehmer ihre eigenen Zusammenfassungen erstellt haben, wurden diese von den anderen Juroren bewertet. Jeder bekam somit für ein Dokument insgesamt fünf Zusammenfassungen, einschliesslich jener vom System. Die Zusammenfassungen sollen auf einer Skala von 1 bis 5 benotet werden, wobei 1 als "'sehr gut"' gilt. Errechnet man nun den Durchschnitt aller Benotungen jedes einzelnen Teilnehmers, so erhält man in Tabelle 4.4 folgende Durchschnittswerte und deren Platzierungen:

Judges	System	Juror 1	Juror 2	Juror 3	Juror 4	Juror 5
Juror 1	3,0 (2)		2,1 (1)	3,1 (3)	3,2 (4)	3,8 (5)
Juror 2	4,0 (5)	2,6 (2)		2,0 (1)	2,7 (3)	3,4 (4)
Juror 3	3,1 (3)	3,2 (4)	3,3 (5)		2,8 (1)	2,9 (2)
Juror 4	3,9 (5)	3,0 (3)	2,8 (2)	2,1 (1)		3,2 (4)
Juror 5	4,0 (5)	3,2 (3)	1,8 (1)	2,5 (2)	3,5 (4)	
Overall	3,6 (6)	2,975 (3)	2,45 (2)	2,425 (1)	3,05 (4)	3,325 (5)

Table 4.4: Ergebnisse der Benotung

Wie man aus der Tabelle 4.4 herauslesen kann, wurden die automatisch erstellten Zusammenfassungen von Juror 1 und Juror 3 gut bewertet. Alle übrigen Teilnehmer benoteten die Zusammenfassungen eher schlecht. Insgesamt wurden die Zusammenfassungen von Juror 3 am besten benotet.

4.3.5 Objektive Evaluierung

Für die Erstellung der "'idealen"' Zusammenfassung[3], welche als Vergleichsobjekte für die automatisch erstellten Zusammenfassungen dienen, werden dieselben Juroren eingesetzt. Zur objektiven Bewertung der Ähnlichkeit zwischen den beiden Arten der Zusammenfassungen wird hierfür die Precision&Recall-Methode verwendet.

Precision&Recall und F-Measure

Precision und Recall sind zwei Maße zur Beschreibung der Güte eines Suchergebnisses beim Information Retrieval oder für die Evaluierung eines Information-Retrieval-Systems. Die Precision beschreibt dabei die Genauigkeit eines Ergebnisses. In diesem Fall gibt sie also den Anteil der relevanten Sätze unter den für die Zusammenfassung ausgewählten Sätze an. Der Recall beschreibt die Vollständigkeit eines Ergebnisses. Hierbei wird gemessen wie viele Sätze aus der "'idealen"' Zusammenfassung auch in der automatisch generierten Zusammenfassung enthalten sind.

Das F-Measure (oder F-Maß) ist ein Evaluationsmaß, welches Precision und Recall gleichgewichtet kombiniert.

Im Folgenden bezeichne *REL* die relevanten Sätze aus der idealen Zusammenfassung und *GEF* die Menge aller Sätze in der automatisch erstellten Zusammenfassung, so errechnet man die drei Maßzahlen durch folgende Formeln:

$$Pr = \frac{REL \cap GEF}{GEF} \tag{4.1}$$

$$Re = \frac{REL \cap GEF}{REL} \tag{4.2}$$

$$F = \frac{2 * Pr * Re}{Pr + Re} \tag{4.3}$$

[3]Um Missverständnisse zu vermeiden, soll an dieser Stelle angemerkt werden, dass man hierbei unter einer "'idealen"' Zusammenfassung einen vom Menschen erstellten Extrakt versteht. Es handelt sich also nicht, wie im vorherigen Kapitel, um eine eigene verfasste Zusammenfassung.

Bei vielen Evaluierungen von Retrievalexperimenten wird oft ein Precision&Recall-Graph verwendet. In diesem Graph wird auf der x-Achse der Recall und auf der y-Achse die Precision aufgetragen und so versucht ein Bewertungsmass zu schaffen, dass beide Grössen miteinbezieht. Die Abbildung 4.5 zeigt einen solchen Graphen. Dabei können folgende Extremfälle auftreten:

1. Ist der Precision-Wert 1, der Recall-Wert jedoch 0, so würde die automatisch erstellte Zusammenfassung zwar relevante Sätze enthalten, es würden aber auch viele Sätze fehlen, die wichtig sind.

2. Ist der Recall-Wert 1 und der Precision-Wert 0, so enthält die Zusammenfassung zwar die meisten, für den Nutzer relevanten Sätze, sie würde aber auch von vielen unwichtigen Informationen behaftet sein.

Ideal ist es also, wenn beide Größen gegen 1 streben. Die Abbildung zeigt den typischen Verlauf eines solchen Precision&Recall-Graphen beim Information Retrieval. Es wird jedoch eine andere Methode der Gegenüberstellung von Precision und Recall hergenommen, da man sich speziell dafür interessiert bei welchen Juroren das Zusammenfassungssystem gut bzw. schlecht abschneidet.

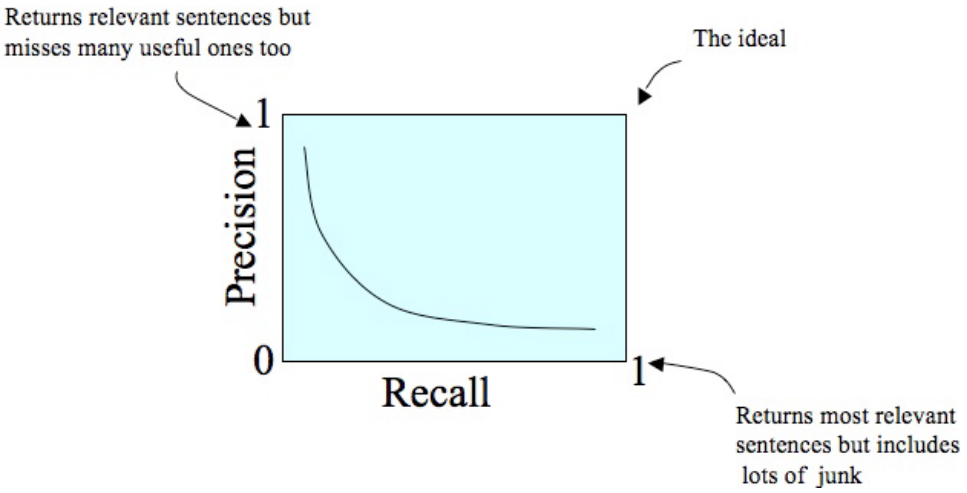


Figure 4.5: Precision&Recall - Graph [Moo06]

Nachdem man die vom System erstellten Zusammenfassungen mit den manuell erstellten Extrakten der Juroren verglichen hat, werden Precision-, Recall- und F-Maßwerte errechnet. Als Berechnungsbasis werden die Sätze der zu vergleichenden Zusammenfassungen festgelegt.

