

Bauhaus-Universität Weimar  
Fakultät Medien  
Studiengang Medieninformatik

# Query Spelling Correction

## Bachelorarbeit

Anja Rathgeber  
geb. am: 27.07.1991 in Erfurt

Matrikelnummer 100621

1. Gutachter: Junior-Prof. Dr. Matthias Hagen
2. Gutachter: Dr. rer. nat. Martin Potthast

Datum der Abgabe: 09. Mai 2016

# Selbstständigkeitserklärung

Hiermit versichere ich, dass ich diese Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

Weimar, 09. Mai 2016

.....

Anja Rathgeber

# Abstract

This work concerned with the review, analysis and evaluation of an existing corpus with almost 55,000 requests for the use in context of *query spelling correction*. The corpus was controlled with automated and manual procedures in terms of spelling. Using a Python script an english dictionary based spell checker was accomplished. In the manual method, the errors were found then reviewed and corrected accordingly. As part of the following error analysis an automatic check for six different types of errors (insertion, deletion, transposition, substitution, and inserting spaces or special characters) was performed. Finally, the finished corpus was evaluated with an existing system.

# Zusammenfassung

Diese Arbeit befasst sich mit der Überarbeitung, Analyse und Evaluierung eines vorhandenen Korpus mit beinahe 55.000 Anfragen für eine Nutzung im Rahmen von *query spelling correction* (zu Deutsch: Korrektur von Suchanfragen). Dabei wurde das Korpus mit automatisierten und manuellen Verfahren hinsichtlich der Rechtschreibung kontrolliert. Mittels eines Python-Skripts wurde eine Rechtschreibprüfung, basierend auf einem englischen Wörterbuch, durchgeführt. Im manuellen Verfahren wurden dann die gefundenen Fehler nachgeprüft und entsprechend korrigiert. Im Rahmen der anschließenden Fehleranalyse erfolgte eine automatische Prüfung auf sechs verschiedene Fehlerarten (Insertion, Deletion, Transposition, Substitution, sowie das Einfügen von Leer- oder Sonderzeichen). Abschließend wurde das fertige Korpus mit einem vorhandenen System evaluiert.

# Inhaltsverzeichnis

<b>1. Einleitung</b>	5
<b>2. Grundlagen</b>	8
2.1 Schreib- und Tippfehler	8
2.2 Suchmaschinen	13
<b>3. Verwandte Arbeiten</b>	20
3.1 Vergleichbares Korpus	20
3.2 Existierende Verfahren	21
<b>4. Webis Query Spelling Correction 2016 Korpus</b>	24
4.1 Korpusannotation	24
4.1.1 Entfernung von Duplikaten	24
4.1.2 Semiautomatische Rechtschreibkontrolle	25
4.1.3 Review der Rechtschreibkontrolle von 2010	28
4.2 Fertigstellung des Korpus	29
<b>5. Korpusanalyse</b>	32
5.1 Korpusvergleich	32
5.2 Anfragenlängenverteilung	33
5.3 Fehleranalyse	34
<b>6. Evaluierung</b>	40
<b>7. Fazit und Ausblick</b>	42
<b>Literaturverzeichnis</b>	43
<b>Anlage 1: Tabelle für Buchstabenauslassung</b>	48

# Kapitel 1

## Einleitung

Diese Arbeit beschäftigt sich mit der Korrektur von Tipp- und Rechtschreibfehlern von Suchanfragen. Ziel ist es, vorhandene Verfahren zur *query spelling correction* (zu Deutsch: Korrektur von Suchanfragen) zu untersuchen und einen zur Verfügung stehenden umfangreichen Korpus für diese Verfahren aufzubereiten und diesen anschließend zu analysieren.

In der gegenwärtigen Zeit ist es einfacher denn je, zu jeder Zeit an jedem Ort, an Informationen zu gelangen. Wer eine Wissenslücke schließen will oder vertiefende Informationen zu einem bestimmten Thema sucht, wird nicht, wie es im letzten Jahrhundert noch gängig war, eine Bibliothek aufsuchen um dort die Antwort in Millionen von Büchern nachzuschlagen. Selbst der Gang zum eigenen oder öffentlichen Computer ist mittlerweile nicht mehr der schnellste Weg. Denn jeder Zweite besitzt heutzutage ein leistungsstarkes Smartphone mit Internetanbindung [StatD2016], welches die benötigten Informationen binnen kürzester Zeit liefern kann. Letztendlich verfolgt selbst Google, der Marktführer unter den Internet-Suchmaschinen [StatMS2016], den Ansatz „Man sitzt nicht immer am Schreibtisch, wenn man eine Antwort benötigt.“ [GoGS] und arbeitet seither an einer stetigen Verbesserung seiner mobilen Anwendungen.

