Ruprecht-Karls-Universität
Heidelberg

Insitut für Computerlinguistik
Neuphilologische Fakultät

Bachelor Arbeit
zur Erlangung
des Grades eines Bachelor of Arts

Akquisition vergleichbarer Absätze aus Wikipedia

Danny Rehl

Danny Rehl, 2220185

Oberdorfstraße 58
69124 Heidelberg

rehl@cl.uni-heidelberg.de

Computerlinguistik, NBA 75%
Deutsche Philologie, NBA 25%

Betreuer	Prof. Dr. Sebastian Padó
Semester	Sommersemester 2011
Abgabedatum	**21. Juli 2011**

Inhaltsverzeichnis

abstract

Computers are present in almost any task in our modern world, yet they cannot inherently understand our languages and world. Researchers have developed mechanisms and algorithms to enable computers to learn, mostly through organized information.
Natural language processing heavily relies on written text data that is organized in a corpus, which is at best annotated and domain specific. Translation models for machine translation are just one of the various applications that make use of a special form of corpus, a parallel corpus, which features aligned sentences in different languages.

The objective of this work is to create a system to find and reliably identify generally comparable paragraphs in texts of a german-english online encyclopedia rather than parallel text passages. Incorporating state-of-the-art sentence alignment procedures, text-mining and methods for identification of parallel text passages, a sophisticated system with a feature-based point system and an optimized parameter was developed to enable binary classification of candidates for comparable paragraphs.

This work not only resulted in a classification tool to acquire comparable paragraphs. There were two uniquely new resources created: a german-english corpus aligned by articles and split into individual paragraphs, as well as a new gold standard containing comparability annotations.

Finally, the newly developed system is evaluated against the new gold standard and the results are presented.

1 Einleitung

Um mit Hilfe von Computern die Welt mit ihrer Sprachenvielfalt besser in den Griff zu bekommen, sind annotierte, also mit Metadaten versehene, Korpora, die aus mehreren Sprachen bestehen[1], wichtig. Computer verstehen die Welt per se nicht, können aber aus Informationen lernen. Beispielsweise können Übersetzungsmodelle für maschinelle Übersetzungssysteme mit parallelen Korpora erstellt werden.

Parallele Korpora sind Textsammlungen, welche eine Zuordnung eines Textes zu einer entsprechenden Übersetzung liefern. Sie sind eine Teilmenge von vergleichbaren Korpora.[2] Vergleichbare (nicht nur multilinguale) Korpora sind demnach Textsammlungen, welchen eine gemeinsame, semantische Schnittmenge ihrer vergleichenden Texte inhärent ist.

Es gibt ein sehr bekanntes Beispiel eines solchen Korpus: Wikipedia. Durch die *interlanguage links*, die von einem Artikel einer Sprache zu den vergleichbaren Artikeln der anderen Sprache führen, ist diese multilinguale Textsammlung, ein großes vergleichbares Korpus.

Mit Texten sind hier ganze Artikel gemeint, welche eben mit diesen *interlanguage links* in Verbindung gebracht werden können. Dabei kann der Textinhalt eines solchen Artikels sehr groß sein und der Inhalt von miteinander in Verbindung stehenden Artikeln zum Ende hin immer mehr divergieren.
Aus diesem Grund soll, basierend auf Wikipedia, ein neues Korpus geschaffen werden, um diesem Umstand entgegenzuwirken. Die Artikel werden in ihre Absätze zerlegt, und diese Absätze bilden dann die Grundlage für die Suche und für die Akquisition vergleichbarer Textstellen.

[1] Diese werden gängigerweise als parallele Korpora bezeichnet, wenn eine Zuweisung von Texten der einen Sprache zu Texten einer anderen Sprache modelliert ist.
[2] Ist ein Textabschnitt mit einem anderen vergleichbar, so könnte er auch parallel sein.

Der Unterschied zu bisherigen Ansätzen besteht darin, dass die Absätze nicht vorwiegend als Strukturelemente gesehen werden, denen mehrere inhaltstragende Bedeutungen zugemessen werden, sondern dass Absätze an sich schon eine inhaltliche Einheit bilden. Es werden Methoden des *sentence alignments* oder der Identifikation paralleler Textstellen aufgegriffen und auf diese größere Texteinheit übertragen.

Schließlich soll herausgefunden werden, wie gut vergleichbare Textstellen auf der Absatzebene identifiziert werden können.

2 Bisherige Ansätze und Ressourcen

2.1 Bisherige Ansätze

Zur absatzorientierten Identifikation von vergleichbaren bilingualen Textstellen gibt es bislang nur wenige, bzw. keine Ansätze. Hier sollen lediglich ähnliche Ansätze aufgezeigt werden, deren Methoden teilweise für diesen Ansatz übernommen wurden.

"Automatic Acquisition of Parallel Corpora from Websites with Dynamic Content" von (Tsvetkov and Wintner, 2010)

(Tsvetkov and Wintner, 2010) haben einen Wörterbuch-basierten Algorithmus entwickelt, um parallele Textstellen im World Wide Web zu extrahieren. Dafür setzen sie ihren Algorithmus auf bilinguale Newspaper-Seiten an.
Um die parallelen Textstellen zu identifizieren, nehmen sie einen *bag-of-words*-Ansatz an. Diese werden mit statistischen Verfahren auf ihre Ähnlichkeit hin untersucht. Diese Untersuchung wird bidirektional durchgeführt und der Ansatz beschränkt sich auf die beiden Sprachen Hebräisch und Englisch.

"The Web as a Parallel Corpus" von (Resnik and Smith, 2003)

Das entwickelte System von (Resnik and Smith, 2003) sucht Kandidatenpaare im World Wide Web, die möglicherweise parallel sein könnten.
Diese werden dann hinsichtlich struktureller Merkmale analysiert und mit statistischen Ähnlichkeitsmaßen kombiniert, um parallele Textstellen identifizieren zu können. Parameter werden mit maschinellen Lernmethoden optimiert.

"A Program for Aligning Sentences in Bilingual Corpora" (Gale and Church, 1993)

Die Methode von (Gale and Church, 1993) sucht alinierte Sätze im *Canadian Hansards*, das sind Parlamentsdebatten, die in Textform und in mehreren Sprachen vorliegen.

Dabei liegt ihr Hauptaugenmerk auf der Zeichenkettenlänge und nicht etwa der Wortlänge von Sätzen. Ein statistisches Modell der Zeichenkettenlänge teilt Kandidatenpaare Wahrscheinlichkeitswerte zu. Um auf die Satzebene zu kommen, werden zuvor Absätze miteinander aliniert.

2.2 Ressourcen

Um die Aufgabe bewältigen zu können, ein vergleichbares absatzaliniertes Korpus aus Wikipedia zu akquirieren, wurde, inspiriert durch (Tsvetkov and Wintner, 2010), ein lexikonbasierter Ansatz herangezogen. Neben Wikipedia selbst als deutsch-englische Textquelle ist dabei ein bilinguales Wörterbuch, das beide Sprachen umfasst, unerlässlich.[1] Im weiteren soll zusätzlich aufgezeigt werden, welche Ressourcen für diese Aufgabe in Betracht gezogen wurden oder prinzipiell in Betracht gezogen werden können. Dann wird Auskunft darüber gegeben, warum eine Ressource letztendlich auserkoren wurde.

2.2.1 Wikipedia

Wikipedia kann als großes[2], multilinguales[3] Korpus gesehen werden. Dieses ist frei verfügbar und es werden freundlicherweise regelmäßige *dumps*[4] zum Herunterladen angeboten. In diesem Wikipedia-Abbild werden keine Multimedia-Dateien wie etwa Bilder, Grafiken oder Audio-Dateien abgespeichert. Daher eignet es sich besonders gut für maschinelle Textverarbeitung.
Als problematisch erweist sich hier eine uneinheitliche Typologie in der eigens entwickelten Wikipedia-Syntax, was sich in zahlreichen Überresten dieser bestimmten Syntax in den *dumps* niederschlägt.

[1] Die Schwierigkeiten in der Behandlung von Mehrsprachigkeit wurde aufgrund des zeitlichen Umfangs dieser Arbeit auf die Alinierung von deutschen mit englischen Absätzen heruntergebrochen.
[2] Wikipedia selbst gibt Auskunft über die Zahl der Artikel. Der Stand von Ende Juni 2011 besagt, dass die englische Wikipedia 3.649.867 Artikel vorzuweisen hat, die deutsche Wikipedia immerhin mit 1.241.312 an zweiter Stelle steht. (vgl. `http://stats.wikimedia.org/`)
[3] Wikipedia ist ein multilinguales Korpus, weil verschiedene Sprachen dem Korpus inhärent sind. So gibt es die englische Wikipedia, die deutsche Wikipedia, die türkische, usw. Die Beziehung von Artikeln, die das Gleiche beschreiben, aber in verschiedenen Sprachen, wird über sogenannte *interlanguage links* hergestellt, welche in den Artikeln selbst zu finden sind.
[4] Ein dump ist ein Abbild eines Datenbestandes zu einer bestimmten Zeit.

Eine Bereinigung dieser Wikipedia Syntax ist nicht nötig, da dem Institut für Computerlinguistik in Heidelberg bereits ein englisches und deutsches, bereinigtes Wikipedia in Textform vorliegt. Die englische Wikipedia-Datei besteht aus 2.841.715 Artikeln und die deutsche aus 781.564 Einträgen. Hier sind nicht nur multimediale Elemente entfernt, sondern auch mathematische Formeln, Infoboxen, Überschriften, usw.
Der wesentliche Vorteil liegt in der einfachen Extraktion von Absätzen. Diese sind mit zwei Leerzeilen voneinander getrennt. Als problematisch hat sich allerdings vorliegende Zeichenkodierung[5] herausgestellt. Ein Nachteil für vorliegenden Ansatz ist zudem, dass die Artikel selbst nicht zueinander aliniert sind. Es wurden zwar IDs für die deutschen und englischen Artikel vergeben, diese entsprechen aber jeweils der alphabetischen Sortierung der Artikelnamen. Ein artikelaliniertes Wikipedia ist sinnvoll, da man vermutlich vergleichbare Absätze eher in vergleichbaren Artikeln findet.

Eine Verwendung dieser Ressource, um den Vorteil einer einfachen Absatzextraktion nutzen zu können, ist dennoch möglich, wenn eine Artikelalinierung dafür nachträglich hergestellt werden kann. Dies wird in Kap. 3.4 ausführlich dargestellt.

2.2.2 Bilinguales Wörterbuch

Absätze bestehen aus Sätzen, die wiederum aus Wörtern bestehen. In einem bilingualen Ansatz, in dem Absätze auf Vergleichbarkeit geprüft werden sollen, ist es sinnvoll, dieses hauptsächlich auf der Wortebene zu vollführen. Dafür muss man wissen, ob ein Wort in der einen Sprache auch in der anderen Sprache in seiner übersetzten Form zu finden ist. Dafür benötigt man ein bilinguales Wörterbuch, das diesen Abgleich leisten kann.

Ein maschinenlesbares, bilinguales Wörterbuch mit großer Abdeckung und ohne lexikographische Fehler wäre dabei nicht nur für vorliegende Arbeit wünschenswert. Solch eine Ressource für das Deutsche ist aber nur sehr schwer zu bekommen. Prinzipiell kommt, um ein prominentes Beispiel zu nennen, der Langenscheidt Verlag[6] als Firma mit einem

[5] Unter Zeichenkodierung (encoding) versteht man die Speicherung eines Zeichens eines beliebigen Zeichensatzes als eine bestimmte Abfolge von Werten (meist Zahlen). Diese Werte werden über ein bestimmtes encoding als das Zeichen repräsentiert, welches in der Kodierungstabelle dieses encodings vorliegt. Bei unterschiedlichen encodings ist die korrekte Zeichenwiedergabe nicht garantiert.
[6] Langenscheidt KG (`http://www.langenscheidt.de/`)

hervorragenden bilingualen Wörterbuch als Kooperationspartner in Frage.
Des weiteren gibt es aber auch lizenzrechtlich weniger bedenkliche Alternativen. Hier sei lediglich auf *Wiktionary*[7] als ein Beispiel verwiesen.