Das folgende Beispiel zeigt, wie Precision- und Recall-Werte anhand des unteren Textes errechnet werden:

Würde das System die Sätze 1, 5, 6, 8, 9 und 12 aus dem Text auswählen, wohingegen eine manuell erstellte Zusammenfassung die Sätze 1, 2, 4, 6, 9 beinhaltet, so ist der Precision-Wert $\frac{3}{6} = 0,5$ laut Formel 4.1 und der Recall-Wert $\frac{3}{5} = 0,6$ laut Formel 4.2. Das F-Maß beträgt somit $0,54$.

[1] Yale University computer scientist and veteran developer David Gelernter says he is now focusing on creating tools that make it easier for users to find "stuff" on their computers and otherwise improve the end user's computer experience.
[2] Gelernter says the mouse, icon, and windows metaphors are no longer able to manage the flood of information on most people's PCs.
[3] He says, "As email and the Web became a big thing, it was clear that the hierarchical file systems and tools we've inherited from the 70s would not work."
[4] To solve this problem, his company, Mirror Worlds Technologies, has released a beta version of Scopeware, software that runs stop normal desktop operating systems.
[5] Scopeware, available free via download, allows users to search for standard documents on their PC by keyword, but presents the results as a visual, time-sequence narrative.
[6] Gelernter says the user should determine the presentation of information, not the machine.
[7] "I want my information management software to have the same shape as my life, which is a series of events in time," he says.
[8] "I want the flow to determine the shape of the picture I see on the screen."
[9] Gelernter says that future iterations of Scopeware could allow a community of users to share documents pertinent to them through peer-to-peer systems.
[10] Gelernter was instrumental in devising the parallel programming techniques that allowed for the Linda language;
[11] his work also laid the foundation for Java and distributed memory architectures.
[12] He says it's now time to create software "for the user as an everyday tool," not to meet the needs of code developers.

Bei der Erstellung der Zusammenfassungen gab es für die Juroren keine Längenvorgaben. Im Zuge dessen stellte sich heraus, dass alle manuell verfassten Zusammenfassungen kürzer sind, als jene vom System.

Nimmt man nun die Mittelwerte der drei genannten Maßzahlen, so erhält man folgende Tabelle 4.5. Die Werte im linken Teil beziehen sich auf den ACM-, im mittleren

Teil auf den Banksearch-, und im rechten Teil auf den Lyrics-Corpus. Wie im linken Teil der Tabelle abzulesen, erreicht das System bei Juror 3 und Juror 4 die besten Ergebnisse. Der Vergleich mit den übrigen Referenzzusammenfassungen erziehlt allerdings auch brauchbare Ergebnisse. Im mittleren Teil der Tabelle erkennt man, dass bei Juror 5 die Precision- und Recall-Werte besser sind, als bei den anderen Juroren. Der rechte Teil der Tabelle zeigt wiederum, dass das System bei Juror 3 ein gutes Ergebnis liefert.

	ACM			Banksearch			Lyrics		
Juroren	Pr	R	F-Maß	Pr	R	F-Maß	Pr	R	F-Maß
Juror 1	0,58	0,63	0,59	0,26	0,86	0,32	0,31	0,58	0,38
Juror 2	0,52	0,68	0,58	0,15	0,88	0,24	0,34	0,62	0,42
Juror 3	0,55	0,69	0,61	0,25	0,84	0,33	0,50	0,61	0,52
Juror 4	0,53	0,71	0,61	0,24	0,86	0,34	0,39	0,59	0,46
Juror 5	0,53	0,67	0,58	0,28	0,90	0,40	0,37	0,69	0,44

Table 4.5: Precision, Recall und F-Maß

Die Abbildungen 4.6, 4.7 und 4.8 stellen nochmals Precision und Recall, mit ihren Maximal- und Minimalwerten, grafisch gegenüber. Dabei werden auf der x-Achse die Juroren, von denen die für die Evaluierung verwendeten Referenzzusammenfassungen stammen, aufgetragen. Es ist interessant zu erkennen, dass in den meisten Diagrammen eine fast horizontale Avg-Linie zu verzeichnen ist.

Vergleicht man beim ACM-Corpus Precision und Recall, so sieht man anhand der Avg-Linie, dass ein Ausgleich besteht (Abbildung 4.6).

In der Abbildung 4.7 ist klar ersichtlich, dass die automatisch erstellten Zusammenfassungen aus dem Banksearch-Corpus aufgrund der hohen Recall- und niedrigen Precision-Werte viele irrelevante Informationen enthalten, auch wenn sie größtenteils die Kernpunkte wiedergeben. Abbildung 4.8 zeigt hingegen etwas bessere Ergebnisse.