Mit Hilfe von Suchmaschinen ist es möglich, mit nur wenigen Suchbegriffen/Schlagwörtern, die gewünschten Informationen auf mehr als 136 Milliarden digitalen DIN A4-Seiten zu suchen [HK2015]. Ob über einen bekannten ehemaligen Basketballspieler wie Lamar Odom, die französische Satirezeitschrift Charlie Hebdo oder das Webbrowser-Multiplayerspiel Agar.io (die Top 3 der Suchanfragen bei Google weltweit im Jahr 2015) [GoT2015], für alles liefert das World Wide Web innerhalb von Sekunden die passende Antwort.

Doch gerade durch das schnelle Suchen mit dem Smartphone, kann es beim Eingeben der Anfrage zu vielen Fehlern kommen. Denn oft wird bei der Bedienung nicht die volle Konzentration aufgebracht, da das Smartphone unser Leben in ein permanentes Multitasking verwandelt. Hinzu kommt dann unter Umständen auch noch die Unwissenheit, über die richtige Schreibweise einzelner Wörter und Namen. Diese sind jedoch kein Problem für die heutigen Suchmaschinen. Neben der automatischen Vervollständigung der Suchanfragen, welche so Fehler direkt vermeidet, bieten die Suchmaschinen oft auch direkt Alternativsuchen an á la „Meinten Sie:“.

Um diese Korrekturvorschläge und Rechtschreibverbesserungen zu ermöglichen, wird in dieser Arbeit ein zur Verfügung stehendes umfangreiches Korpus mit mehr als 50.000 Suchanfragen aufbereitet und für die entsprechenden Verfahren optimiert. Das gegebene Korpus entstand 2010 in drei Schritten aus den Logdateien der Suchmaschine AOL, welche rund 36 Millionen Anfragen enthielten. Im ersten Schritt wurden zunächst ungewollte Anfragen nach festgelegten Kriterien entfernt. Aus den verbleibenden etwa 6 Millionen Anfragen wurden dann zufällig 55.555 ausgewählt und zum *Webis Query Segmentation Corpus* zusammengefasst. Im dritten Schritt wurden fehlerhafte Anfragen korrigiert. [HPSB2011] Dabei wurden für fast 15% der Anfragen Tipp- und Rechtschreibfehler gefunden und berichtigt. Weitere 611 Anfragen wurden aus dem Korpus entfernt, da diese nicht der englischen Sprache angehörten. Es folgt eine eingehende Untersuchung, welche Fehler bei der Eingabe von Suchanfragen entstehen können, sowie eine Betrachtung der Geschichte und Funktionsweise der gängigsten Suchmaschinen. Nach einem Überblick, über einen vergleichbaren Korpus am Markt und existierende Verfahren zur *query spelling correction*, erfolgt die Bearbeitung des als Grundlage dienenden Korpus. Eine erneute Korrektur und Annotation von weiteren Varianten für einen Teil der Anfragen sollen diesen jetzt weiter verbessern. Dazu wird in dieser Arbeit das aktuell bestehende Korpus mit einer Kombination aus automatisierten und manuellen Verfahren

aufbereitet und für die Weiterverarbeitung optimiert. Mit Hilfe von Python-Skripten und der aktuell am Markt stärksten Suchmaschine erfolgt eine halbautomatische Fehlerkorrektur. Bei Unklarheiten über die korrekte Schreibweise wird sich außerdem an einem englischen Wörterbuch, sowie an den Einträgen von Wikipedia orientiert. Des Weiteren wird ein besonderer Wert auf die Annotation von Sonderzeichen gelegt, um eine möglichst korrekte Schreibweise zu bieten. Im Anschluss werden die Fehler, welche durch die Nutzer der Suchmaschine entstanden sind, analysiert und ausgewertet. Eine Evaluierung des Korpus erfolgte abschließend mit einem vorhandenen Verfahren. Das aus dieser Arbeit resultierende Korpus soll als Grundlage für die Nutzung von Verfahren der *query spelling correction* dienen.