Da der Heidelberger Computerlinguistik bereits ein gecrawltes[8] Online-Wörterbuch von dict.cc[9] mit fast 700.000 Einträgen zur Verfügung steht, wurden oben erwähnte Möglichkeiten nicht weiter verfolgt. Dieses Wörterbuch wurde jedoch erst nach einer Bereinigung mittels regulärer Ausdrücke und einem *recode*[10] auf utf-8 nutzbar, wie es im Kapitel 3.1 näher beschrieben ist.

Um eine größere Abdeckung zu erreichen, wurde das unter der GPL[11] stehende deutsch-englische Wörterbuch von Linguee[12] dem bereinigten dict.cc-Wörterbuch hinzugefügt. Die Wortliste von Linguee umfasst 51.575 Einträge (Stand: Juli 2011).

Somit sind beide Wörterbücher zusammen sehr gut für die Zwecke vorliegender Arbeit geeignet.

2.2.3 Lemmatisierer

Die deutsche Sprache lässt sich mehr als die englische Sprache in die Richtung der lexozentrischen Sprachen einordnen, da jene eine komplexere Morphologie aufweist (vgl. (Falk, 2001)). Die Rückführung eines flektierten oder deklinierten deutschen Wortes auf seine Grundform ist deshalb keine triviale Aufgabe.

Eine Möglichkeit der Lemmatisierung ist die Benutzung einer komplexen Software.

[7] siehe `http://www.wiktionary.org/`
[8] Ein *crawler* ist in diesem Zusammenhang ein Programm, das auf eine Internetseite (hier: dict.cc) angesetzt wird, mit dem Ziel deren online Angebot abzugreifen und auf ein eigenes Speichermedium zu übertragen.
[9] siehe `http://www.dict.cc/`
[10] recode konvertiert Dateien zwischen diversen Zeichensätzen und -formaten. (vgl. man page von recode)
[11] GPL steht für GNU General Public License. Diese ist ausführlich unter `http://www.gnu.org/copyleft/gpl.html` (Stand: Juli 2011) erklärt.
[12] Die Gründer von Linguee GmbH (`http://www.linguee.de`) haben einen Crawler geschrieben, der das Web ständig nach bilingualen Ressourcen durchsucht und mittels eines ausgefeilten Machine-Learning-Algorithmus parallele Sätze extrahiert. (vgl. http://www.linguee.de/deutsch-englisch/page/about.php).

XFST[13] oder das frei verfügbare Morphisto[14] sind zwei Beispiele dafür. Diese erfordern aber ein gewisses Know-how und einen schwer abschätzbaren zeitlichen Aufwand, um neben oder innerhalb des eigenen Systems lauffähig zu sein.

Eine zweite Möglichkeit liegt in der Benutzung des tree-taggers[15]. Dieser taggt nicht nur, sondern lemmatisiert auch relativ schnell, arbeitet allerdings mit einem statistisch trainierten Sprachmodell und erkennt nicht alle Wörter oder Lemma-Formen korrekt. Dennoch wäre der tree-tagger eine gute Wahl, da die Fehlerrate doch relativ gering ist und eine robuste und effiziente Verarbeitung gewährleistet wäre.

Nach einer Internetrecherche tat sich eine interessante dritte Möglichkeit auf. Dank des gefundenen Vollformenlexikons vom Wortschatz Universität Leipzig[16] mit der Abbildung auf die möglichen Lemmata für die deutsche Sprache, wurde auf die Einbindung obiger Tools bzw. Software verzichtet, da dritte Möglichkeit die performanteste und komfortabelste ist. Das Vollformenlexikon umfasst 944.383 Vollformen-Einträge.

2.2.4 NLTK

Ob mehrere hintereinander vorkommende Wörter in einem Text zusammengehören, sogenannte Wortverbindungen, ist für vorliegenden Ansatz von Belang. Wenn mehrere solcher *compound words* in einem deutschen Textabschnitt mit seinen englischen Übersetzungen in einem englischen Textabschnitt zu finden sind, so sollte die Wahrscheinlichkeit einer inhaltlichen Schnittmenge beider Textabschnitte höher sein, als wenn sich lediglich einzelne Wörter gleicher Anzahl entsprechen. Das liegt einfach daran, dass Wortverbindungen insgesamt seltener vorkommen als Einzelwörter.

Eine inhaltliche Schnittmenge zweier Textstellen besteht auf der Wortebene aus den gemeinsamen inhaltstragenden Wörtern. Funktionswörter und andere inhaltsleere Wörter sollten daher bei einem Vergleich der beiden Textstellen nicht berücksichtigt werden. Diese

[13] siehe `http://www.cis.upenn.edu/~cis639/docs/xfst.html`
[14] siehe `http://code.google.com/p/morphisto/`
[15] siehe dazu die Projekt-Übersichtsseite `http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/DecisionTreeTagger.html`
[16] siehe `http://wortschatz.uni-leipzig.de/`. Das Herunterladen der Lemmata-List ist über diesen Link nicht möglich. In Google Code wurde dafür folgender Link gefunden: `http://code.google.com/p/nltk/issues/detail?id=604`

sind nämlich nicht nur inhaltsleer, sondern auch hochfrequent in Texten unterschiedlicher Domänen zu finden.

Um *compound words* verarbeiten zu können und die später untersuchten Absätze von inhaltsleeren Wörtern zu befreien, wird auf das Natural Language Toolkit (NLTK, siehe `http://www.nltk.org/`) zurückgegriffen. NLTK ist eine Sammlung von Tools zur maschinellen Verarbeitung natürlicher Sprache.

Die Erstellung von N-Grammen[17] ist in NLTK als einfach aufzurufende Methode verfügbar. Diese sollen die Mehrwortausdrücke in den Griff bekommen, indem die entstehenden Zerlegungen im Wörterbuch nachgeschlagen werden. Wenn diese Zerlegungen im Wörterbuch vorhanden sind, sind es Mehrwortausdrücke.

Die Bereinigung von inhaltsleeren Wörtern wird über eine Stoppwortfilterung betrieben. Wörter in zu untersuchenden Textabschnitten werden dazu mit der von NLTK bereitgestellten Stoppwortliste abgeglichen. Ist ein Abgleich erfolgreich, wird das gerade abgeglichene Wort von der weiteren Analyse ausgeschlossen, beispielsweise einfach gelöscht.

2.2.5 Tokenisierer

Um auf der Wortebene agieren zu können (z.B. das Nachschlagen eines Wortes im Wörterbuch), müssen diese Wörter auch zugänglich sein. Tokenisierung ist die Zerlegung eines Textes in seine *token*. *Token* sollen hier als die im Text vorkommenden Wörter definiert sein (vgl. (Carstensen et al., 2004, 406)).

Diese Wortsegmentierung ist keine allzu schwere Aufgabe und kann auf verschiedene Weisen bewerkstelligt werden. Es wurde hier auf das Tokenisierungsskript des vielfach bewährten und bereits erwähnten[18] tree-taggers zurückgegriffen. Dieser kann nicht nur Wörter taggen und lemmatisieren, sondern auch tokenisieren. Das Perl-Skript kann

[17] Ein N-Gramm ist das Ergebnis einer Zerlegung eines Textes in zusammengehörige Tokens, wobei N die Anzahl der Elemente angibt, welche zusammengehören sollen. So kann beispielsweise "Hans überredet Maria zu kommen." in Bigramme zerlegt werden, welche den Text in zwei zusammengehörige Wortfolgen zerlegt. Das Resultat kann in 2-Tupeln sehr gut dargestellt werden: ("Hans", "überredet"), ("überredet", "Maria"), ("Maria", "zu") und ("zu", "kommen") (vgl. (Jurafsky and Martin, 2000).

[18] siehe Kap. 2.2.3

einfach als externe Ressource eingebunden werden und verarbeitet übergebene Texte ohne Verzögerung.

3 Vorverarbeitung

Alle notwendigen Ressourcen stehen zur Verfügung. Nutzbar sind sie in dieser Form allerdings noch nicht. Vor allem das dict.cc-Wörterbuch muss bereinigt werden und die Artikel in der deutschsprachigen Wikipedia mit den Artikeln in der englischsprachigen in Verbindung gebracht werden, um ein artikelaliniertes Korpus zu erhalten. Die Suche nach vergleichbaren Absätzen soll sinnvollerweise auf vergleichbaren Artikeln verlaufen, da dort die Chance am größten ist, solche zu finden.

3.1 Aufbereitung des gecrawlten dict.cc Wörterbuches

Das unbereinigte Wörterbuch dict.cc, wie es den Heidelberger Computerlinguisten als Ressource vorliegt, wurde weder in iso-8859-1(5) noch in utf-8 abgespeichert. Einige Sonderzeichen, vor allem aber die deutschen Umlaute, würden so für die weitere Verarbeitung nicht weiter nutzbar sein. Diesem Problem wurde mit *recode* entgegnet, mit der Angabe die Textdatei von iso-8859-15 nach utf-8 zu konvertieren. Das Ergebnis war zufriedenstellend, die deutschen Umlaute bekamen eine valide Repräsentation, lediglich sehr wenige Zeichen konnten nicht wiederhergestellt werden.

Desweiteren ist das Wörterbuch so aufgebaut, dass jedes zu übersetzende Wort, Wortverbindung oder gar ganze Sätze mit einem eindeutigen Trennungszeichen von der englischen Übersetzung getrennt in einer einzelnen Zeile steht. Das ist prinzipiell eine gute Möglichkeit der Darstellung. Genusinformationen, Informationen über die Verwendungsweise, Numerusinformationen, alternative Übersetzungen und vieles mehr in ambigen Klammernotationen machte es notwendig, folgende problematische Zeichen und deren Inhalt, falls vorhanden, zu entfernen: ’’’ “ { } [ ] ( ) & .

Klammerung	deutscher Eintrag		englische Übersetzung
[...]	(Australische) Schwarze Witwe {f} [Giftspinne]	::	red-back spider [Aus.] [Latrodectus hasselti]
	(die) Werbung unterdrücken [bei aufgezeichneten Fernsehsendungen]	::	to zap the ads [Am.] [coll.]
{ ... }	(anglikanischer) Pfarrer {m}	::	rector [Br.]
	(chemische) Verbindungsgruppen {pl}	::	(chemical) compound groups
(...)	(dreiblättriges) Kleeblatt {n}	::	trefoil
	Abhören {n}	::	wiretap (activity)
< ... >	(der) Irak m	::	Iraq <.iq>

Abbildung 3.1: Beispiele problematischer Klammerungen im dict.cc Wörterbuch

In obiger Tabelle sind problematische Klammerungen, wie sie im Wörterbuch vorkommen, auszugsweise dargestellt. Eckige Klammern denotieren in obigem Fall eine Spinnenklassifikation, den lateinischen Namen dieser, eine zusätzliche Anmerkung oder eine sprachspezifische Angabe.
Geschweifte Klammern können Numerus- oder Geschlechtsangaben repräsentieren.
Ob es sich bei *wiretap* um eine Aktivität handelt oder doch *trefoil* noch spezifischer übersetzt werden kann, ist in runden Klammern formuliert.
Die Domainnamen-Angaben in spitzen Klammern für ein bestimmtes Land wurden auch bereinigt.

3.2 Einbindung des Wörterbuches von Linguee

Das zugrunde liegende Format des (bereinigten) dict.cc Wörterbuchs entspricht genau dem Format des Linguee-Wörterbuches. Dieses konnte einfach der Text-Datei ohne weitere Verarbeitung hinzugefügt werden. Somit können beide Wörterbücher gleichzeitig als Streuwerttabelle[1] eingelesen werden.

3.3 Einbindung der Lemmata-Liste

Die Liste der Lemmata konnte ebenso unproblematisch übernommen werden. Auch hier wurde ein einfach einzulesendes Format gewählt. Die deutschen Vollformen stehen

[1] besser bekannt als hashmap (Java) oder dictionary (Python)

mit einer zugehörigen Lemma-Form in einer Zeile und beide sind durch ein eindeutiges Trennungszeichen voneinander getrennt. Hat eine Vollform mehrere Möglichkeiten einem Lemma zugeordnet zu werden, so stehen weitere Möglichkeiten jeweils in einer neuen Zeile, die deutsche Vollform taucht somit mehrmals in der Liste auf.