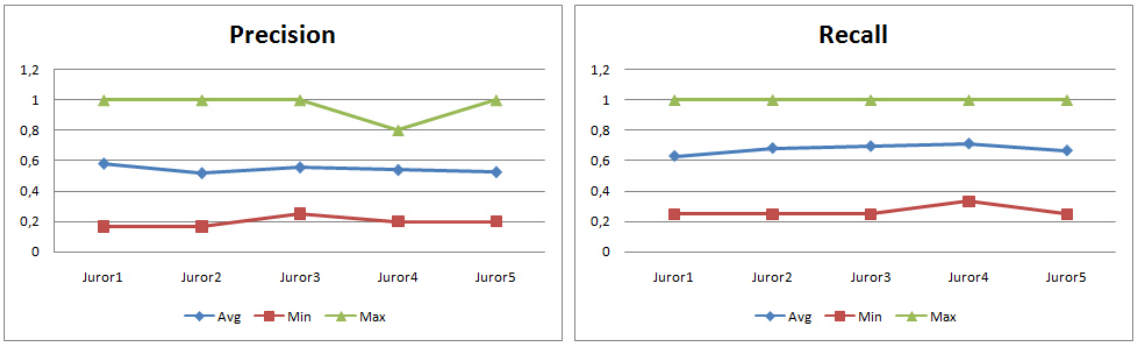


Figure 4.6: ACM-Corpus: Precision&Recall

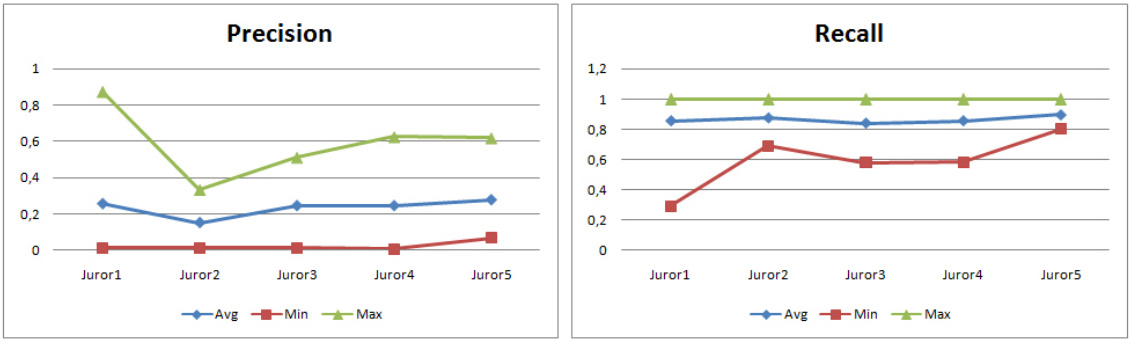


Figure 4.7: Banksearch-Corpus: Precision&Recall

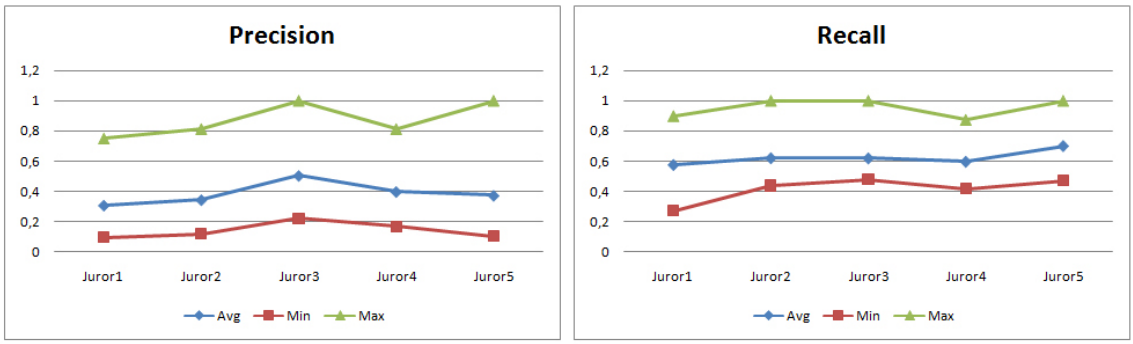


Figure 4.8: Lyrics-Corpus: Precision&Recall

Messung der Indexierungskonsistenz

In diesem Abschnitt soll nun auch die "'idealen"' Zusammenfassungen der Juroren untereinader verglichen werden. Dazu wird die Indexierungskonsistenz gemessen. Die Indexierungskonsistenz, im Englischen *inter-indexer consistency* genannt, ist ein Maß für die Übereinstimmung der vergebenen Deskriptoren zu einem Dokument durch zwei oder mehrere unabhängige Indexer.

Es gibt zahlreiche Möglichkeiten die Indexierungskonsistenz festzulegen. In dieser Arbeit wird speziell auf das Rolling-Maß bzw. Cosine-Maß eingegangen [Med06].

Seien A und B die Satzgrössen der beiden Indexer und C die gemeinsame Menge von A und B, so errechnet man die Indexierungskonsistenz mit dem Rolling measure bzw. dem Cosine measure durch folgende Formel:

$$R = \frac{2C}{A+B} \tag{4.4}$$

$$Cos = \frac{C}{\sqrt{AB}} \tag{4.5}$$

Sind A und B disjunkt, so erhalten beide Werte 0. Sind sie identisch, so ist sind die Werte 1.

Mit dieser Messung soll die Qualität der automatisch erstellten Zusammenfassungen wie auch die Unterschiede der manuell verfassten Zusammenfassungen untereinander abgeschätzt werden. Dazu werden die Sätze der zu überprüfenden Zusammenfassungen als Vergleichsbasis hergenommen. Errechnet man für jedes Paar von Extrakten das Rolling- bzw. Cosine-Maß, so erhält man folgende Tabellen 4.6 mit den entsprechenden Konsistenzen der Indexer in Prozent. Im rechten Teil der Tabellen werden die Cosine-Avg-Wert eingetragen.

Anhand der Tabelle 4.6 ist unschwer zu erkennen, dass Juror 5 bzw. Juror 3 den höchsten Durchschnittswert haben. Das System hingegen hat den niedrigsten Wert. Das bedeutet, dass automatisch erstellte Zusammenfassungen offenbar keine allzu großen Übereinstimmungen mit manuell verfassten Zusammenfassungen aufweisen wie bei jenen untereinander. Das System liefert nichts desto trotz brauchbare Ergebnisse.

	System	Juror 1	Juror 2	Juror 3	Juror 4	Juror 5	Avg	Cosine Avg
System		60,3	57,5	61,4	60,8	59,2	59,8	60,3
Juror 1	60,3		71,2	68,3	62,1	69,2	66,2	66,3
Juror 2	57,5	71,0		72,3	68,8	66,8	67,3	67,4
Juror 3	61,4	68,3	72,3		69,8	66,8	67,7	67,8
Juror 4	60,8	62,1	68,8	69,8		84,2	69,1	66,9
Juror 5	59,2	69,2	66,8	66,8	84,2		69,2	67,0
						Overall	66,6	66,0

Table 4.6: ACM Corpus: Inter-indexer consistency

Beim Banksearch-Corpus (Tabelle 4.7) liegt der Grad der Übereinstimmung zwischen den automatisch und der manuell erstellten Zusammenfassungen durchschnittlich bei 24,4% bzw. 34%. Hierbei haben Juror 5 bzw. Juror 1 durchschnittlich die meisten Übereinstimmungen.

	System	Juror 1	Juror 2	Juror 3	Juror 4	Juror 5	Avg	Cosine Avg
System		20,2	17,8	27,3	25,9	31,0	24,4	34,0
Juror 1	20,2		49,8	42,2	39,3	31,3	36,6	38,4
Juror 2	17,8	49,8		37,3	42,3	33,1	36,0	37,3
Juror 3	27,3	42,2	37,3		26,1	33,4	33,3	36,9
Juror 4	25,9	39,3	42,3	26,1		59,6	38,6	37,0
Juror 5	31,0	31,3	33,1	33,4	59,6		37,7	36,1
						Overall	34,4	36,6

Table 4.7: Banksearch Corpus: Inter-indexer consistency

Im Falle von Liedtexten erkennt man anhand der Tabelle 4.8, dass die vom System erzeugten Zusammenfassungen den Zusammenfassungen einiger Juroren mehr ähneln, als die der anderen Juroren untereinander. Beispielsweise haben Juror 3 und Juror 4 mehr Übereinstimmung mit dem System, als Juror 1 mit Juror 5. Juror 3 und 4 sind auch diejenigen mit den größten Durchschnittswerten.

	System	Juror 1	Juror 2	Juror 3	Juror 4	Juror 5	Avg	Cosine Avg
System		36,7	38,8	51,3	44,9	40,4	42,4	44,6
Juror 1	36,7		61,7	55,4	51,5	43,2	49,7	50,5
Juror 2	38,8	61,7		64,6	66,2	51,7	56,6	57,6
Juror 3	51,3	55,4	64,6		81,3	57,2	62,0	63,0
Juror 4	44,9	51,5	66,2	81,3		71,7	63,1	61,8
Juror 5	40,4	43,2	51,7	57,2	71,7		52,8	51,9
						Overall	54,5	54,9

Table 4.8: Lyrics Corpus: Inter-indexer consistency

4.3.6 Signifikanztest

Signifikanztests dienen zumeist dazu, die Aussagekraft der Ergebnisse zu überprüfen. Bevor man sie widerlegt, muss sie zunächst formuliert werden. In diesem Fall besagt die Nullhypothese, dass keine Unterschiede zwischen der automatisch erstellten Zusammenfassung und der idealen Zusammenfassungen bestehen. Nun ist die Wahrscheinlichkeit für das Zutreffen der Nullhypothese anhand des festzulegenden Signifikanzniveaus, welches hierbei auf den Standardwert 0,05 angesetzt wird, mittels eines geeigneten Verfahrens zu überprüfen. Da nicht angenommen werden kann, dass die ermittelten Daten normalverteilt sind, ist ein nichtparametrisches Verfahren zu wählen.