## Kapitel 2

# Grundlagen

Die heutigen Suchmaschinen liefern nicht nur für korrekte Anfragen die passenden Antworten. Nahezu mühelos gehen sie auch mit beliebigen Schreib- und Tippfehlern um und wissen, was wir eigentlich meinten. In diesem Kapitel werden die Arten von Schreib- und Tippfehlern, sowie die gängigsten Suchmaschinen behandelt. Da das entstandene Korpus ausschließlich aus englischen Anfragen besteht, wird auch bei der Betrachtung der Fehlerarten der Fokus auf die englische Sprache gelegt.

## 2.1 Schreib- und Tippfehler

Bei der Eingabe von Suchanfragen können durch den Nutzer verschiedene Fehler auftreten. Diese Eingabefehler lassen sich auf zwei Ursprungsarten zurückführen. Eine dieser Arten ist, dass der Nutzer nicht weiß, wie man das Wort richtig schreibt. Die andere ist motorisch basiert, das heißt, die Fehler treten durch fehlerhafte Bedienung der Tastatur auf, zum Beispiel durch die Eingabe auf einer kleineren Tastatur wie bei einem Smartphone.

Ein möglicher Schreibfehler ist die homophone Variante eines Wortes. [BL2011] Dieser kommt zum Beispiel zu Stande, wenn der Nutzer das Wort bisher nur gehört, aber noch nie geschrieben gesehen hat. Die Ursachen hierfür liegen somit im Linguistischen. Dies ist beispielsweise bei den englischen Wörtern „night“ und „knight“ der Fall. Beide werden [nahyt] ausgesprochen und klingen somit identisch. Die Bedeutungen jedoch, reichen mit „Nacht“ und „Ritter“ weit auseinander. Bei diesem Fall ist es allerdings nicht direkt nachzuvollziehen, ob hier wirklich ein phonetischer Fehler vorliegt. Denn die Worte unterscheiden sich nur in einem

Buchstaben, dem Anfangsbuchstaben, und somit ist hier auch ein Tippfehler nicht unwahrscheinlich. Ein eindeutigeres Beispiel für einen phonetischen Fehler bilden die Worte „slay“ (zu Deutsch „töten“) und „sleigh“ (zu Deutsch „Schlitten“), mit der identischen Lautsprache [sley]. Die Schreibweisen unterscheiden sich deutlich und lassen die Schlussfolgerung zu, dass ein Schreibfehler vorliegt. Die Aussprache kann auch dann der Grund des Fehlers sein, wenn sich diese deutlich von der Schreibweise unterscheidet. Bei „lettuce“ (zu Deutsch „Kopfsalat“) lässt die Aussprache [let-is] nicht automatisch „uce“ vermuten. Bei „colleague“ [kol-eeg] (zu Deutsch „Kollege“) sind die letzten Buchstaben stumm und werden nicht gesprochen. So kann es also passieren, dass Buchstaben aufgrund ihrer anderen oder gar Auslassung in der Lautgebung auch in der Schrift verändert oder vergessen werden. Die Seite spellchecker.net stellte im Jahr 2010 die in Tabelle 1 gezeigte Statistik über die häufigsten Schreibfehler in den Vereinigten Staaten von Amerika auf.

	<b>falsche Schreibweise</b>	<b>richtige Schreibweise</b>
1.	thier	their
2.	alot	a lot
3.	recieved	received
4.	seperate	separate
5.	untill	until
6.	becuase	because
7.	begining	beginning
8.	diffrent	different
9.	occured	occurred
10.	beleive	believe
11.	behaviour	behavior
12.	wich	which
13.	truely	truly
14.	realy	really
15.	definatly	definitely

**Tabelle 1: Top 15 der häufigsten Schreibfehler in den Vereinigten Staaten von Amerika [GRMW]**

In der Top 15 ist zu erkennen, dass sich unterscheidende Schriftbilder und Lautbilder eine häufige Ursache für Schreibfehler sind.