3.4 Erstellung eines artikelalinierten Korpus

Um zwei vergleichbare Absätze in verschiedenen Sprachen in Wikipedia zu finden, sucht man am Besten in den zueinander passenden Artikeln beider Sprachen. Dafür benötigt man ein artikelaliniertes Korpus.

Vorliegende bereinigte Artikelsammlungen aus Wikipedia in deutscher und englischer Sprache haben keine Informationen über die Verbindung zu Artikeln der anderen Sprachen. Diese musste also noch hergestellt werden. Encoding-Probleme waren zusätzlich vorhanden.

3.4.1 Encoding-Probleme

Wie bei den Ressourcen zuvor, wurden auch hier die beiden Dateien in utf-8 rekodiert. Die beim Erstellen des Korpus durch Unterstriche ersetzten Symbole anderer Sprachen konnten damit nicht wieder hergestellt werden, wie folgendes Schaubild illustrieren soll:

Soll:	Ein Alphabet (über altgriechisch ἀλφάβητος alphábētos) ist eine geordnete ...
Ist:	Ein Alphabet (über altgriechisch _____ alpháb_tos) ist eine geordnete ...

Abbildung 3.2: Vermisste Einträge durch encoding Probleme im IMS-Wikipedia

Ob ein Fremdwort in einem gegenübergestellten Kandidatenabsatz wieder aufgegriffen wird, wäre ohne encoding-Probleme sicherlich ein zuverlässiges Feature.

3.4.2 Artikelalinierung

Derzeit liegen zwei Dateien vor, welche keine Zuordnung der Artikel in deutscher Sprache mit denen in englischer Sprache oder umgekehrt bieten.
Um eine Zuordnung herzustellen, wurde ein aktueller Wikipedia-*dump* für die deutsche

Wikipedia heruntergeladen. Anschließend wurden die Artikelbezeichnungen der bereinigten deutschen Wikipedia-Datei extrahiert. Jede Artikelbezeichnung wurde in dem neu heruntergeladenen Abbild gesucht. Sobald der gleiche Artikel gefunden wurde, konnte mittels eines regulären Ausdrucks der *interlanguage link* für die englische Wikipedia entnommen werden. Dieser Hyperlink verweist bereits auf den Namen der vergleichbaren, englischsprachigen Seite. Die Ergebnisse wurden in einer Datei zwischengespeichert, welche eine mögliche Alinierung darstellen.

Trotz des Umstandes, dass sich Artikelnamen im Laufe der Zeit in Wikipedia ändern oder Artikel gänzlich gelöscht werden können und durch bereits erwähnte encoding-Probleme, die auch in den Artikelnamen problematisch sind, konnten immerhin 669.972 Artikelnamen von ursprünglich 781.564 deutschen Artikelnamen erfolgreich in der Wikipedia-Abbilddatei, wie oben beschrieben, gefunden werden und mit den englischen Artikelbezeichnungen der heruntergeladenen Abbilddatei in Verbindung gebracht werden.

Obiger Umstand betrifft natürlich auch die dem Computerlinguistik-Institut in Heidelberg vorliegende bereinigte englische Wikipedia-Datei. Letztendlich konnten von diesen 669.972 möglichen Zuweisungen 343.273 erfolgreich hergestellt werden. Das ist zwar etwas weniger als die Hälfte der Ursprungsgröße des deutschen vorliegenden Korpus, aber für vorliegende Zwecke eine mehr als ausreichende Textsammlung.

3.4.3 Erstellung eines neuen Wikipedia-Korpus

Die beiden zuvor seperaten Sprachdateien konnten durch die gewonnenen mapping-Informationen in einem weiteren Schritt zu einer XML-Datei[2] zusammengefasst werden. Die Absätze konnten auf einfache Weise extrahiert werden, da diese mit zwei Zeilenabständen hintereinander realisiert waren.

Bereits hier wurde eine neue Ressource geschaffen: Ein artikelaliniertes vergleichbares Korpus, bei dem die einzelnen Absätze mit fortlaufender Nummerierung vorliegen.
Valide ist diese XML-Datei nicht, dazu müssten jedoch lediglich die Sonderzeichen (z.B. durch ein Skript) ersetzt werden oder mittels einer DTD oder einem CDATA-Block

[2] zu den Vorteilen von XML siehe (Carstensen et al., 2004, 409)

spezifiziert werden.

Zur Veranschaulichung soll folgende XML-Struktur, welche der Ressource zu Grunde liegt, hier skizziert sein:

```
<?xml version='1.0' encoding='UTF-8' 'standalone=yes' ?>
<wikipedia>
  <article id='1'>
    <german name='...'>
      <paragraph number='1'>
        ...
      </paragraph>
      <paragraph number='...'>
        ...
      </paragraph>
    </german>
    <english name='...'>
      <paragraph number='1'>
        ...
      </paragraph>
      <paragraph number='...'>
        ...
      </paragraph>
    </english>
  </article>
  <article id='...'>
    ...
  </article>
</wikipedia>
```

Abbildung 3.3: XML-Struktur für das neu erstellte artikelalinierte Korpus

Die Vorraussetzungen für den vorliegenden Ansatz sind hiermit erfüllt.

4 Ansatz

Bislang wurde dargelegt, was die eigentliche Forschungsfrage ist - nämlich wie miteinander vergleichbare Absätze identifiziert werden können (und ob überhaupt) -, welche ähnlichen Ansätze es dazu bereits gab und welche Ressourcen am Besten für diesen Ansatz ausgewählt werden sollten und warum.

In diesem Kapitel soll der in dieser Arbeit beleuchtete Ansatz vorgestellt werden. Dazu gibt es zuerst eine Systemübersicht, um zu zeigen, was der Ansatz überhaupt leisten soll. Nachfolgend wird erklärt, wie man von der Artikelebene auf die Absatzebene kommt, um überhaupt Absatzkandidaten finden zu können, welche man miteinander vergleichen kann.
Nachdem die einzelnen Absätze gegenübergestellt sind, werden diese anhand einiger Features untersucht. Die Methoden, welche die Absätze nach gewissen Eigenschaften hin untersuchen, bewegen sich auf der Wortebene.

Insgesamt wird hier kein bidirektionaler Vergleich vorgenommen. Das heißt, es wird lediglich geprüft, ob es sich bei einem englischen um einen vergleichbaren Absatz zum untersuchten deutschen handelt, aber nicht umgekehrt.[1]

4.1 Überblick

Wie in Abb. 4.1 ersichtlich ist, werden gegenübergestellte Absätze in Artikeln aus Wikipedia mit Hilfe gängiger Features untersucht und nach einer erfolgreichen Klassifikation in ein vergleichbares Korpus extrahiert.

[1] Ein bidirektionaler Vergleich hätte den zeitlichen Rahmen gesprengt.

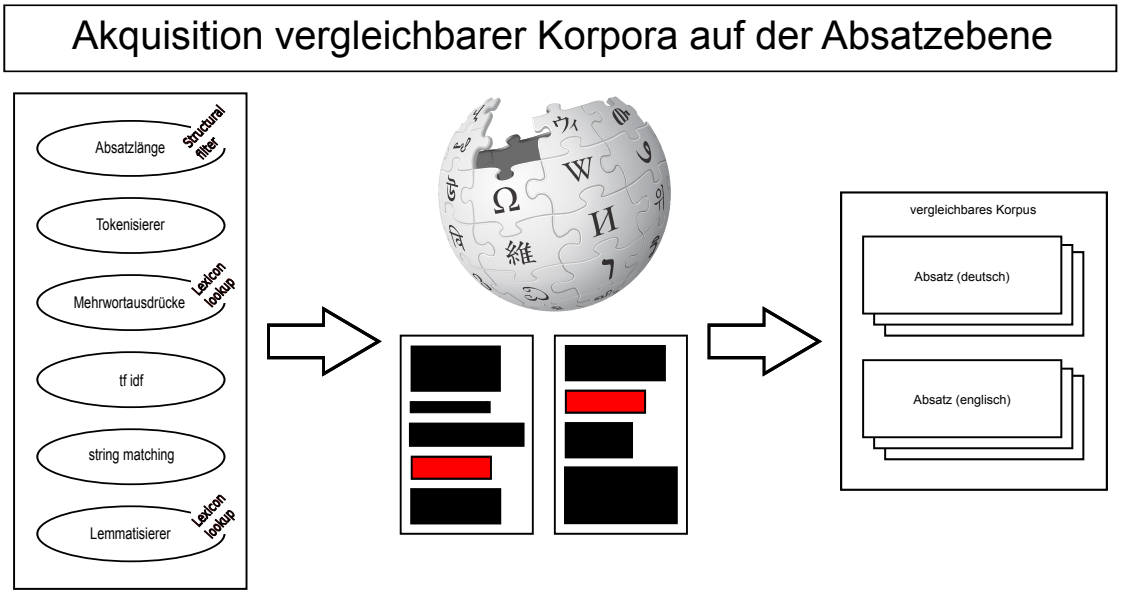

Abbildung 4.1: Ausgangslage, Methoden und Ziel des Algorithmus

In der Mitte sind beispielhaft zwei zueinander gehörende Wikipedia-Artikel dargestellt. Die linke davon soll den deutschen Artikel mit all seinen Absätzen repräsentieren, die rechte den dazu vergleichbaren englischen mit dessen Absätzen. Das entspricht genau dem derzeitigen Stand und skizziert die derzeitige Ausgangslage.
Die in hellerer Farbe markierten Absätze sollen vergleichbare Absätze sein. Ziel des Ansatzes ist es, dass diese als vergleichbar identifizert werden.
Dazu sollen die Features der Absätze mit den ganz links dargestellten Methoden untersucht werden.
Wenn eine erfolgreiche Identifikation stattgefunden hat, werden die beiden Absätze aliniert und in einem neuen Korpus gesammelt.

Bevor allerdings die Absätze untersucht werden können, müssen diese zunächst gegenübergestellt werden.

4.2 Kandidatenpaare finden

Ausgehend von zwei miteinander vergleichbaren Artikeln, der eine in deutsch und der andere in englisch verfasst, werden vergleichbare Kandidatenpaare, ein Absatz in der

deutschen Variante und der andere in der englischen, gebildet. Dazu müssen alle deutschen Absätze des derzeitigen Artikels mit englischen Absätzen im vergleichbaren Artikel gegenübergestellt werden.[2]

Die für einen Artikel geltende Menge aus Tupeln (vgl. Abb. 4.2) sind die bereits erwähnten Kandidatenpaare, weil noch keine Aussage darüber getroffen werden kann, ob diese miteinander vergleichbar sind, und es sich lediglich um eine Gegenüberstellung, eine mögliche Alinierung handelt.

$$(\texttt{german\_paragraph}_i\ ,\ \texttt{english\_paragraph}_j)$$

Abbildung 4.2: ein Kandidatenpaar

Für jeden Artikel gilt: Es gibt i deutsche Absätze und j mögliche Alinierungen mit einem englischen Absatz.

In der vorliegenden XML-Datei (siehe Abb. 3.3) ist jedem deutschen Artikel die englische Entsprechung zugeteilt. Dadurch ist es möglich, über die jeweiligen Artikel zu iterieren und anschließend über jeden deutschen Absatz, um Kandidatenpaare zu erhalten.

Um vergleichbare Absätze zu finden, könnte man jeden Absatz mit jedem gegenüberstellen und diese dann untersuchen. Dieses Kreuzprodukt ist allerdings weder performant, noch linguistisch motiviert. Deshalb ist eine Beschränkung der Kandidatenauswahl sehr sinvoll.

4.2.1 Beschränkung der Kandidatenauswahl

Nachfolgende Formel (4.1) stützt sich auf die Belegung der vernünftigen These, dass vergleichbare Absätze in beiden Textstellen an ungefähr der gleichen (relativen) Position zu finden sein sollten(vgl. (Gale and Church, 1993)).

[2] Zur Erinnerung: Es wird hier kein bidirektionaler Vergleich vorgenommen. Prinzipiell ist demnach auch Deutsch als Ziel- und nicht als Quellsprache möglich.

$$\forall n : \quad \frac{n}{i} \cdot j \quad \approx \quad m_{\pm 2} \tag{4.1}$$

```
n  =  Absatzposition Quellsprache
m  =  Absatzposition Zielsprache
i  =  Anzahl an Absätzen in der Quellsprache
j  =  Anzahl an Absätzen in der Zielsprache
```

Jedem deutschen Absatz werden maximal fünf mögliche englische Absätze gegenübergestellt. Der möglichen unterschiedlichen Absatzanzahl in Quell- und Zielsprache wurde durch einer Normalisierung entgegengetreten.