Als nichtparametrisches Verfahren wird der Wilcoxon-Vorzeichenrang-Test [4] verwendet. Als Vergleichsbasis wird der Precision-Level festgelegt. Dazu nimmt man die Precision-Werte der zu prüfenden Systeme (also die des Zusammenfassungssystems und die eines der vier Juroren), die durch den Vergleich mit den Referenzzusammenfassungen eines anderen Juroren errechnet werden. Zur Verdeutlichung der Vorgehensweise eines solchen Tests ist ein Beispiel in Anhang A aufgeführt.

Die Tabellen 4.9, 4.10 und 4.11 zeigen die Ergebnisse des Wilcoxon-Vorzeichenrang-Tests. Im linken Teil der Tabellen sind die Juroren, von denen die Referenzzusammenfassungen stammen, angeführt während im oberen Teil die zu vergleichenden Systeme stehen. Die mit "'X"' versehenen Felder bedeuten ein Zutreffen der Nullhypothese. Der Eintrag "'zw"' bedeutet ein Zurückweisen der Nullhypothese.

Anhand der Tabelle 4.9 sieht man, dass unter Verwendung der Referenzzusammenfassungen von Juror 4 die Unterschiede zwischen dem System und dem Juror 1 nicht signifikant sind. Allerdings ist in allen anderen Fällen ein signifikanter Unterschied zu verzeichnen.

[4] Sidney Siegel, Nichtparametrische statistische Methoden, Eschborn 1997, S.72

Referenz	SOM:Juror1	SOM:Juror2	SOM:Juror3	SOM:Juror4	SOM:Juror5
Juror1		zw	zw	zw	zw
Juror2	zw		zw	zw	zw
Juror3	zw	zw		zw	zw
Juror4	X	zw	zw		zw
Juror5	zw	zw	zw	zw	

Table 4.9: ACM Corpus: Signifikanztabelle

Tabelle 4.10 sagt aus, dass bei Verwendung der Referenzdokumenten von Juror 1 bzw. Juror 3 die Zusammenfassungen von Juror 5 und Juror 3 gegenüber den Zusammenfassungen des System keine signifikanten Unterschiede aufweisen.

Referenz	SOM:Juror1	SOM:Juror2	SOM:Juror3	SOM:Juror4	SOM:Juror5
Juror1		zw	zw	zw	X
Juror2	zw		zw	zw	zw
Juror3	zw	zw		X	zw
Juror4	zw	zw	zw		zw
Juror5	zw	zw	zw	zw	

Table 4.10: Banksearch Corpus: Signifikanztabelle

Auch bei den Liedtexten ist es ersichtlich, dass das System und die Juroren signifikant unterschiedlich sind. Es gibt einen Fall, in dem kein signifikanter Unterschied zwischen dem System und dem Juror 1, wobei als Referenz die Zusammenfassungen von Juror 5 genommen wurde.

Referenz	SOM:Juror1	SOM:Juror2	SOM:Juror3	SOM:Juror4	SOM:Juror5
Juror1		zw	zw	zw	zw
Juror2	zw		zw	zw	zw
Juror3	zw	zw		zw	zw
Juror4	zw	zw	zw		zw
Juror5	X	zw	zw	zw	

Table 4.11: Lyrics Corpus: Signifikanztabelle

4.4 Zusammenfassung

In diesem Kapitel wurden die Zusammenfassungen, erzeugt vom eigenen System, subjektiv und objektiv analysiert. Die subjektive Evaluierung erfolgte durch die Beantwortung von speziellen Fragen über Lesbarkeit, Aussagekraft und Kohärenz. Dabei stützte man sich an die von [Mur05] vorgestellte Evaluierung. In einem weiteren Schritt versuchte man dann festzustellen, ob manuell verfasste Zusammenfassungen gegenüber automatisch erstellten Extrakten überlegen sind.

Im objektiven Teil der Evaluierung wurden Extrakte von insgesamt fünf Juroren erstellt, die dann mit denen des Systems verglichen wurden. Methoden, wie das Berechnen der Precision- und Recall-Werte zur Beschreibung der Güte des Ergebnisses, wie auch die Messung der Indexierungskonsistenz wurden dabei angewandt. Zum Schluss wurden die Ergebnisse auf Signifikanz überprüft. Dabei wurde statistisch nachgewiesen, dass automatisch erstellte Zusammenfassungen sich von manuell erzeugten Zusammenfassungen signifikant unterscheiden. Dies bedeutet aber nicht, dass diese unbrauchbar sind. Denn im subjektiven Teil der Evaluierung hat man gesehen, dass im Falle von ACM-Texten automatische Zusammenfassungen ihre Zwecke erfüllen.

5 Conclusion & Future Work

5.1 Zusammenfassung

Ziel dieser Arbeit war es, Satzextraktionsalgorithmen zur automatischen Zusammenfassung von Textclustern zu analysieren, manche davon, wie auch eigene zu implementieren und die Ergebnisse anhand einer Reihe von manuell verfassten Zusammenfassungen zu vergleichen. Es wurden verschiedene Algorithmen, wie beispielsweise die von Luhn und Edmundson, bezüglich ihrer Satzauswahlmethoden untersucht. Man befasste sich aber auch mit aktuellen Forschungsarbeiten auf dem Gebiet der Automatischen Textzusammenfassung, wie beispielsweise die Multidokument-Zusammenfassung.

Als im Evaluierungsteil dieser Arbeit die eigenen impementierten Algorithmen an drei unterschiedliche Corpora getestet werden sollten, entschied man sich für eine intrinsische Evaluierung, in der die automatisch erstellten Zusammenfassungen mit von Hand erzeugten Referenzzusammenfassungen verglichen wurden. Dazu waren insgesamt fünf Teilnehmer aus verschiedenen Altersgruppen anzuwerben.

Die Durchführung der Evaluierung erwies sich als schwierig, da keine anerkannten Standards auf dem Gebiet der Evaluierung von Systemen zur Automatischen Textzusammenfassung existieren. Der Grund dafür liegt hauptsächlich in der Schwierigkeit Zusammenfassungen zu bewerten. Da es keine "'ideale"' Zusammenfassung gibt, ist es schwierig ein aussagekräftiges Ergebnis aus den Zusammenfassungen zu erhalten. Beispielsweise hat man im subjektiven Teil der Evaluierung gesehen, dass Juror 5 die automatisch erstellten Zusammenfassungen schlecht bewertete, im objektiven Teil jedoch gab es in einigen Fällen Gemeinsamkeiten in der Auswahl der relevanten Sätze zwischen dem Zusammenfassungssystem und dem Teilnehmer. Man sieht also, dass es zwischen ein und derselben Person Differenzen bei der Beurteilung gibt.

Auch wenn es sich um Extrakte handelt, wird es kaum der Fall sein, dass ein Mensch in einem entsprechenden Zeitabstand zwei exakt gleiche Zusammenfassungen deselben

Ausgangstextes erstellt. Umso weniger kann man sich von einem System erwarten, dass es eine Zusammenfassung erstellt, die mit denen von Menschen übereinstimmen.