Bei der Nutzung von Suchmaschinen spielen neben den Schreibfehlern auch die Tippfehler eine entscheidende Rolle. Diese lassen sich in vier Arten (Insertion, Deletion, Substitution, Transposition) unterteilen.

### **Insertion**

Bei Insertion-Fehlern wurde durch den Nutzer mindestens ein Buchstabe zu viel eingefügt. Dieser Fehler kann zum Beispiel auftreten, wenn unbeabsichtigt mehrere Tasten auf einmal gedrückt werden. Geläufig ist dies auch als „Fat Finger“ (zu Deutsch „Fette Finger“) Fehler, welcher vor allem bei Geschäften im Finanzmarkt dadurch bekannt wurde, dass deutliche andere Geldsummen bewegt wurden als beabsichtigt. [FAZ2014] Eine weitere Möglichkeit ein zusätzliches Zeichen zu verursachen ist das zu lange Verweilen auf einer Taste. Durch diesen „Long Press“-Fehler kann es zu der ungewollten Dopplung eines Buchstabens kommen.

#### *Beispiel für „Fat Finger“-Fehler:*

Statt *search* wurde *seasrch* eingegeben, da die Tasten A und S gleichzeitig gedrückt wurden, da diese auf der Tastatur neben einander liegen.

#### *Beispiel für „Long Press“-Fehler:*

*search* wurde durch das zu lange Verweilen auf der Taste A als *seearch* eingegeben.

### **Deletion**

Wenn der Nutzer bei der Eingabe eines Wortes einen Buchstaben vergisst, wird dies als „Deletion“, also als Löschung, bezeichnet. Besonders durch sehr schnelles Tippen auf der Tastatur oder auf dem Smartphone können einzelne Tasten unbeabsichtigt vergessen werden. Es kann auch

passieren, dass sie zu durch einen zu geringen Druck auf der Taste nicht erkannt werden und aus diesem Grund in der Anfrage fehlen. Ein Unterschied des Druckpunkts macht sich vor allem im direkten Vergleich von halbmechanischen und mechanischen Tastaturen bemerkbar. Bei den halbmechanischen befindet sich unter der Taste ein Rubberdome aus Gummi oder Silikon, welcher die Taste nach Druck automatisch wieder in ihre Ausgangsstelle zurückbringt. Bei mechanischen Tastaturen hingegen wird mit Schaltern und Federn aus Metall gearbeitet, welche durch ihre Härte den benötigten Druck bestimmen. [DS2012]

*Beispiel für Schnelligkeits-Fehler:*

Das Wort *search* wurde durch das schnelle Tippen zu *serch*.

*Beispiel für Druck-Fehler:*

Der Nutzer hat bei der Eingabe von *search* die Taste C mit zu wenig Druck betätigt und somit *searh* eingegeben.

## **Substitution**

Bei der Substitution wird mindestens ein Buchstabe im Wort durch einen anderen ersetzt. Vor allem durch das blinde Schreiben auf der Computertastatur kann es passieren, dass versehentlich eine andere Taste gedrückt wird. Damit dies nicht aufgrund der Nutzung einer anderen Tastatur als üblich passiert, sind die Tastenabstände und Versätze nach der deutschen Norm DIN 2137-2:2012-06 genau festgelegt. Auch bei der Nutzung einer Tastatur in einem anderen Land kann es zu Substitutionsfehlern kommen. In den USA, den meisten englischsprachigen Ländern und Dänemark findet man eine QWERTY-Buchstabenanordnung. Die Lage der Buchstaben unterscheidet sich im Vergleich zur deutschsprachigen QWERTZ-Tastatur nur in der Vertauschung zwischen Y und Z. In Frankreich und Belgien wird die AZERTY-Tastaturbelegung verwendet. Diese unterscheidet sich zur QWERTY-Buchstabenanordnung

durch die Vertauschung von A und Q, Z und W und der Buchstabe M ist eine Reihe nach oben, neben das L, verschoben.

*Beispiel für einen Fehler durch blindes Schreiben:*

Bei der Eingabe von *search* wurde durch blindes Schreiben statt dem A der benachbarte Buchstabe S eingegeben und somit *sesrch*.