Ein angenehmer Seiteneffekt tritt dabei auf: Die Performanz des Systems ist besser, da eben nicht mehr jeder deutsche Absatz mit jedem englischen Absatz (n×m) untersucht werden muss.

4.2.2 Filtern von Absatzpaaren

Desweiteren wurde ein heuristischer Wert gesetzt, um Kandidaten zu filtern, deren Längen sehr verschieden voneinander sind. Ein Kriterium für Vergleichbarkeit ist mitunter eine nicht allzu stark abweichende Spezifität. Hier wird angenommen, dass die Spezifität zunimmt, je länger ein Absatz eines Kandidatenpaares im Vergleich zum anderen ist.

Dieses Vorgehen stützt sich auf (Gale and Church, 1993). Sie haben Sätze gesucht, die parallel zueinander sind. Diese erhofften sie in Absätzen von zwei bilingualen Texten zu finden, deren Längen nicht zu divergent sind. Sie haben dabei die Anzahl der Zeichen, nicht etwa die Anzahl der Wörter, als Längenmaß verwendet.

Im vorliegenden Ansatz bedeutet Länge eines Absatzes, die Anzahl der darin enthaltenen Wörter. Strukturelle Eigenschaften von Absätzen sollen hier lediglich als weicher Filter eingesetzt werden, nicht als hartes Kriterium für Vergleichbarkeit dienen. Bei der Findung von parallelen Absätzen könnte diese Entscheidung anders ausfallen, hier fallen Spezifitätsunterschiede in den zu vergleichenden Absätzen nicht so stark ins Gewicht. Es werden demnach lediglich die Kandidatenpaare nicht untersucht, deren englischsprachige Absätze mehr als doppelt so lang sind wie die deutschen. Dieser Wert wurde heuristisch bestimmt.

4.3 Kandidatenpaare untersuchen

Die gefundenen Alinierungskandidaten, ein deutscher Absatz bekommt 1-5 englische Absätze zugeteilt (vgl. Formel (4.1)), können nun untersucht werden.

Dabei werden die Methoden wie Tokenisierung, Mehrwortausdrücke erkennen, tf·idf-Wert eines Wortes berechnen, usw. (vgl. Abb. 4.1) auf die Kandidatenabsätze angewandt. Jedes dieser Feature steuert mit Hilfe der erwähnten Methoden durch eine positive und negative Punktevergabe dazu bei, einen Absatz eher als vergleichbar oder eher nicht vergleichbar einzustufen.

Am Ende entscheidet die Summe der ermittelten Feature-Werte über die binäre Klassifikationsentscheidung. Dazu wird die Endsumme auf die Anzahl der untersuchten Wörter normalisiert, so dass ein Wert zwischen 0 und 100 möglich ist.

Die Optimierung dieses Parameters (dieses Wertes) wird in Kap. 4.6.2 vorgenommen, im Folgenden wird auf die einzelnen Features und Methoden eingegangen.

4.3.1 Tokenisierer

Wie bereits in Kap. 2.2.5 erwähnt, wird hier der Tokenisierer des tree-taggers benutzt. Dieser arbeitet schnell und zuverlässig und ist zudem bequem einzubinden.

Eine Tokenisierung ist notwendig, um eine Segementierung der Wörter vorliegen zu haben. Diese Wörter können dann einzeln im Wörterbuch nachgeschlagen werden. Ein Absatz besteht jetzt nicht mehr aus einer Anreihung von Zeichenketten, sondern aus einer Liste von Wörtern.

An dieser Stelle werden noch beide Absätze des jeweiligen Kandidatenpaares (d.h. der deutsche und der englische Absatz) tokenisiert, da ja sowohl auf einzelne deutsche, als auch auf einzelne englische Wörter zugegriffen werden können sollte.

Für die weiteren Untersuchungen liegen also nun die Absätze in deutsch und englisch tokenisiert vor. Im Folgenden werden allerdings nur noch die deutschen Absätze unter-

sucht und anschließend mittels eines Wörterbuchabgleichs mit den englischen Absätzen verglichen.

4.3.2 Mehrworterkennung

Die Erkennung von Mehrwortausdrücken ist sehr simpel, da sie sich lediglich auf die relevanten Vorkommnisse beschränkt. Relevante Vorkommen sind solche Ausdrücke, die nur übersetzt werden können, wenn diese auch im Wörterbuch enthalten sind.
Kommen diese darin vor, d.h. ein deutscher Wörterbucheintrag besteht aus mehr als einem Wort, dann ist das zumindest eine wichtige Kollokation.

Zur Illustration sei folgende Tabelle dargestellt:

Wortanzahl	**deutscher Eintrag**		**englische Übersetzung**
2	fahrendes Volk	::	travellers
	im Kampfgetümmel	::	in the thick of the fight
3	Stützen der Gesellschaft	::	Pillars of Society
	zur Verschiffung erhalten	::	received for shipment
4	Abschaffung von beschränkenden Bestimmungen	::	deregulation
	Zunahme des kombinierten Transports	::	extension of combined transport
4+	Abschätzung der Strahlenexposition und des Strahlenrisikos	::	estimation of radiation exposure and radiation risk
	'Nen interessanten Job habt ihr Jungs da.	::	Interesting job you guys have.

Abbildung 4.3: Mehrwortausdrücke im bereinigten dict.cc

Da die Wahrscheinlichkeit für jeden um ein Wort vergrößerten Mehrwortausdruck stark abnimmt, die passende Übersetzung im gegenübergestellten Absatz zu finden und die Wahrscheinlichkeit zunimmt, jedes hinzukommende Wort durch einen einfachen Lexikonabgleich zu 'erwischen', wurde pauschal eine relativ hohe Grenze von fünf Wörtern für das Finden von Mehrwortausdrücken angenommen.

Es wird dabei so vorgegangen, dass der deutsche Absatz in N-Gramme zerlegt wird, wobei das N für eine Zahl von zwei bis fünf steht. Die gefundenenen 2-, 3-, 4- oder 5-Tupel werden nun im Wörterbuch nachgeschlagen. 1-Gramme, also Unigramme, sind die regulären Wörter und werden von anderen Methoden untersucht.

Dieses Feature soll im Vergleich zu den anderen relativ hoch gewichtet werden, da angenommen wird, dass bei mehr passenden Mehrwortausdrücken eine höhere Ähnlichkeit vorherrscht.[3] Daher erhält jede gefundene Wortkombination einen vollen Punkt.

Am Ende werden die Punkte dieses Features auf die Anzahl der möglichen Wortkombinationen normalisiert und dem gerade untersuchten Kandidatenpaar gutgeschrieben.

4.3.3 Stoppwortfilterung

Die Stoppwortliste soll die inhaltslosen Wörter und Funktionswörter herausfiltern, damit nur die aussagekräftigen Inhaltswörter weiterhin untersucht werden.

Diese Filterung findet erst nach der Suche nach Mehrwortausdrücken ihren Platz, da bei den Mehrwortausdrücken Stoppwörter noch gewollt sind (z.B. *in Gefahr* oder *nach Ablauf der Garantie*, usw.).

Ansonsten gilt, dass Stoppwörter hochfrequent und domänenunabhängig vorkommen und somit einem Punkte-System diametral entgegen stehen. Ein Absatzpaar soll nicht dann als vergleichbar klassifiziert werden, wenn zufällig die gleichen Präpositionen und viele Artikel im Kandidatenpaar auftauchen.

Nachdem diese Filterung zur Geltung kam, besteht das derzeit untersuchte Absatzpaar aus einem stoppwortgefilterten, tokenisierten deutschen Absatz und einem lediglich tokenisierten englischen Absatz. Der deutsche Absatz wird im Folgenden weiter untersucht, während der englische auf seinen Abgleich wartet.

[3] Im Information Retrieval beispielsweise ist diese Annahme sehr ersichtlich. Dies kann sogar in einem Eigenversuch selbst nachvollzogen werden: Man gebe einen Suchterm in eine Suchmaschine ein, welche die Suche von Wortverbindungen erlaubt. Diesen Term erweitert man sukkzessive um weitere Wörter, bis man letztendlich keine Dokumente mehr zurückbekommt, da der Ausdruck nicht mehr gefunden werden kann. Ersichtlich sollte dabei werden, dass die Dokumentmenge ebenfalls sukzessive eingeschränkt wird und immer mehr unpassendere Dokumente verworfen werden.

4.3.4 Wortgewichtung mit tfidf

Eine Vergabe von Punkten für jedes Wort ist nicht sehr sinnvoll, wenn jedes Wort als gleich wichtig angesehen wird und die gleichen Punkte zugeschrieben bekommt. Ein adäquates Mittel zur Gewichtung von Wörtern kommt aus dem Information Retrieval, das sogenannte tf·idf-Maß. Wie wichtig ein Wort in einem Dokument gegeben einer zugehörigen Dokumentmenge ist, kann mit diesem Maß statistisch ermittelt werden.

Das Termfrequenz-Maß, die linke Seite der Formel (4.2), ist an (Manning and Schütze, 1999, 541ff) angelehnt, die Bestimmung der inversen Dokumentfrequenz, die rechte Seite der genannten Formel, ist derart beispielsweise in (Apache Lucene) modelliert.

Sei das Dokument der derzeit betrachtete deutschsprachige Absatz und die Dokumentmenge die Menge aller Absätze des gerade betrachteten Artikels in der gleichen Sprache, so ergibt sich durch Anwendung der Formel (4.2) eine Gewichtung für jedes Wort in jenem gerade betrachteten deutschen Absatz.

$$\frac{\texttt{word frequency}}{\texttt{number of words in paragraph}} \cdot (1 + \log(\frac{\texttt{number of paragraphs}}{\texttt{number of paragraphs with word inside} + 1})) \quad (4.2)$$

So bekommt ein Wort eine höhere Gewichtung, wenn es in allen Absätzen eher selten vorkommt, aber verbreitet in einigen Absätzen.

Außerdem werden die tf·idf-Gewichtungen der einzelnen Wörter in Relation zu dem besten Ergebnis des momentan betrachteten deutschen Absatzes gesetzt:

$$\omega = \frac{\texttt{tfidf}_\texttt{w}}{\max(\texttt{tfidf}_\texttt{v})} \quad (4.3)$$

Somit erhält jedes Wort w eine endgültige Gewichtung ω im Wertebereich 0-1.

Die deutschen Wörter eines Absatzes bestehen nun aus dem jeweiligen Wort selbst mit seinem relativen Gewicht. Der englische Absatz ist immer noch lediglich tokenisiert und bleibt das auch. Die jeweiligen deutschen Wörter werden nun iterativ den folgenden drei Methoden übergeben, damit diese einen möglichen Abgleich finden und das deutsch-englische Absatzpaar bewerten können.

4.3.5 String matching

String matching, das heißt ein Wort der Quellsprache wird in der gleichen Zeichenkettenkonkatenation auch in der Zielsprache repräsentiert, stellt ein einfaches, aber auch wichtiges Verfahren dar, um Eigennamen oder gar Named Entities zu erkennen, aber auch Wörter, die gar nicht übersetzt werden müssen, da sie bereits aus der Zielsprache kommen.

Named Entity	**deutscher Eintrag**	**englische Übersetzung**
Städte	Im Jahr 1954 zog Snedden nach **Melbourne**, wo er ...	In 1954 he moved to **Melbourne**, where he ...
Eigennamen	Im Jahr 1971 wurde **Snedden** durch William McMahon zum Finanzminister ernannt ...	In 1971, however, **Snedden** was appointed Treasurer by William McMahon ...
Zahlen	Im Jahr **1982** bekam Sedden die Möglichkeit zur Rache an Fraser.	In **1982** he had revenge of sorts on Fraser ...
Institutionen	Das **Eller College of Management** ist die Business School der ...	The **Eller College of Management** is a top-ranked business school ...