Ein weiteres Problem bei der Evaluierung von Zusammenfassungen ist die Originaltreue bei kritischen Zusammenfassungen. Da diese nicht nur relevante Informationen des Originals, sondern auch Meinungen und Behauptungen des Textes beinhalten, ist es schwer für das System die Qualität des Textes ermitteln.

Der Grund für die großen Unterschiede zwischen den automatisch erstellten und den manuell erzeugten Zusammenfassungen liegt insbesonderte darin, dass das Einschätzen des Informationsgehalts einer Zusammenfassung von Menschen viel einfacher erfolgen kann als von einem System. Beim Lesen eines Textes kann ein Mensch sofort beurteilen, ob die Zusammenfassung kohärent ist oder nicht. Weiters kann er feststellen, ob eine Zusammenfassung die Kernpunkte des Ausgangstextes wiedergeben. Er kann auch erkennen, wenn Teile des Textes Synonyme oder Anonyme, also das In-Erscheinung-Treten ohne Identitätspreisgabe, beinhalten.

Um Textzusammenfassungssysteme kontinuierlich zu verbessern, gibt es jedes Jahr die Document Understanding Conference (DUC), deren Aufgaben darin bestehen, einen Überblick über das Thema Textzusammenfassung zu schaffen, Textzusammenfassungssysteme zu evaluieren und Forschern die Möglichkeit zur Teilnahme an umfangreichen Experimenten zu geben (siehe Abschnitt 2.6).

5.2 Verbesserungsansätze

Das Zusammenfassungssystem kann um viele Funktionen erweitert werden. Beispielsweise könnte es Dokumente in den unterschiedlichen Formaten, wie RTF, XML oder PDF, einlesen.

Bereits implementierte Methoden können durch verschiedene Maßnahmen, wie die Optimierung bei der Berechnung der Gewichte, verbessert werden.

Es können aber auch weitere Methoden implementiert werden, sowohl im Bereich der Singledokument- als auch der Multidokument-Zusammenfassung, um eine Verbesserung der Zusammenfassungen zu erzielen. So kann das System beispielsweise nutzer-orientierte Zusammenfassungen erstellen, die als Antwort einer Frage, die vom Benutzer an das System gestellt wurde, geliefert werden [Far03].

Eine andere Möglichkeit das Zusammenfassungssystem zu erweitern wäre, wenn es nicht nur Texte, sondern auch Bilder, die in den Ausgangstexten enthalten sind, in die Zusammenfassung miteinbezieht. Man kann sogar einen Schritt weitergehen und sich vorstellen, dass das System Bilder oder Diagramme zusammenfasst, indem es deren Aussagen in einer knapperen Form reduziert. Arbeiten, die sich mit diesen Themen befassen, gibt es bereits [Fut99].

Das System sollte auch in naher Zukunft in der Lage sein Texte in verschiedenen Sprachen zusammenzufassen. Somit kann es dazu verwendet werden relevante Informationen anhand von multilingualen Texten zu extrahieren ohne Wissen über die jeweiligen Fremdsprachen zu haben.

Die Suchfunktion kann in Verbindung mit einem Thesaurus verbessert werden, sodass bei der Suche nicht nur das eingegebene Wort, sondern auch Wortgruppen mit ähnlicher Bedeutung als Ergebnis zurückgeliefert werden. Ein Thesaurus ist ein Modell, das aus einer systematisch geordneten Sammlung von Begriffen, die in thematischer Beziehung zueinander stehen, besteht. So kann bei einer Suchanfrage von "'Tischler"' Worte wie "'Handwerker"' oder "'Tisch"', wobei diese aus anderen Dokumenten stammen könnten, als Ergebnis geliefert werden.

Bibliography

[All01] A.James, R.Gupta, V.Khandelwal, "'Temporal summaries of news topics"', In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM Press, New Orleans, Louisiana, United States, pages 10-18, 2001

[Ami00] E.Amitay, C.Paris, "'Automatically summarising Web sites: is there a way around it?"', In *Proceedings of the 9th International Conference on Information and Knowledge Management*, ACM Press, McLean, Virginia, United States, pages 173-179, 2000

[Aon99] C.Aone, M.E.Okurowski, J.Gorlinsky, B.Larsen, "'A trainable summarizer with knowledge acquired from robust NLP techniques"', In *Advances in Automatic Text Summarization*, MIT Press, Cambridge, pages 71-80, 1999

[Bar99] R.Barzilay, M.Elhadad, "'Using lexical chains for text summarization"', In *Advances in Automatic Text Summarization*, MIT Press, Cambridge, pages 111-121, 1999

[Bar99a] R.Barzilay, K.McKeown, M.Elhadad, "'Information fusion in the context of multi-document summarization"', In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, College Park, Maryland, pages 550-557, June 1999

[Bar01] R.Barzilay, N.Elhadad, K.McKeown, "'Sentence ordering in multidocument summarization"', In *Proceedings of the 1st International Conference on Human Language Technology Research*, Association for Computational Linguistics, Morristown, NJ, USA, pages 1-7, 2001

[Buc97] C.Buckley, C.Cardie, "'Using empire and smart for high-precision IR and summarization"', In *Proceedings of the TIPSTER Text Phase III 12-Month Workshop*, San Diego, CA, October 1997

[Car97] J.G.Carbonell, Y.Geng, J.Goldstein, "'Automated query-relevant summarization and diverstiy-based reranking"', In *Proceedings of the IJCAI-97 Workshop on AI in Digital Libraries*, Melbourne, Australia, pages 12-19, 1997

[Car98] J.G.Carbonell, J.Goldstein, "'The use of MMR, diversity-based reranking for reordering documents and producing summaries"', In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM Press, Melbourne, Australia, pages 335-336, 1998

[Con01] J.Conroy, D.P.O'Leary, "'Text summarization via hidden Markov models"', In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM Press, New Orleans, Louisiana, United States, pages 406-407, 2001

[Dan05] H.T.Dang, "'Overview of DUC 2005"', http://duc.nist.gov/pubs/2005papers/OVERVIEW05.pdf, National Institute of Standards and Technology, Gaithersburg, 2005

[Dej78] G.F.DeJong, "'Fast Skimming of News Stories: The FRUMP System"', Ph.D.thesis, Yale University, New Haven, CT, 1978

[Edm69] H.P.Edmundson, "'New Methods in Automatic Extracting"', In *Journal ACM*, ACM Press, Maryland, 16(2): 264-285, 1969

[Far03] A.Farzindar, G.Lapalme, "'Using Background Information for Multi-document Summarization and Summaries in Response to a Question"', In *HLT-NAACL 2003 Workshop on Text Summarization*, Edmonton, Canada, 2003

[Fut99] R.P.Futrelle, "'Summarization of Diagrams in Documents"', In *Advances in Automated Text Summarization*, MIT Press, Cambridge, pages 403 - 422, 1999

[Gol99] J.Goldstein, M.Kantrowitz, V.Mittal and J.Carbonell, "'Summarizing text documents: Sentence selection and evaluation metrics"', In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM Press, Berkeley, California, United States, pages 121-128, 1999.