*Beispiel für einen Fehler durch Nutzung einer anderssprachigen Tastatur:*

Der Nutzer hat statt einer QWERTY-Tastatur eine AZERTY-Tastatur verwendet und somit durch die verschiedene Buchstabenanordnung *seqrch* statt *search* eingegeben.

## **Transposition**

Wenn eine Vertauschung von zwei Buchstaben in einem Wort vorliegt, spricht man von einer Transposition. Dieser Fehler kann ebenfalls durch das blinde und schnelle Schreiben auf der Tastatur auftreten.

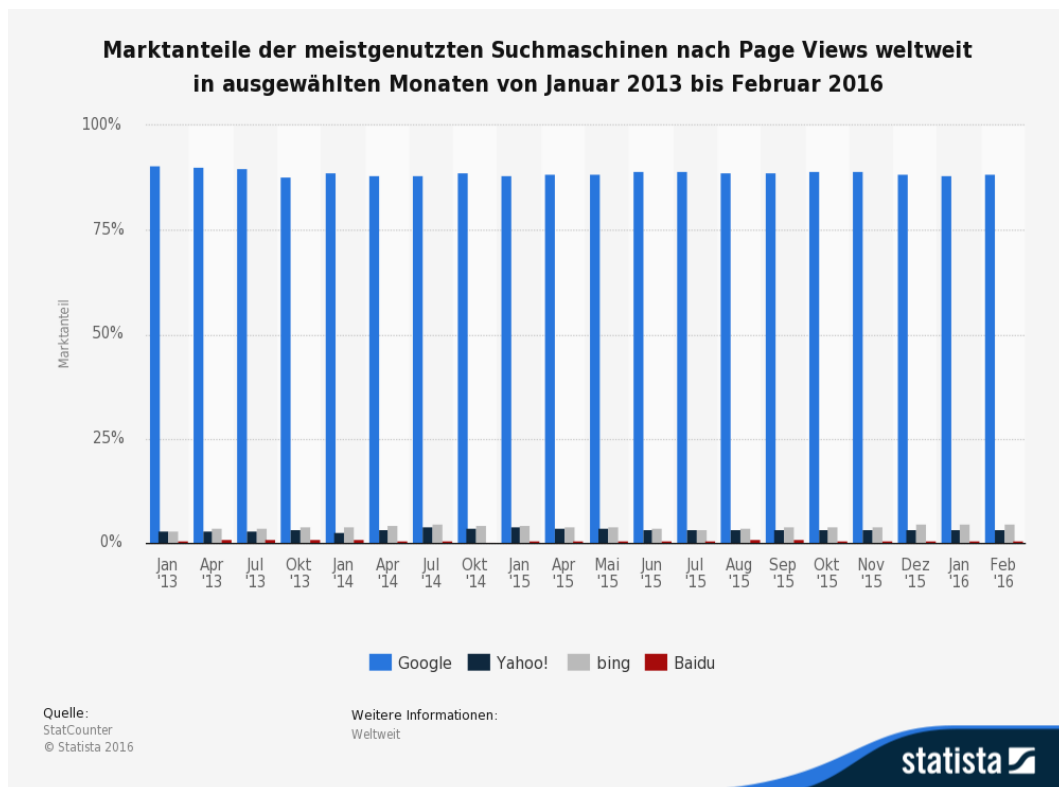
*Beispiel für einen Transposition-Fehler:*

Statt *search* wurde durch das schnelle Tippen *saerch* eingegeben.

Laut einer von Dvorak, Merrick, Dealet und Ford verfassten Statistik in „Typewriting Behaviour“ tritt der Substitutionsfehler bei der QWERTY-Tastatur mit 40% statistisch am häufigsten auf, gefolgt von der Auslassung von Buchstaben mit 20% und der Transposition mit 15%. Das Einfügen von zusätzlichen Buchstaben, also die Insertion, ist dagegen in dieser Statistik mit gerade mal 3% sehr selten. [DMDF1936]