Abbildung 4.4: Einfaches *string matching*

Obige Tabelle veranschaulicht den Vorteil von *string matching*. Es sind keine raffinierten Tools notwendig, ein einfacher Abgleich beider Absätze bringt bereits Ergebnisse. Zur Erinnerung: An dieser Stelle existieren keine ganzen Sätze mehr, sondern tokenisierte und stoppwortgefilterte Wörter. Das Wort *of* in *Eller College of Management* taucht nicht mehr auf. Es wird hier also sowohl *Eller*, als auch *College*, sowie *Management* erkannt. Dabei spielt auch der jeweilige Klassifikationstyp der Named Entity keine Rolle und kann getrost ignoriert werden.

Falls ein *string matching* für ein Wort vorliegt, wird den Gesamtpunkten des deutsch-englischen Absatzpaares der relativ gewichtete Wert des Wortes hinzuaddiert. Eine weitere Untersuchung mit anderen Methoden ist für dieses Wort dann hinfällig. Der Abgleich wird in diesem Fall einfach mit dem nächsten Wort fortgesetzt.

4.3.6 Nachschlagen im Wörterbuch

Nur in dem Falle, dass *string matching* für das derzeit betrachtete deutsche Wort kein Ergebnis bringt (das wird meistens der Fall sein), soll ein Abgleich des Wortes mit dem Wörterbuch durchgeführt werden.

Das Wörterbuch bietet eventuell mehrere Übersetzungsvarianten an. Diese werden dann nacheinander als Übersetzungsmöglichkeit vorgeschlagen. Sobald eine englische Übersetzung des deutschen Wortes ein *string matching* liefert, wird der relativ gewichtete Wert des Wortes zur Gesamtpunktzahl des deutsch-englischen Absatzpaares hinzuaddiert und mit dem nächsten Wort weitergemacht. Ansonsten wird das Wort der nächsten Methode übergeben.

4.3.7 Lemmatisierung

Das Lemma-Feature kommt nur dann zum Einsatz, wenn sowohl *string matching* als auch einfaches Nachschlagen im Wörterbuch dem momentan betrachteten Wort keine Punkte zuteil werden lassen konnten. In diesem Fall steht die Vollformen-Liste mit den Abbildungen auf deren möglichen Lemmata bereit.

Das gerade betrachtete *token* wird im Vollformenlexikon nachgeschlagen. Falls es dort zu finden ist, werden die möglichen Lemmata-Formen (z.B. Infinitivform vs. Partizipform) des Wortes in Betracht gezogen und von diesen ausgehend nach einer möglichen Übersetzung Ausschau gehalten. Dies geschieht, wie bereits in Kap. 4.3.6 beschrieben, durch einen *string*-Abgleich des übersetzten - nun in lemmatisierter Form vorliegenden - Wortes im englischsprachigen Zielabsatz.

Auch hier wird bei einem Treffer sofort das nächste Wort zur weiteren Analyse herangezogen, nachdem dem Absatzpaar die gewichteten Punkte des Wortes gutgeschrieben wurden.

Falls das Wort auch hier nicht erkannt wurde, wird es negativ bewertet, wie es in Kap. 4.4 beschrieben wird.

4.4 Kandidatenpaare auswerten

Die guten ins Töpfchen, die schlechten ins Kröpfchen...

Sobald eine zuvor erwähnte Methode ein Wort aus dem deutschsprachigen Absatz mit einem Wort aus dem englischsprachigen Absatz alinieren konnte, wurde den Gesamtpunkten des gerade beleuchteten deutsch-englischen Absatzpaares der relative tf·idf-Wert (siehe Kap. 4.3.4) dieses Wortes hinzuaddiert und mit dem nächsten Wort fortgefahren.

Bekam das Wort durch die Features im englischen gegenübergestellten Absatz keine Alinierung zugeteilt, so wird ebenfalls der relative tf·idf-Wert (vgl. Formel (4.3)) einer zweiten, seperaten Gesamtsumme des Absatzes gutgeschrieben.

Erkannte Mehrwortausdrücke lieferten der Summe zuvor einen ungewichteten vollen Punkt.

4.4.1 Die Errechnung der Gesamtpunktzahl für jedes Absatzpaar

Somit gibt es sowohl eine Gesamtsumme ϕ, welche die gewichtete Summe aller Wörter eines Absatzes definiert, die von den Methoden aliniert werden konnte, als auch eine Gesamtsumme ψ, welche ebenfalls eine gewichtete Summe definiert, allerdings von den Wörtern, denen keine Features zugewiesen werden konnten.

$$\phi = \frac{\sum\limits_{\substack{i=1 \\ \omega \in \gamma_1}}^{n} \omega_i + \sum\limits_{\substack{i=1 \\ \omega \in \gamma_2}}^{n} \omega_i + \sum\limits_{\substack{i=1 \\ \omega \in \gamma_3}}^{n} \omega_i}{\sum\limits_{i}^{n} \omega} \tag{4.4}$$

Obige Formel soll folgenden Sachverhalt veranschaulichen:
Addiere jede Gewichtung eines Wortes für jedes Wort im Absatz, welches mittels der drei Features (γ_1, γ_2, γ_3) aliniert werden konnte, zu der Gesamtpunktzahl ϕ hinzu.

$$\psi = \frac{\sum\limits_{\substack{i=1 \\ \omega_i \notin \gamma_1, \gamma_2, \gamma_3}}^{n} \omega_i}{\sum\limits_{i}^{n} \omega_i} \tag{4.5}$$

Analog zur Formel (4.4) soll diese Formel folgenden Sachverhalt veranschaulichen: Addiere jede Gewichtung eines Wortes für jedes Wort im Absatz, welches mittels der drei Features (γ_1, γ_2, γ_3) nicht aliniert werden konnte, zu der Gesamtpunktzahl ψ hinzu.

4.4.2 Bestimmung des Konfidenzwertes

Für jedes untersuchte deutsch-englische Absatzpaar konnten zwei Werte ermittelt werden. Ein positiver Wert, der angibt, wie gut die Features in ihrer Gesamtheit für das Paar gegriffen haben und ein negativer[4] Wert, der aussagt, inwieweit die einzelnen deutschen Wörter nicht denen im englischsprachigen Absatz entsprechen.

Um nicht weiterhin mit zwei Werten rechnen zu müssen, wurden beide zu einem einzigen Wert vereinfacht, der eine Aussage darüber treffen soll, ob ein Absatz in der Quellsprache letztendlich mit einem Absatz der Zielsprache vergleichbar ist. Dieser Wert soll nachfolgend Konfidenzwert heißen, da damit in letzter Instanz ausgesagt werden soll, inwieweit das System selbst seiner eigenen Alinierung vertraut.

$$\kappa = \frac{\phi}{\phi + \psi} \cdot 100 \tag{4.6}$$

Der Konfidenzwert κ errechnet sich aus den Werten der Formeln (4.4) und (4.5). ϕ und ψ sind die positiven, bzw. negativen Resultate der Feature-Methoden.

An dieser Stelle ist dieser Wert jedoch noch relativ wenig aussagekräftig, da man hier noch nicht weiß, welches der optimale Konfidenzwert ist, um eine binäre Klassifikationsentscheidung, ob die beiden Absätze des beleuchteten Paares aliniert sind oder nicht, treffen zu können. Dafür wird auf das Kapitel 4.6.1 verwiesen, dort wird dieser Wert bestimmt.

4.5 Ranking erstellen

Jedem deutschen Absatz liegen bis zu fünf englische Absatzgegenüberstellungen zu Grunde. Diesen Tupeln wurden Werte zugewiesen. Die Werte geben eine Aussage darüber

[4] Positiv und negativ sind hier im übertragenen Sinne zu verstehen. Mathematisch gesehen, sind beide Werte positiv.

ab, wie sehr einzelne Features auf diese Gegenüberstellung gepasst haben (siehe Kap. 4.4).

Dies entspricht bereits einem *ranking*. Es müssen lediglich die Absätze gemäß ihrem Wert in eine Reihenfolge gebracht werden.

Artikel	**Absatz (de)**	**Absatz (en)**	**Konfidenzwert**
276824	1	1	68.103677740649431
7281	6	5	22.8069683352951
		1	11.917898203006624
32023	5	1	29.475494361226691
		2	14.37966483607131
		3	14.37966483607131
77884	20	2	36.637104518671819
		1	20.720828526239707
		5	8.2464837806952662
		3	3.1744913102934045
174215	2	1	51.876437959600338
		4	42.118154060683459
		2	34.729107563589928
		3	34.729107563589928
		5	34.729107563589928

Abbildung 4.5: *ranking* Ergebnisse

Jedes Kandidatenpaar-Tupel (de-eng) eines Artikels (vgl. 4.2) wurde um den Konfidenzwert (rechts) erweitert. Hier wurde jeweils ein deutscher Absatz innerhalb eines beliebigen Artikels zufällig ausgewählt, um das *ranking* zu veranschaulichen. Ein Artikel hat in der Regel aber mehrere Kandidatenpaare.
Zusätzlich soll hier nochmals veranschaulicht werden, dass einem deutschen Absatz, ein bis fünf englische Kandidatenabsätze entgegenstehen können.
Da durch die XML-Datei (vgl. Abb. 3.3) ein eindeutiges mapping zwischen Absatznummer und Absatzinhalt gewährleistet ist, bestehen die Kandidatentupel lediglich aus den jeweiligen Absatzpositionen in den jeweiligen Sprachbereichen der XML-Datei.

Ob ein deutscher Absatz mit mehreren englischen Absätzen aliniert werden kann, ist eine interessante Fragestellung. Hier wurde diese Problematik nicht aufgegriffen. Diese Fragestellung kann als Future Work aufgefasst werden.

Der Einfachheit halber wird bei gleichem Konfidenzwert so verfahren, dass die englischen Absätze zu dem deutschen Absatz nach ihrer Reihenfolge im Artikel gerankt werden, wie in Artikel 32023 und 174215 der Tabelle in Abb. 4.5 ersichtlich wird.

4.6 Entscheidung über Vergleichbarkeit

Durch das *ranking* ist bislang festgelegt, welche Absätze in den Kandidatenpaaren die beste Chance haben, vergleichbar zu sein. Ob diese das dann tatsächlich sind, ist noch nicht ausgesagt.

Um diese binäre Entscheidung treffen zu können, könnte man heuristisch einen Konfidenzwert annehmen, der als Schwellenwert dienen soll. Liegt ein Kandidatenpaar mit seinem zugeordneten Konfidenzwert (vgl. Abb. 4.5) über dieser Schwelle, dann wird aus dem Kandidatenpaar ein vergleichbares Paar und darf schließlich extrahiert werden. Diese Extraktion kommt dem neu entstehenden Vergleichskorpus zu Gute, welches aus vergleichbaren Absätzen bestehen soll, wie es eingangs schon in Abb. 4.1 skizziert wurde.

Ein Schwellenwert wird hier die binäre Entscheidung treffen, ob ein Kandidatenpaar vergleichbar ist oder nicht. Dieser Wert soll allerdings optimiert sein und nicht heuristisch festgelegt, um an dieser Stelle möglichst viele vergleichbare Absätze zu finden, welche tatsächlich vergleichbar sind und möglichst wenige zu finden, die nicht vergleichbar sind. Dieser Wert, der sich aus den vorliegenden Konfidenzwerten bestimmen lässt, wird am Ende nicht mehr verändert[5] und dient als Klassifikator für Vergleichbarkeit von Absätzen.

Die Parameter-Optimierung ist nur deshalb möglich, weil ein Goldstandard erstellt worden ist. Hier liegen bereits Informationen vor, welche Absätze aus einer Menge von Kandidatenpaare tatsächlich vergleichbar sind. Dieser Mehrwert an Informationen reicht für eine Optimierung.

[5] Um am Ende eine bessere *precision* zu erhalten, könnte man den Wert nach oben korrigieren, für einen besseren recall nach unten.

4.6.1 Parameter-Bestimmung

Der Goldstandard wurde in ein Trainings- und ein Testset aufgeteilt. Das Testset bleibt unangetastet, es soll die Menge an evaluierbaren ungesehenen Daten repräsentieren.
Das Trainigsset wird verwendet, um die errechneten Konfidenzwerte der einzelnen Kandidatenabsätze zu optimieren.