[Gol00] J.Goldstein, V.Mittal, J.Carbonell, M.Kantrowitz, "'Multi-document summarization by sentence extraction"', In *NAACL-ANLP 2000 Workshop on Automatic summarization - Volume 4*, Association for Computational Linguistics, Seattle, Washington, pages 40-48, 2000

[Gue05] A.Guendogan, R.Schimpfky, "'Text Summarization"', http://141.20.20.55/ schimpfk/download/TextSummarization.pdf, Seminararbeit, Humboldt-Universität Informatik, Berlin, Feb 2005

[Hah97] U.Hahn, U.Reimer, "'Knowledge-based text summarization: Salience and generalizatioin operators for knowledge base abstraction"', In *Advances in Automatic Text Summarization*, MIT Press, Cambridge, pages 215-232, 1997

[Hah00] U.Hahn, I.Mani, "'The Challenges of Automatic Summarization"', IEEE Computer Society Press, Cambridge, pages 29-36, Nov 2000

[Hov88] E.H.Hovy, "'Planning Coherent Multisentential Text"', In *Proceedings of the 26th Annual Meeting of the Association for Computational Linguistics*, Buffalo, New York, pages 163-169, June 1988

[Hov98] E.H.Hovy, C.Y.Lin, "'Automated text summarization and the summarist system"', In *Proceedings of a workshop on held at Baltimore*, Associaton for Computational Linguistics, Baltimore, Maryland, pages 197-214, 1998

[Hov99] E.H.Hovy, C.Y.Lin, "'Automated Text Summarization in SUMMARIST"', In *Advances in Automatic Text Summarization*, MIT Press, Cambridge, pages 81-94, 1999

[Jin99] H.Jing, K.McKeown, "'The decomposition of human-written summary sentences"', In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM Press, Berkeley, California, United States, pages 129-136, August 1999

[Kni00] K.Knight, D.Marcu, "'Statistics-based summarizatioin - Step one: Sentence compression"', In *Proceedings of the 17th National Conference of the American Association for Artificial Intelligence*, Austin, Texas, pages 703-710, 2000

[Kru06] M.Krübel, "'Analyse und Vergleich von Extraktionsalgorithmen für die Automatische Textzusammenfassung"', Diplomarbeit, Technische Universität Chremnitz, 2006

[Kup95] J.Kupiec, J.Pederson, F.Chen, "'A trainable document summarizer"', In *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM Press, Seattle, Washington, United States, pages 68-73, 1995

[Lam01] A.M.Lam-Adesina, G.J.F.Jones, "'Applying summarization techniques for term selection in relevance feedback"', In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM Press, New Orleans, Louisiana, United States, pages 1-9, 2001

[Law03] D.J.Lawrie, "'Language Models for Hierarchical Summarization"', http://www.cs.loyola.edu/" lawrie/papers/lawrieThesis.pdf, PhD thesis, University of Massachusetts Amherst, 2003

[Lie97] R.Lienhart, S.Pfeiffer, W.Effelsberg, "'Video Abstracting"', In *Commun.ACM*, ACM Press, 40(12): 54-62, 1997

[Lin97] C.Y.Lin, E.Hovy, "'Identifying topics by position"', In *Proceedings of the 5th Conference on Applied Natural Language Processing*, Morgan Kaufmann Publishers Inc., Washington, DC, pages 283-290, 1997

[Lin99] C.Y.Lin, "'Training a selection function for extraction"', In *Proceedings of the 8th International Conference on Information and Knowledge Management*, ACM Press, Kansas City, Missouri, United States, pages 55-62, 1999

[Lin02] C.Y.Lin, E.Hovy, "'From single to multi-document summarization: A prototype system and ist evaluation"', In *Proceedings of the 40th Conference of the Association of Computational Linguistics*, Association of Computational Linguistics, Philadelphia, pages 457-464, July 2002

[Luh58] H.P.Luhn, "'The Automatic Creation of Literature Abstracts"', In *IBM Journal of Research and Development*, 2(2): 159–165, April 1958

[Man97] I.Mani, E.Bloedorn, "'Multi-document summarization by graph search and matching"', In *Proceedings of the 14th National Conference on Artificial Intelligence*, AAAI Press, Providence, Rhode Island, pages 622-628, 1997

[Man99] I.Mani, B.Gats, E.Bloedorn, "'Improving summaries by revising them"', In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, College Park, Maryland, pages 558-565, June 1999

[Man01] I.Mani, M.T.Maybury, "'Automatic Summarization"', In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, Toulouse, France, 2001

[Mar01] D.Marcu, L.Gerber, "'An inquiry into the nature of multidocument abstracts, extracts, and their evaluation"', In *Proceedings of the Workshop on Text Summarization at the 2nd Conference of the North American Association of Computational Linguistics*, Pittsburgh, pages 1-8, 2001

[Mar01b] D.Marcu, "'Discourse-Based Summarization in DUC-2001"', In *Proceedings of the 2001 Document Understanding Conference*, New Orleans, Louisiana, United States, September 2001

[Mcd85] D.D.McDonald, J.D.Pustejovsky , "'Description-directed natural language generation"', In *Proceedings of the 9th International Joint Conference on Artificial Intelligence*, Los Angeles, CA, United States, pages 799-807, 1985

[Mck85] K.R.McKeown, "'Text Generation: Using Discourse Strategies and Focus Constraints to Generate Natural Language Text"', Cambridge University Press, Cambridge, England, 1985

[Mck95] K.R.McKeown, D.R.Radev, "'Generating summaries of multiple news articles"', In *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM Press, Seattle, Washington, United States, pages 74-82, 1995

[Mck99] K.R.McKeown, J.L.Klavans, V.Hatzivassiloglou, R.Barzilay, E.Eskin, "'Towards multidocument summarization by reformulation: Progress and prospects"', In *Proceedings of the 16th National Conference on Artificial Intelligence and the*

11th Innovative Applications of Artificial Intelligence Conference Innovative Applications of artificial intelligence, American Association for Artificial Intelligence, Orlando, Florida, United States, pages 453-460, July 1999

[Med06] O.Medelyan, I.H.Witten, "'Measuring Inter-Indexer Consistency using a Thesaurus"', In *Proceedings of the 6th ACM/IEEE-CS Joint Conference on Digital Libraries*, ACM Press, Chapel Hill, NC, United States, pages 274-275, 2006

[Mil95] G.A.Miller, "'WordNet: a lexical database for English"', In *Commun.ACM*, ACM Press, 38(11): 39-41, 1995

[Moo06] R.J.Mooney, "'Performance Evaluation of Information Retrieval Systems"', http://www.cs.utexas.edu/" mooney/ir-course/slides/Evaluation.ppt, University of Texas at Austin, 2006

[Mur05] G.Murray, S.Renals, J.Carletta, J.Moore, "'Evaluating Automatic Summaries of Meeting Recordings"', In *Workshop on Intrinsic and Extrinsic Evaluation Measures for MT and/or Summarization at the 43rd Annual Meeting of the Association for Computational Linguistics*, Ann Arbor, MI, United States, 2005

[Nen04] A.Nenkova, R.Passonneau, "'Evaluating content selection in summarization: the pyramid method"', In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics and Human Language Technology*, Boston, United States, pages 145-152, 2004