## 2.2 Suchmaschinen

Zur richtigen Zeit und an jedem Ort an Informationen zu gelangen, welche zu diesem Zeitpunkt von großer Notwendigkeit sind, spielt im heutigen Leben eine wichtige Rolle. Wenn zum Beispiel beim Einkauf im Supermarkt die notwendigen Zutaten für ein Rezept entfallen sind, lassen sich diese schnell und einfach mit Hilfe von Suchmaschinen nachlesen. Denn mittels dieser und internetfähigen Geräten wie Smartphones, Laptops oder Rechner, können heutzutage jegliche Informationen binnen weniger Sekunden abgerufen werden. Die dabei häufigste verwendete Suchmaschine ist Google. Mit fast 90% Marktanteil ist diese in den letzten Jahren mit Abstand die meist genutzte Suchmaschine weltweit. Wie in Abbildung 1 zu sehen ist, folgen Bing und Yahoo mit großem Abstand. Diese sind mit gerade mal 3-4% Anteil am Markt vertreten. [StatMS2016]

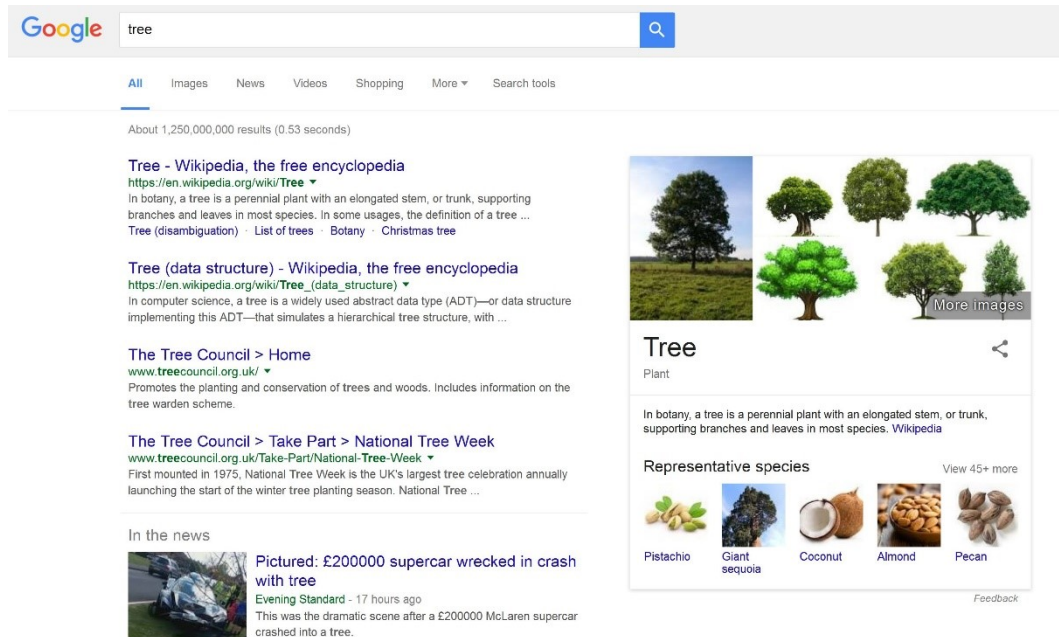


**Abbildung 1: Marktanteile der meistgenutzten Suchmaschinen nach Page Views weltweit in ausgewählten Monaten von Januar 2013 bis Februar 2016 [StatMS2016]**

Die an vierter Stelle stehende Suchmaschine Baidu ist zwar in ihrem Ursprungsland China führend, kann aber im weltweiten Vergleich die 1%-Grenze nicht übersteigen. Aus diesem Grund wird hier auf Baidu und auf weitere Suchmaschinen unter 1% nicht weiter eingegangen.