Das Trainingsset kann hierfür, formal gesehen, aus einer Menge von 4-Tupeln[6] dargestellt werden:

$$(\texttt{article}_i\ ,\ \texttt{german\_paragraph}_j\ ,\ \texttt{english\_paragraph}_k\ ,\ \texttt{alignment})$$

Abbildung 4.6: Kandidatenpaare im Trainingsset, Tupeldarstellung

Die erste Position bezeichnet die Artikelnummer, in denen das deutsch-englische Absatzpaar zu finden ist. Jedem deutschen Absatz (Tupel-Position 2) können dabei mehrere englische Absätze (Tupel-Position 3) gegenübergestellt sein. Jede Gegenüberstellung ist mit einem *boolean*-Wert (alignment = True/False) versehen, der eine Alinierung oder Nicht-Alinierung darstellt.

Dies korreliert mit einer möglichen Darstellung der *ranking*-Ergebnisse, welche man erhält, wenn das System die gleichen Daten (also das Trainingsset) ranked.

$$(\texttt{article}_i\ ,\ \texttt{german\_paragraph}_j\ ,\ \texttt{english\_paragraph}_k\ ,\ \texttt{confidence\_value})$$

Abbildung 4.7: gerankte Kandidatenpaare im Trainingsset, Tupeldarstellung

In unterer Abb. 4.7 gibt es ebenso wie in oberer Abb. 4.6 ($i \times j \times k$) Paare. Während es in oberer Abbildung allerdings keine Kandidatenpaare sind, sondern schon klassifizierte Absätze, handelt es sich in unterer Darstellung noch um unklassifizierte Absätze, also um Kandidatenpaare. Diesen sind immer noch die relativ wenig aussagekräftigen Konfidenzwerte zugeteilt.

[6] Diese können zu einem 2-Tupel vereinfacht werden, da die Zusammenführung der ersten drei Positionen zu einer die gleiche Menge bildet, also in etwa: (article$_i$_german_paragraph$_j$_english_paragraph$_k$, alignment).

Um einen optimierten Schwellenwert herauszufinden, der als Klassifikator für die Vergleichbarkeit von Absätzen für die jeweiligen Konfidenzwerte dient, kann man die Tupel aus Abb. 4.6 mit den Tupeln aus Abb. 4.7 direkt vergleichen.

So erhält man für jedes Tupel aus der Trainingsmenge die Information, bei welchem Konfidenzwert des gerade betrachteten Tupels eine Alinierung vorherrscht oder eben keine Zuordnung möglich sein sollte. Gesucht wird derjenige Konfidenzwert, bei dem die meisten Tupel eine richtige Zuordnung erhalten und die wenigsten eine falsche. Dieser Wert ist optimal bezüglich der *true positives* und der *true negatives*[7] und liefert den Schwellenwert.

Dies stellt ein Optimierungsproblem dar. Da jedoch lediglich ein Parameter bestimmt werden muss, kann hier noch durch Ausprobieren aller Konfidenzwerte eine Annäherung an diesen Wert gefunden werden. Die Optimierung des Parameters besteht hier also in der Erstellung einer Wertetabelle.

Diese gibt für jeden *möglichen* Konfidenzwert die Summe an Treffern an. Ein Treffer liegt dann vor, wenn ein Kandidatenpaar aus der Menge aller Kandidatenpaare mit seinem *zugeordneten* Konfidenzwert (aus dem *ranking* Ergebnis) den *möglichen* Konfidenzwert übersteigt und dieser tatsächlich aliniert im Trainingsset vorzufinden ist oder den *möglichen* Konfidenzwert unterschreitet und dieser tatsächlich nicht aliniert im Trainingsset vorzufinden ist.

Diesen etwas komplizierten Sachverhalt soll nachfolgender in der Programmiersprache Python implementierter Algorithmus verdeutlichen. Die Kommentare, welche nach einem beginnenden Raute-Zeichen folgen, sollen dabei dem allgemeinen Verständnis dienen, sodass nicht zwingend Python-Kenntnisse erforderlich sein müssen, um folgenden *code* verstehen zu können:

[7] *true positives* bezeichnen diejenigen Absätze, deren Konfidenzwert den Schwellenwert übersteigen und tatsächlich vergleichbar sind. Die *true negatives* sind diejenigen Absätze, die den Schwellenwert unterschreiten, also vom System nicht als vergleichbar eingestuft werden, und auch tatsächlich keine alinierten Absätze sind.

```
    # initialize an empty data-structure which will be filled later
    resultsDict = {}
    # iterate in 0.0001 steps from 0-100
    for r in map(lambda x: float(x)/10000+1, xrange(990001)):
      # each r gets its own value
      hits = 0
      # iterate over all tuples of the ranking result ...
      for artNum, artVal in rankedDict.iteritems():
        # for each tuple with same article do:
        # iterate over all german paragraphs
        for gerNum, gerVal in artVal.iteritems():
          # for each tuple with same german paragraph do:
          # iterate over the possible candidates
          for engNum, engVal in gerVal.iteritems():
            # for each candidate pair do:
            # get the confidence value of this tuple
            score = float(engVal[0])
            # get the alignment result (True/False) from the training set
            # (this was done by previous methods)
            aligned = engVal[1]
            # if the confidence value is higher than r and is aligned OR
            # if the confidence value is lower than r and is NOT aligned
            if score >= r and aligned or score < r and not aligned:
              # we have a hit, add +1 to the sum of hits for this r
              hits += 1
      # save the final sum of hits for this r.
      # This will create the lookup-table
      resultsDict[r] = hits
```

Abbildung 4.8: Algorithmus zur Bestimmung des optimalen Parameters θ

Nachdem die Wertetabelle nun vorliegt, wird der höchste Wert *hits* ermittelt und alle zugehörigen möglichen Konfidenzwerte r für diesen Wert ausgegeben. Der optimale Parameter θ ist dann der Mittelwert aller ausgegebenen Konfidenzwerte für ein maximales r (Letzteres ist nicht mehr im *code-snippet* ersichtlich).

Eine etwas mehr mathematische Sicht auf dieses Problem kann durch folgende Gleichung dargestellt werden:

$$\theta = \arg\max_{r} \sum_{(d,e,al)\in train} 1 \Leftrightarrow ((al = True) \ \wedge \ (conf(d,e) \geq r) \ \ \vee \ \ (al = False) \ \wedge \ (conf(d,e) < r)) \tag{4.7}$$

Theta ist derjenige optimierte Wert für alle möglichen Konfidenzwerte r der Kandidatenpaare (d, e) , bei dem der Wert für r die größte Summe ergibt. Es wird nur dann 1 zu der Summe addiert, wenn folgende Bedingung gilt: Der Absatz ist aliniert und der Konfidenzwert von (d, e) ist größer/gleich r oder der Absatz ist nicht aliniert und der Konfidenzwert von (d, e) ist kleiner als r.

4.6.2 Resultat

Der tatsächliche Wert von θ, der aus der Menge aller Tupel der Trainingsmenge bestimmt werden konnte, lautet: 57.9083928571.[8]

Somit ist der optimale Schwellenwert θ für jedes Kandidatenpaar bestimmt und eine binäre Klassifikation kann erfolgen. Liegt ein Kandidatenpaar mit seinem aus dem *ranking* errechneten Konfidenzwert über θ, nur dann gelten der deutsche und der englische Absatz des Kandidatenpaares als miteinander vergleichbar.

Die Menge aller miteinander als vergleichbar geltenden deutsch-englischen Absatzpaare bilden dann das vergleichbare Korpus. Die Akquisition vergleichbarer Absätze aus Wikipedia kann vorgenommen werden, indem diese als vergleichbar geltenden, bilingualen Absatzpaare aus der artikelalinierten Wikipedia-XML-Datei (siehe Kap. 3.4.3) extrahiert werden.[9]

[8] Durch den Mittelwert aller möglichen Ergebnisse bilden sich mehr Nachkommastellen.
[9] Zum Zeitpunkt der Abgabe konnte dies nicht mehr bewerkstelligt und wird post Abgabe noch erledigt werden.

5 Evaluation

Dieser Ansatz wird gegen einen selbst erstellten Goldstandard, der mit Hilfe einiger Kommilitonen angefertigt wurde, evaluiert.
Es wird zuerst beschrieben wie der Goldstandard zustande kam, unter welchen Gesichtspunkten ein Absatz von den Annotatoren als aliniert bewertet werden sollte und welche Kriterien dem zu Grunde liegen.

Der fertige Goldstandard wurde weiterhin aufgeteilt in ein Trainingsset und in ein Testset. Das Trainingsset diente zur heuristischen Bestimmung des Absatzlängenfilters (vgl. Kap. 4.2.2) und vor allem zur Optimierung des Konfidenzwertes (vgl. Kap. 4.6.1), um den Schwellenwert zu erhalten, der für den binären Klassifikator (ein Absatz ist vergleichbar/-nicht vergleichbar) maßgeblich ist. Das Testset darf nur für die Evaluation benutzt werden.

Die Auswertung gegen den anfangs erwähnten Goldstandard wird in zwei Teile aufgeteilt: Um eine Aussage darüber treffen zu können, wie gut der vorliegende Ansatz überhaupt funktionieren kann, wird eine Bewertung des Ansatzes auf den Trainingsdaten vorgenommen.
Eine weitere Evaluation auf dem Testset gibt dann Auskunft darüber wie gut der Ansatz bei ungesehenen Daten funktioniert.

5.1 Erstellung des Goldstandard

Recherchen über die Existenz von Korpora, welche Absätze annotiert haben, die vergleichbar sind, blieben erfolglos. Es musste also selbst ein solches Korpus geschaffen werden, damit der Ansatz in geeigneter Weise bewertet werden kann.

Die Voraussetzungen, ein solch annotiertes Korpus erschaffen zu können, sind gegeben: Es liegt das selbst erstellte artikelalinierte und in Absätzen unterteilte Wikipedia in einer XML-Datei vor (siehe Abb. 3.3). Kandidatenpaare können also den Annotatoren genauso

angeboten werden wie dem Programm, das in Kap. 4 komplett beschrieben wird. Potentielle Annotatoren stellen die Studenten/Kommilitonen der Computerlinguistik dar. Diese eignen sich sehr gut für die Aufgabe, binäre Klassifikationsentscheidungen über die Vergleichbarkeit von zwei Absätzen zu treffen, da sie sowohl sprachwissenschaftlich geschult sind, als auch am Computer die Annotation bewerkstelligen können.

Nachfolgend wird auf die Annotatoren eingegangen, wie diese ihre vergleichbaren Absätze erhalten haben und was die Kriterien für ihre binäre Entscheidung sein sollen.

5.1.1 Annotatoren

Diejenigen Annotatoren, welche für die Annotation des Goldstandards zur Evaluierung des Systems herangezogen wurden, sind allesamt im Studiengang Computerlinguistik eingeschrieben und hatten aufgrund der linguistischen Ausbildung die Voraussetzung, um den Goldstandard nach vorgegebenen Kriterien (diese werden später in Kap. 5.1.4 aufgeführt) erstellen zu können.[1]

Bis auf eine Ausnahme sind alle Annotatoren deutsche Muttersprachler. Alle beherrschen die deutsche Sprache in Wort und Schrift fließend. Für den Studiengang ist eine fundierte Kenntnis der englischen Sprache notwendig. Somit stehen auch keine Sprachschwierigkeiten im Wege, eine Entscheidung bezüglich einer möglichen Vergleichbarkeit von englischen und deutschen Absätzen zu treffen. Zumal liegt es auch in der Natur von Wikipedia als Enzyklopädie, Sachverhalte nicht allzu kompliziert auszudrücken.

5.1.2 Artikelauswahl

Um eine mögliche Beeinflussung zu vermeiden, wurden den Annotatoren zufällig ausgewählte Artikel zugewiesen. Von jedem Artikel bekamen sie jeden deutschen Absatz mit den möglichen zugehörigen englischsprachigen Kandidatenpaaren nach der Formel (4.1)

[1] Besonderen Dank möchte ich hiermit folgenden Personen zuteil werden lassen, die maßgeblich an der Erstellung des Goldstandards beteiligt waren: Mareike Hartmann, Annika Berger, Dominic Jehle, Marc Dohm, Lyubov Nakryyko, Jan Pawellek, Damian Gorski, Mirco Hering, Jonathan Geiger, Dustin Heckmann, Jutta Pieper, Mirjam Eppinger, Benjamin Körner, Anja Summa, Christoph Mayer, Isabell Augenstein, Xenia Kühling, Eric Hildebrand, Britta Zeller, Anna-Katharina Feldmann, Chen Li, Anna Meisinger und Saskia Vola.

zu Gesicht.