[Net00] J.L.Neto, A.D.Santos, C.A.A.Kaestner, A.A.Freitas, "'Document Clustering and Text Summarization"', In *Proceedings of the 4th International Conference Practical Applications of Knowledge Discovery and Data Mining*, The Practical Application Company, London, UK, pages 41-55, 2000

[Net00b] J.L.Neto, A.D.Santos, C.A.A.Kaestner, A.A.Freitas, "'Generating Text Summaries through the Relative Importance of Topics"', In *Proceedings of the International Joint Conference, 7th Ibero-American Conference on AI*, Springer-Verlag, Atibaia, SP, Brazil, pages 300-309, 2000

[Ove03] P.Over, J.Yen, "'An Introduction to DUC 2003: Intrinsic Evaluation of Generic News Text Summarization Systems"', In *Proceedings of the 2003 Document Understanding Conference*, Edmonton, Canada, 2003

[Pai81] C.D.Paice, "'The automatic generation of literature abstracts: an approach based on the identification of self-indicating phrases"', In *Proceedings of the 3rd Annual ACM Conference on Research and Development in Information Retrieval*, Butterworth & Co., Cambridge, England, pages 172-191, 1981

[Pas06] R.J.Passonneau, K.McKeown, S.Sigelman and A.Goodkind, "'Applying the Pyramid Method in the 2006 Document Understanding Conference"', In *Proceedings of the 2006 Document Understanding Conference*, Brooklyn, New York, United States, 2006

[Rad98] D.R.Radev, K.R.McKeown, "'Generating natural language summaries from multiple on-line sources"', In *Computational Linguistics*, MIT Press, 24(3): 470-500, 1998

[Rad99] D.R.Radev, V.Hatzivassiloglou, K.R.McKeown, "'A description of the CIDR system as used for TDT-2"', In *Proceedings of DARPA HUB4 Broadcast News Workshop*, Washington D.C., United States, 1999

[Rad00] D.R.Radev, H.Jing, M.Budzikowska, "'Centroid-based summarization of multiple documents: Sentence extraction, utility-based evaluation and user studies"', In *NAACL-ANLP 2000 Workshop on Automatic summarization - Volume 4*, Association for Computational Linguistics, Seattle, Washington, pages 21-30, 2000

[Rad02] D.R.Radev, K.R.McKeown, E.H.Hovy, "'Introduction to the special issue on Summarization"', In *Computational Linguistics*, MIT Press, 28(4): 399-408, 2002

[Rad02a] D.R.Radev, S.Teufel, H.Saggion, W.Lam, J.Blitzer, A.Celebi, H.Qi, E.Drabek, D.Liu, "'Evaluation of text summarization in a cross-lingual informatioin retrieval framework"', Technical Report, Center for Language and Speech Processing, Johns Hopkins University, Baltimore, June 2002

[Rau91] L.F.Rau, P.S.Jacobs, "'Creating segmented databases from free text for text retrieval"', In *Proceedings of the 14th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM Press, Chicago, Illinois, United States, pages 337-346, 1991

[Sag03] H.Saggion, K.Bontcheva, H.Cunningham, "'Robust Generic and Query-based Summarisation"', In *Proceedings of the 10th Conference on European Chapter of*

the Association for Computational Linguistics, Association for Computational Linguistics, Budapest, Hungary, pages 235-238, 2003

[Sal97] G.Salton, A.Singhal, M.Mitra, C.Buckley, "'Automatic text structuring and summarization"', In *Information Processing and Management*, Pergamon Press Inc., 33(2): 193-207, 1997

[Spi02] A.Spink, B.Jansen, D.Wolfram, T.Saracevic, "'From e-sex to e-commerce: Web search changes"', In *Computer*, IEEE Computer Society Press, 35(3): 107-109, 2002

[Str99] T.Strzalkowski, G.Stein, J.Wang, B.Wise, "'A robust practical text summarizer"', In *Advances in Automatic Text Summarization*, MIT Press, Cambridge, pages 137-154, 1999

[Tai83] J.I.Tait, "'Automatic Summarizing of English Texts"', PhD thesis, University of Cambridge, Cambridge, UK, 1983

[Whi02] M.White, C.Cardie, "'Selecting sentences for multidocument summaries using randomized local search"', In *Proceedings of the ACL-02 Workshop on Automatic Summarization*, Association for Computational Linguistics, Phildadelphia, Pennsylvania, pages 9-18, 2002

[Wit99] M.Witbrock, V.O.Mittal, "'Ultra-summarization: A statistical approach to generating highly condensed non-extractive summaries"', In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM Press, Berkeley, California, United States, pages 315-316, 1999

A Anhang

A.1 Verwendete Dokumente für die Evaluierung

A.1.1 ACM-Corpus

0407w_4.txt, 0218w_0.txt, 0524m_1.txt, 1210w_4.txt, 1010f_4.txt, 0614m_14.txt, 0811m_1.txt, 0423f_18.txt, archives4d71_10.txt, 1103m_2.txt, 0331w_3.txt, 1020m_4.txt, 0220f_16.txt, archivese507_0.txt, 0226w_0.txt, archivesf93e_0.txt, 0204f_0.txt, 0211f_0.txt, 0702f_0.txt, 0308m_4.txt, 0411m_0.txt, 0804m_3.txt, 0211w_14.txt, 0621m_10.txt, 0823m_0.txt, 0426m_13.txt, 0324m_5.txt, 0220f_15.txt, 0214m_4.txt, 1020w_11.txt, 0521w_3.txt, 0103f_8.txt, 0611w_13.txt, 0416f_13.txt, 0725f_4.txt, 0319w_5.txt, 0130f_4.txt, 1103w_14.txt, archives6b2b_2.txt, 0303m_0.txt, 0324w_7.txt, 0412m_3.txt, 0317m_3.txt, 0428m_6.txt, archives6115_8.txt, 1222w_0.txt, 0114w_3.txt, 0514f_6.txt, 0604w_11.txt, archives7bce_4.txt, archivesdbd7_8.txt, 0321f_18.txt, 0213f_7.txt, 0210m_6.txt, 1010f_10.txt, 0609m_6.txt, 0328f_18.txt, 0112m_14.txt, archives3ef0_9.txt, 0109f_12.txt, archivesd40e_3.txt, 0226w_9.txt, 0825w_12.txt, archives4988_12.txt, 1008w_11.txt, 0401f_0.txt, archives75db_3.txt, 0630m_14.txt, 1020m_18.txt, 1210w_8.txt, 0416w_11.txt, 0910w_9.txt, 0714w_16.txt, archives30cb_11.txt, 0716f_3.txt, 0331w_17.txt, 0804w_13.txt, 0402w_7.txt, 0107w_7.txt, 1015f_11.txt, 1015f_13.txt, 1022f_5.txt, 0408f_17.txt, archivesa24d_12.txt, 0519w_8.txt, 0206f_7.txt, 0315m_10.txt, archives0aa6_11.txt, archives7bd9_9.txt, archives1fb8_3.txt, 0428m_13.txt, 0602m_12.txt, archives7860_3.txt, 0319w_14.txt, archivesac3b_4.txt, 0917f_10.txt, archivesc613_6.txt, archives821_10.txt, archives3adc_0.txt, archives5389_1.txt, 0922w_2.txt, archivesdaa70_00.txt, archives8725_18.txt, 0806f_0.txt, archives2b7f_10.txt, archives030b_9.txt, archives14a7_8.txt, archivesec45_4.txt, 0214f_5.txt, archives4504_11.txt, 0813f_1.txt, 0416f_8.txt, 0809m_0.txt, archives8f88_1.txt, archivesdbd7_2.txt, 0825w_15.txt, archivesac1c_0.txt, 0707w_0.txt, 0318f_6.txt, 0301m_0.txt, archives47cd_4.txt,