## **Google**

Die Geschichte von Google begann im Jahr 1995 mit einer Begegnung des technikbegeisterten Larry Page und Sergey Brin an der Stanford University. Ausgehend von einer akademischen Forschungsarbeit, konzipierten sie mit der Suchmaschine *BackRub* einen Vorläufer von Google. 1998 gründeten sie gemeinsam in einer Garage in Kalifornien die Firma Google Inc. Der Name leitete sich von dem englischsprachigen mathematischen Begriff „Googol“ ab. Googol steht für die Zahl 1 gefolgt von 100 Nullen ( $1^{100}$ ). Dieser steht symbolisch für die enormen Datenmengen, mit denen Google arbeitet. Finanziert wurde das Start-Up nicht wie andere durch einen Venture-Capital-Finanzierer. Stattdessen liehen sich Page und Brin das benötigte Geld von Familie und Freunden und erhielten außerdem von dem deutschen Investor Andreas von Bechtolsheim einen Scheck von 100.000 US-Dollar. So brachten sie insgesamt ein Startkapital von 1,1 Millionen US-Dollar auf. Im Gegensatz zu anderen Suchmaschinen war die Seite zunächst komplett frei von Werbung, was bei den damaligen Datenübertragungsraten von großem Vorteil für die Nutzer war. Google beendete im September 1999 die Testphase von Google mit etwa 500.000 Suchanfragen täglich und war bereits Mitte 2000 der Marktführer unter den Suchmaschinen. In den weiteren Jahren entwickelte sich das Unternehmen Google Inc. stets weiter und baute die Seite mit verschiedenen Services wie zum Beispiel Google Maps, Gmail, Google Docs, Google Books fortwährend aus. [GoCo] Das schlichte und minimalistische Design der Seite blieb jedoch stets erhalten, wie in der folgenden Abbildung 2 zu sehen ist.



**Abbildung 2: Ergebnisse bei Google für das Suchwort „tree“ [GoTree2016]**

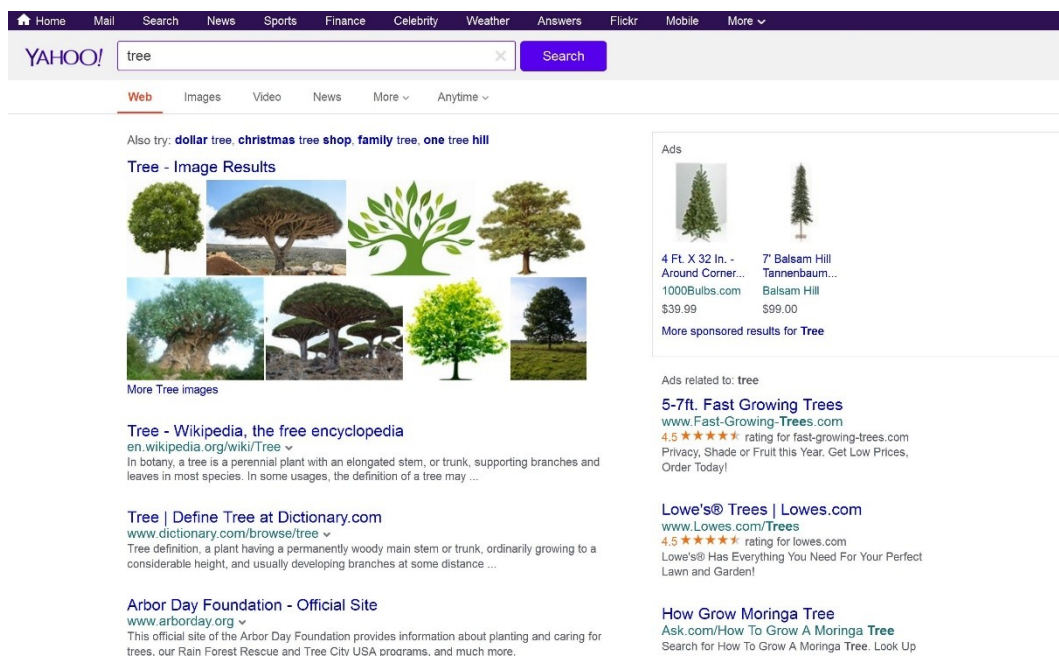
Heute werden bei Google täglich mehr als 3 Milliarden Suchvorgänge [GoGr] durchgeführt. Um diese alle zu beantworten, beginnt die Arbeit von Google zunächst mit dem Crawling von über 60 Trillionen Seiten im World Wide Web. Nachdem die Webseiten analysiert wurden, werden sie nach ihrem Inhalt und weiteren Faktoren sortiert und anschließend im Index gespeichert. Dieser fasst insgesamt über 100 Millionen Gigabyte. Mit Hilfe von Algorithmen und Formeln werden die Suchanfragen interpretiert und die passenden Dokumente aus dem Index ausgewählt. Diese werden nach mehr als 200 Faktoren sortiert, um den Nutzer die relevantesten Ergebnisse an den ersten Stellen anzubieten. [GoIn]