Außerdem gab es mehrere Überschneidungen[2], so dass Teile gegenübergestellter Absätze teilweise schon bekannt sein mussten. Dies verkleinerte den zu leistenden Leseaufwand der Annotatoren.

5.1.3 Artikelpräsentation

Um den Aufwand für die Annotatoren so gering wie möglich zu halten und somit auch wenig Ergebnis-Beeinflussung vorliegen zu haben (z.B. durch Demotivation bei der Erledigung ihrer Aufgabe, weil diese zu umständlich zu bearbeiten ist), wurde ein Skript geschrieben, welches die ein bis fünf Kandidaten für einen deutschen Absatz klar erkennbar gegenüberstellt.

Die Annotatoren müssen dann lediglich eine von mehreren zuvor festgelegten Tasten drücken und mit der *return*-Taste bestätigen, damit der Absatz bezüglich ihrer Wahl annotiert wird.[3] Der Rest erledigt das Skript vollautomatisch. Das Skript und somit die Aufgabe konnte jederzeit unterbrochen werden und zu einem beliebigen Zeitpunkt wieder fortgesetzt werden, sodass auch erholsame Pausen möglich waren.

Nachfolgend sei die Absatzpräsentation für die Annotatoren dargestellt, um sich ein Bild davon machen zu können, wie diese ihre Entscheidungen treffen sollten:

[2] nach der Formel in (4.1) bekommt jeder Absatz in deutscher Sprache ein bis fünf auf englisch zugeteilt. Der nächste deutsche Absatz, der ebenfalls den Annotatoren präsentiert wurde, enthält meistens eine Schnittmenge der vorigen englischen Absätze.

[3] die Bestätigung mit der *return*-Taste macht eine Verbesserung der Eingabe möglich, falls aus Versehen eine falsche Zifferntaste betätigt wurde.

Abbildung 5.1: Präsentation der Absätze

Ganz oben in der Abbildung ist der deutschsprachige Absatz zu sehen, welchem hier nach der Formel aus (4.1) vier englische Absätze gegenübergestellt werden.[4] Damit auch hier eine tatsächliche Positionsangabe nicht ersichtlich ist (das könnte auch eine Beeinflussung darstellen), wurde die deutsche Absatzposition entfernt und die möglichen vergleichbaren ein bis fünf englischen Absätze von 1 bis 5 durchnummeriert (tatsächlich weiß das System noch, welche Absatzpositionen genau vorliegen).
Mit den angebotenen Zahlen kann eine Alinierung direkt bestimmt werden, n bedeutet "keine Alinierung vorhanden" (das sollte hier die Wahl gewesen sein) und f bedeutet, dass der deutsche Absatz fehlerhaft ist (z.B. aus einem Formelrest besteht, der offensichtlich

[4] Wenn weniger als fünf gegenübergestellte Absätze vorliegen, dann gibt es für die normalisierte Absatzposition im englischsprachigen Artikel im Vergleich zu der Absatzposition des deutschen Absatzes weniger als zwei Absätze davor oder danach.

nicht richtig bereinigt wurde).

Wurde eine Alinierung ausgewählt, so wurden die ausgewählten Absätze als vergleichbar eingestuft, die anderen präsentierten Kandidaten erhielten das Prädikat “nicht vergleichbar”. So erhält der Goldstandard bei einer erfolgreichen Alinierung bei n gegenübergestellten Absätzen auch n Alinierungen, wobei eine davon als vergleichbar gewertet wird und $n - 1$ als nicht vergleichbar.

5.1.4 Kriterium & Fragestellungen

Eine gute Definition für Vergleichbarkeit ist schwer zu leisten und auch schwer zu finden. Vergleichbarkeit wird immer unter gewissen Gesichtspunkten gesehen.

Für vorliegenden Ansatz wurde festgelegt, dass ein Absatz in deutscher Sprache mit einem Absatz in der englischen Sprache vergleichbar ist, wenn der gleiche Aspekt beleuchtet wird. Hilfreiche Fragestellungen wurden überlegt, um das Aspekt-Kriterium erfassen zu können. Dabei sollen diese den Anspruch haben, größtmöglichste Akzeptanz bei den Annotatoren zu erlangen, als auch das Aspekt-Kriterium näher zu spezifizieren.

Folgende Fragestellungen stellten sich dabei als hilfreich heraus:

- Wird in etwa das Gleiche ausgesagt?
- Kann eine gleiche Überschrift gefunden werden?
- Weicht die Spezifität allzu sehr voneinander ab?

Somit war es den Annotatoren möglich, den Goldstandard unter bestimmten Gesichtspunkten zu erstellen.

5.1.5 Goldstandard-Statistiken

Die Entscheidung, ob ein Absatzpaar in den Goldstandard aufgenommen werden darf, wurde nicht einem Annotator alleine überlassen. Es galt das Vier-Augen-Prinzip. Jede von einem Annotator erstellte Alinierung musste erst durch einen zweiten Annotator bestätigt werden. Ansonsten wurde die Alinierung verworfen.

So wurde ein artikelaliniertes, aber absatzunaliniertes Korpus, welches 3.156 Absätze mit ein bis fünf Gegenüberstellungen enthielt (das entspricht 867.063 Wörtern), auf insgesamt 23 Annotatoren nach dem Vier-Augen-Prinzip aufgeteilt.[5] Durchschnittlich bearbeitete also jedes Annotatoren-Paar 263 deutschsprachige Absätze mit englischsprachigen Gegenüberstellungen (das waren durchschnittlich 72.255 Wörter).

Bei diesen bearbeiteten 3.156 deutschsprachigen Absätzen mit ihren möglichen englischen Entsprechungen hatten 2.288 ein Inter-Annotator-Agreement von 100% (also beide Annotatoren waren sich einig). Aus dieser Menge konnten insgesamt 509 Absatzpaare als vergleichbar eingestuft werden und 1779 deutschsprachige Absätze fanden laut Annotatoren-Paar keine Entsprechung in der Menge der angebotenen englischsprachigen Absätzen.

Diese 509 erfolgreich als vergleichbar eingestuften Absatzpaare wurden extrahiert und in den Goldstandard aufgenommen. Dabei sind implizit die ein bis vier anderen Absatzpaare der untersuchten deutschen Absätze dieser 509 Absatzpaare nicht miteinander vergleichbar.
Diese werden bei der Optimierung eines Parameters im Trainingsset und bei der Evaluation berücksichtigt. Das ist möglich, da für die Annotatoren die gleiche Formel (4.1) benutzt wurde, die auch das Programm zum Gegenüberstellen von deutschen und englischen Absätzen benutzt.

5.1.6 Trainings- und Testset

Um Parameter optimieren oder Heuristiken festlegen zu können, wird ein bereits absatzaliniertes Korpus benötigt. Man hätte sonst bei verschiedenen Einstellungen von Systemwerten keine Überprüfbarkeit, ob diese zu einer Verbesserung führen.

Aus diesem Grund wird der Goldstand in zwei Sub-Korpora aufgeteilt. Ein kleiner Teil wird das Trainingskorpus, welches für obige Belange genutzt werden soll (z.B. die Optimierung eines Parameters, vgl. Kap. 4.6.1).
Der andere Teil bildet den Testkorpus. Dieser repräsentiert ungesehene Daten und ent-

[5] Wobei drei Annotatoren freundlicherweise für den 24. eingesprungen sind, der aufgrund technischer Probleme die Annotation nicht bewerkstelligen konnte.

hält absatzalinierte Daten, weshalb er sich für eine Bewertung des Ansatzes, bzw. der Modellierung des Ansatzes, eignet.

Das Trainings- und Testkorpus wurde etwa im Verhältnis 1:5 aufgeteilt. Das Trainingsset besteht nun aus 102 deutschsprachigen Absätzen, denen ein englischer Absatz als vergleichbar gegenübergestellt ist. Das Testset besteht aus 407 solcher Alinierungen.
Dabei sei hier erwähnt, dass diesen nicht overt, aber später bei der Evaluation und auch schon bei der Bestimmung des optimalen Parameters ein bis vier weitere Absätze zugeteilt sind, welche per definitionem nicht aliniert sind.

5.2 Evaluation gegen den Goldstandard

Der mühevoll und sorgfältig erstellte Goldstandard ist ein sehr gutes Vergleichskriterium für vorliegenden Ansatz. Dabei können beide Teile des Goldstandards, also das Trainings- und das Testset, für diese Evaluation genutzt werden.

Die Evaluation gegen das Trainingsset gibt Auskunft darüber, wie gut der Ansatz überhaupt funktionieren kann, die Evaluation gegen das Testset, wie gut der Ansatz auf bislang ungesehenen Daten ist.

Für die Evaluation wurden die *true positives*, die *true negatives*, die *false positives* und die *false negatives* gezählt und ausgewertet.

True positives sind die Absätze, welche sowohl der vorliegende Ansatz als auch die Annotatoren als vergleichbar eingestuft haben.
Die Menge der *true negatives* bilden diejenigen Absätze, welche sowohl vom Ansatz als auch von den menschlichen Annotatoren als nicht alinierbar eingeordnet wurden.
Die für diesen Ansatz schlechten Werte bestehen aus *false positives* und *false negatives*. Ersteres der beiden bezeichnet diejenigen Absätze, welche der Ansatz gerne akquirieren würde, die Annotatoren aber damit nicht einverstanden sind.
Letzteres bedeuted, dass vorliegender Ansatz diejenigen Absätze in der Menge der *false negatives* nicht in das neue Korpus mitaufnehmen will, die Annotatoren diese aber doch gerne akquiriert haben möchten.

Aus diesen vier Klassifikationen ergeben sich verschiedene Gütemaße, die im Folgenden vorgestellt werden:

$$\texttt{precision} = \frac{\texttt{number of true positives}}{\texttt{number of true positives} + \texttt{number of false positives}} \tag{5.1}$$

Die `precision` ist ein Proportionsmaß für diejenigen Absätze, die das System richtig erkannt hat (vgl. (Manning and Schütze, 1999, 268)). Dieser ist für die Akquisition ein wichtiger Wert, da ein Korpus möglichst fehlerfrei sein sollte.

$$\texttt{recall} = \frac{\texttt{number of true positives}}{\texttt{number of true positives} + \texttt{number of false negatives}} \tag{5.2}$$

`Recall` ist definiert als die Proportion zu den Absätzen, welche das System hätte erkennen können (vgl. (Manning and Schütze, 1999, 268f)). Dieser Wert ist dann wichtig, wenn die Größe des entstehenden Korpus eine relativ große Rolle spielen soll, da mit niedrigem `recall` ein kleineres Korpus zu erwarten ist, bzw. langsamer anwächst - je nachdem, ob eine quasi unendliche Menge an Daten zur Verfügung steht (z.B. das Internet) oder eine endliche Menge (z.B. Ein Wikipedia-*dump*).

Das sind im Prinzip sich ausschliessende Werte. Um diese doch vereinigen zu können, können diese in ein einzelnes kombiniertes Maß münden:

$$\texttt{F-measure} = \frac{1}{\alpha\frac{1}{\texttt{precision}} + (1-\alpha)\frac{1}{\texttt{recall}}} \tag{5.3}$$

Das kombinierte Maß aus Formel (5.1) und (5.2) namens `F-measure` ist noch konfigurierbar. Bei einem Wert von 0.5 für α gewichtet man `precision` und `recall` gleichermaßen. Legt man nun mehr Wert auf `precision`, so kann α einfach erhöht werden (vgl. (Manning and Schütze, 1999, 269)).

All diese Gütemaße werden im Folgenden für die Bewertung herangezogen.

5.2.1 Evaluation gegen das Trainingsset

	positives	negatives
true	42	230
false	10	60

Abbildung 5.2: Konfusionsmatrix aus dem Trainingsset

Die guten Werte stehen in der ersten Zeile: Das System hat 42 Absatzpaare erkannt, die auch die Annotatoren aufnehmen würden, und 230 verworfen, welche auch im Goldstandard verworfen wurden. Dafür wären aber zehn Absatzalinierungen aufgenommen worden, welche nichts im Korpus zu suchen hätten, 60 alinierte Absatzpaare blieben unerkannt.