0730f_4.txt, 0108w_7.txt, 1203w_8.txt, 0110f_0.txt, archives4400_6.txt, 0411m_1.txt, archivesa62f_13.txt, archives1292_5.txt, archives6b7b_4.txt, archives1fb8_0.txt, 0408f_0.txt, archivesb635_0.txt, archives0aa6_16.txt, archives7521_0.txt, archivesa94f_6.txt, archivesac3b_9.txt, 1124w_8.txt, archives795c_12.txt, 0721w_0.txt, archivesc488_0.txt, 1029f_4.txt, archives30d1_6.txt, 1013w_12.txt, archives5389_0.txt, archivese02c_12.txt, archives266a_12.txt, 0330w_1.txt, archives1bfb_4.txt, archives7862_16.txt, 0426m_12.txt

A.1.2 Banksearch-Corpus

e0793.txt.txt, e0300.txt.txt, e0123.txt.txt, e0813.txt.txt, e0617.txt, e0497.txt, e0449.txt, e0096.txt, e0069.txt, e0545.txt, e0569.txt, e0042.txt, e0275.txt, e0664.txt, e0708.txt, e0852.txt, e0686.txt, e0641.txt, g0889.txt, g0925.txt, g0961.txt, g0997.txt, g0359.txt, g0399.txt, g0439.txt, g0517.txt, g0556.txt, g0594.txt, g0631.txt, g0668.txt, g0705.txt, g0742.txt, g0779.txt, g0816.txt, g0853.txt, g0319.txt, g0079.txt, g0047.txt, g0327.txt, g0199.txt, g0201.txt, g0676.txt, g0279.txt, g0159.txt, g0450.txt, g0127.txt, g0039.txt, g0849.txt, g0116.txt, g0353.txt, g0923.txt, g0357.txt, x0916.txt, x0811.txt, x0967.txt, x0302.txt, x0838.txt, x0420.txt, x0362.txt, x0507.txt, x0094.txt, x0123.txt, x0182.txt, x0213.txt, x0093.txt, x003.txt, x0478.txt, x0063.txt, x0332.txt, x0890.txt, x0449.txt, x0002.txt, x0212.txt, x0391.txt, x0152.txt, i0094.txt, i0274.txt, i0244.txt, i0613.txt, i0968.txt, i0717.txt, i0691.txt, i0034.txt, i0424.txt, i0304.txt, i0364.txt, i0334.txt, i0992.txt, i0743.txt, i0944.txt, i0064.txt, i0920.txt, i0508.txt, i0665.txt, i0481.txt, i0896.txt, i0821.txt, i0561.txt, i0004.txt, i0811.txt, f0521.txt, i0226.txt, f0508.txt, i0497.txt, i0629.txt, c0471.txt, f0725.txt, c0960.txt, i0406.txt, i0441.txt, j0025.txt, x0764.txt, x0516.txt, x0871.txt, f0507.txt, f0534.txt, i0316.txt, e0118.txt, i0436.txt, i0681.txt, a0701.txt, x0106.txt, x0312.txt, d0688.txt, i0707.txt, i0260.txt, d0672.txt, c0400.txt, a0842.txt, d0973.txt, f0711.txt, d0850.txt, d0917.txt, i0893.txt, g0465.txt, i0610.txt, i0505.txt, g0984.txt, e0868.txt, j0235.txt, a0232.txt, a0010.txt, c0478.txt, j0392.txt, j0127.txt, f0780.txt, i0286.txt, f0284.txt, i0238.txt, e0491.txt, e0539.txt

A.1.3 Lyrics-Corpus

Richard12918.txt.txt, Richard12922.txt.txt, Richard12916.txt, Beethoven1247.txt, Mozart7287.txt, Richard12924.txt, Richard12900.txt, Beethoven1259.txt, Beethoven1245.txt, Beethoven1305.txt, Franz2813.txt, Mozart7029.txt, Beethoven1251.txt, Richard12878.txt, Richard12902.txt, Richard12914.txt, Beethoven1277.txt, Richard12884.txt, Mozart6941.txt, Richard12890.txt, Mozart7225.txt, Richard12882.txt, Beethoven1285.txt, Mozart7315.txt, Beethoven1303.txt, Richard12926.txt, Beethoven1269.txt, Richard12904.txt, Beethoven1279.txt, Richard12894.txt, Slow14760.txt, Hard3279.txt, Rock13076.txt, Metal6499.txt, Punk10870.txt, Metal6493.txt, Pop7892.txt, Metal5921.txt, Hip3895.txt, Metal6245.txt, Pop8376.txt, Slow14594.txt, Indie5485.txt, Grunge3025.txt, Metal6355.txt, Punk10444.txt, Garage2891.txt, Hip4159.txt, Christian1557.txt, Goth2987.txt, Punk11382.txt, Rock14188.txt, Punk10602.txt, Punk10628.txt, Pop8586.txt, Punk9698.txt, Rock13420.txt, Rock13422.txt, Rock13450.txt, Punk10384.txt, Avantgarde1175.txt, Reggae12836.txt, Metal6641.txt, Punk11018.txt, Rock12934.txt, Alternative739.txt, Pop8188.txt, Punk10128.txt, Pop7870.txt, Pop7900.txt, Metal5865.txt, Indie5469.txt, Rock13202.txt, Ska14388.txt, Punk9664.txt, Electronic1957.txt, Metal6143.txt, Alternative641.txt, Metal6697.txt, Rock13784.txt, Rock13800.txt, Punk11938.txt, Pop8616.txt, Soundtrack15476.txt, Metal6749.txt, Emo2501.txt, Metal5787.txt, Emo2389.txt, Indie5381.txt, Indie4955.txt, Hip4325.txt, Hip4333.txt, Hip4351.txt, Rock13610.txt, Hip4307.txt, New7609.txt, Indie5321.txt, Indie5307.txt, Pop8540.txt, Pop8466.txt, Hardcore3613.txt, Punk11870.txt, Folk2769.txt, Indie5293.txt, Indie5085.txt, Pop8596.txt, Punk9508.txt, Indie5369.txt, Indie5317.txt, Rock14160.txt, Punk12124.txt, Pop8758.txt, Hip3915.txt, Punk11242.txt, Slow14756.txt, Alternative325.txt, Pop8646.txt, Metal6349.txt, Rock13952.txt, Rock12968.txt, Metal6383.txt, Rock13896.txt, Pop9158.txt, Punk9854.txt, Hip3891.txt, Punk12054.txt, Punk11246.txt, Punk11480.txt, Electronic2035.txt, Hip4793.txt, Alternative507.txt