Gütemaß	Wert
precision	**80.77** %
recall	41.18 %
$F_{0.5}$	54.55 %
$F_{0.75}$	65.12 %

Abbildung 5.3: Ergebnisse der Trainingsset-Evaluation

Der Ansatz zeigt eine hohe Diskrepanz bezüglich `precision` und `recall`. Das heißt, daß der Ansatz lieber genauer vorgeht, anstatt möglichst viele Absatzpaare aufnehmen zu können, auch auf die Gefahr hin, dass auch ein paar falsche Alinierungen dabei sind. Folgerichtig steigt das F-Maß an, wenn man die `precision` mehr gewichtet:

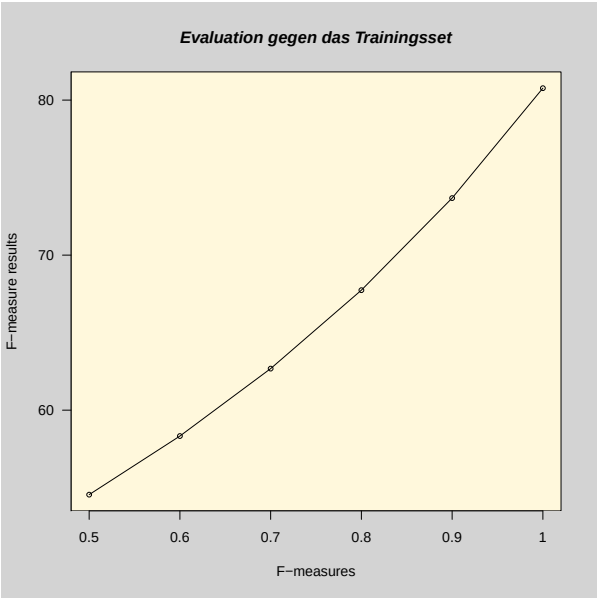

Abbildung 5.4: verschiedene F-Gewichtungen

Je mehr Gewicht auf `precision` gelegt wird, desto besser sind die Ergebnisse.

5.2.2 Evaluation gegen das Testset

	positives	negatives
true	113	941
false	15	294

Abbildung 5.5: Konfusionsmatrix aus dem Testset

Die guten Werte stehen abermals in der ersten Zeile: Es wurden 113 Absatzpaare erkannt, die auch die Annotatoren aufnehmen würden, und 941 fänden richtigerweise keine Aufnahme in das neue Korpus. 15 Absatzalinierungen würden aus diesem Testset fälschlicherweise in das neue vergleichbare Korpus fliessen, und 294 würden vom Ansatz verworfen werden, obwohl sie eigentlich vergleichbar sind.

Gütemaß	Wert
precision	**88.28** %
recall	27.76 %
$F_{0.5}$	42.24 %
$F_{0.75}$	57.14 %

Abbildung 5.6: Ergebnisse der Testset-Evaluation

Auf ungesehenen Daten wirkt sich der `recall` katastrophaler aus. Es gibt hier schlichtweg viele Kandidaten, die für die Erkennung als alinierte Absätze in Frage gekommen wären. Diese wurden vom System aber nicht erkannt. Der Wert der `precision` hingegen hat sich sogar verbessert, dieser erreicht fast die 90%-Marke.

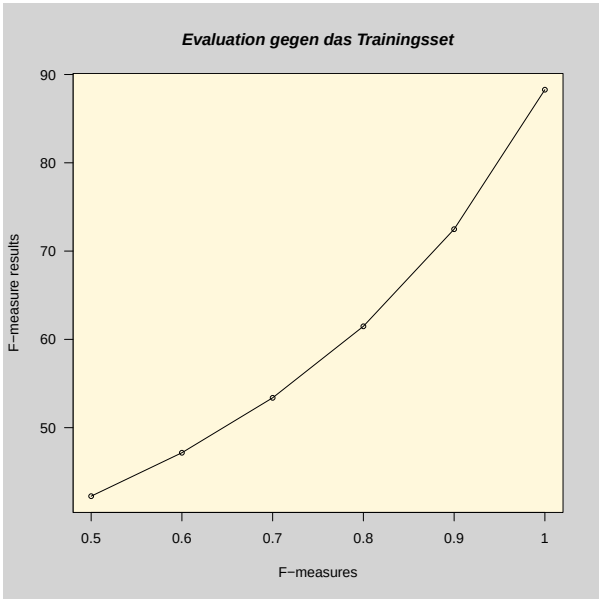

Abbildung 5.7: verschiedene F-Gewichtungen

Hier ist ein wenig mehr noch als im Trainingsset ersichtlich, dass die `precision` ein entscheidendes Kriterium darstellt.

5.3 Resultate und Interpretation

Zur Veranschaulichung seien von $F_{0.5}$ bis $F_{1.0}$ die Ergebnisse des F-Maßes für das Trainingsset und das Testset erneut dargelegt:

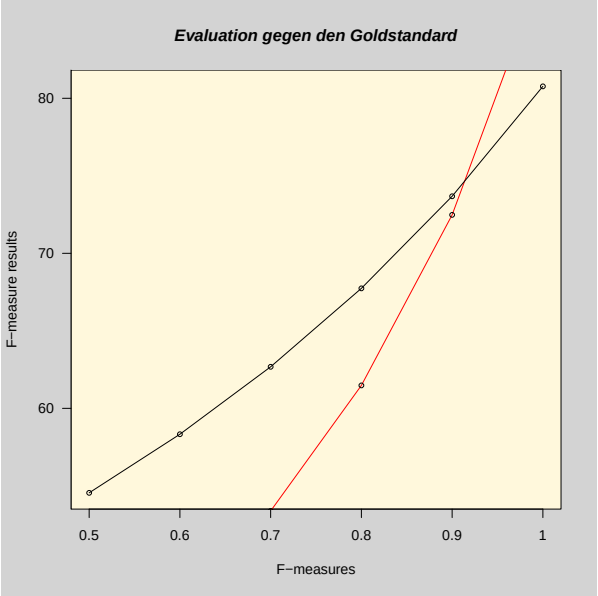

Abbildung 5.8: F-Maß-Korrelation zwischen Trainings- und Testset

Die steilere Kurve ist das Ergebnis aus dem Testset, die nahezu stetig ansteigende Kurve repräsentiert die Werte aus dem Trainingsset.

Durch obig dargestellten Sachverhalt kann man interpretieren, dass das System so modelliert wurde, möglichst fehlerfrei die Absätze zu erkennen. Das resultierte allerdings in schlechten Werten bezüglich dem, was es zu erkennen gab. Erst wenn man die `precision` sehr hoch gewichtet, bei einem α-Wert von ca. 0.9 des F_α-Maßes (siehe Formel (5.3)), bekommt man sowohl für die trainierten Daten als auch für die ungesehenen Daten relativ zufriedenstellende Werte.

Relativ zufriedenstellend sind diese Werte trotz der Tatsache, dass bei einer `precision` von ca. 90% dann ca. 13%[6] irrtümlicherweise in ein neues Korpus überführt würden, da

[6] Die 13% setzen sich dabei aus den 113 *true positives* und den 15 *false positives* aus dem set der ungesehenen Daten (siehe Abb. 5.5) zusammen $\rightarrow \frac{15}{113} \cdot 100$.

auch dieser Ansatz Ressourcen-abhängig ist.

So ist vorliegender Ansatz ein lexikonbasierter Ansatz. Um so dramatischer erscheint es, wenn 34.607 von insgesamt 39.121 vom System untersuchten Wörter aus dem Trainingset und 127.645 von 144.410 aus dem Testset keinem Feature zugeordnet werden konnten. Dieser *bottle-neck* erklärt die hohe Zahl an Absätzen, welche vom Menschen als vergleichbar eingestuft wurden, vom System aber nicht identifiziert werden konnten. Desto erfreulicher ist aber der `precision`-Wert. Das heißt, es braucht keine allzugroße semantische Schnittmenge, um zwei vergleichbare Absätze miteinander alinieren zu können.

6 Schluss

Es besteht Hoffnung für das vergleichbare absatzalinierte, bilinguale Korpus.

Trotz eines großen *bottle-necks* des Wörterbuchs kann semantische Ähnlichkeit mit linguistischen Features durchaus festgestellt werden.

Die Methoden dieses Ansatzes gingen dabei aber auch statistisch vor. Es wurde mit Maßen wie tf·idf und anderen gängigen Formeln gearbeitet, um die einzelnen Wörter zu gewichten. Außerdem wurde ein struktureller Filter integriert, um Absätze frühzeitig aussortieren zu können. Dort kam ein großzügig ermessener heuristischer Wert zum Einsatz.x

Mittels des entwickelten Punktesystems, welches Mehrwortausdrücke hoch gewichtet und einen Abgleich der Wörter mit den Methoden *string-matching*, simples Nachschlagen im Wörterbuch und Lemmatisierung vornimmt und diese mittels einer tf·idf-Gewichtung zu der Absatzpaar-Gesamtsumme addiert, lässt sich also durchaus die Frage positiv beantworten, ob vergleichbare Textstellen auf der Absatzebene überhaupt identifiziert werden können.

Die `precision` dieses Ansatzes für ungesehene Daten liegt bei ca. 90%, allerdings mit einem niedrigen `recall`. Hier besteht noch Ausbaubedarf.

Eine `recall`-Verbesserung könnte man vor allem mit dem Hinzufügen besserer Wörterbücher erreichen, die `precision` kann verbessert werden, indem ein biderektionaler Vergleich angestrebt wird. Also nicht nur die deutschen Absätze mit den englischen verglichen werden, sondern auch umgekehrt.

Das durch diesen Ansatz akquirierte Korpus kann dazu verwendet werden, um Modelle auf semantische Ähnlichkeit zu trainieren. Diese haben durch die Texteinheit `Absatz`

mehr Informationen als bei Sätzen oder gar Wörtern.

Hiermit soll allgemein ein Beitrag zur Bekämpfung von Ressourcen-Knappheit geleistet sein.[1]

[1] Das dürfte ganz im Sinne von (Heid, 2002) sein.

Literaturverzeichnis

Apache Lucene. http://lucene.apache.org/java/docs/index.html.

Natural Language Toolkit. http://www.nltk.org/.

TreeTagger. http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/.

Wortschatz Universität Leipzig. http://wortschatz.uni-leipzig.de/.

Carstensen, K.-U., C. Eber, C. Endriss, S. Jekat, R. Klabunde, and H. Langer (Eds.) (2004). *Computerlinguistik und Sprachtechnologie. Eine Einführung*. Spektrum Akademischer Verlag.

Falk, Y. N. (2001). *Lexical-Functional Syntax: An Introduction to Parallel Constraint-Based Syntax*. Stanford, California: CSLI Publications.

Gale, W. A. and K. W. Church (1993). A program for aligning sentences in bilingual corpora. *Comput. Linguist. 19*, 75–102.

Heid, U. (2002). Computerlinguistische hilfsmittel für die wörterbucherstellung. In G. Willée, B. Schröder, and H.-C. Schmitz (Eds.), *Computerlinguistik. Was geht, was kommt?*, Sprachwissenschaft Computerlinguistik. Neue Medien. Band 4, pp. 128–132. Gardez! Verlag.

Jurafsky, D. and J. H. Martin (2000). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Englewood Cliffs, New Jersey: Prentice Hall.

Manning, C. D. and H. Schütze (1999). *Foundations of Statistical Natural Language Processing*. Cambridge, MA: MIT Press.

Resnik, P. and N. A. Smith (2003). The web as a parallel corpus. *Comput. Linguist. 29*(3), 349–380.

Schmid, H. (1994). Probabilistic part-of-speech tagging using decision trees. In *Proceedings of International Conference on New Methods in Language Processing*.

Tsvetkov, Y. and S. Wintner (2010). Automatic acquisition of parallel corpora from websites with dynamic content. In *proceedings of the seventh international conference on Language Resources and Evaluation*, Valletta, Malta, pp. 3389–3392. LREC 2010.