## Yahoo

Die erste Suchmaschine, die sich weltweit durchsetzen konnte, wurde von David Filo und Jerry Yang entwickelt. Bereits 1994, und somit ein Jahr vor den ersten Schritten Googles, arbeiteten die beiden Doktoranden der Universität Stanford an einem Hilfsmittel zur Informationssuche. Sie stellten eine kommentierte Sammlung von Internet-Adressen auf, welche zunächst



unter dem Namen „Jerry and David’s Guide to the World Wide Web“ veröffentlicht wurde. Diese Bookmark-Sammlung stieß auf reges Interesse, sodass im Herbst 1994 bereits eine Million Anfragen und 100.000 Nutzer gezählt werden konnten. 1995 erhielten Filo und Yang durch eine Venture-Capital-Firma eine Förderung von 1 Million US-Dollar und gründeten das Unternehmen Yahoo. Der Name kann als ein Begriff für „*rude, unsophisticated, uncouth*“ (zu Deutsch ungezogen, unverfälscht, ungehobelt) interpretiert werden, stellt aber außerdem im Englischen eine Abkürzung für „Yet Another Hierarchical Official Oracle“ dar. Im folgenden Jahr ging das Unternehmen an die Börse und Yahoo Deutschland wurde gegründet. Weitere länderspezifische Portale wurden in den folgenden Jahren eröffnet. Ähnlich wie Google integrierte Yahoo zahlreiche Dienste wie Kalender und Mail-Konten in ihre Seite um die Kunden an sich zu binden. Diese werden jedoch, im Gegensatz zu Google, auch bei der Auflistung von Suchergebnissen weiterhin an oberster Stelle angezeigt, wie in Abbildung 3 zu sehen.

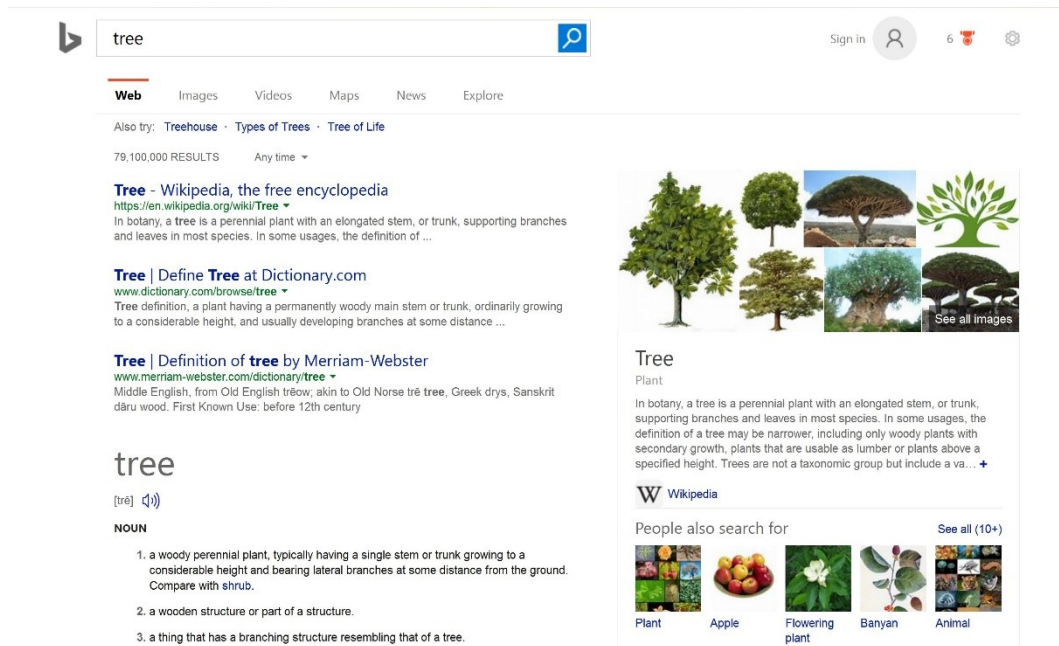


**Abbildung 3: Ergebnisse bei Yahoo für das Suchwort „tree“ [YaTree2016]**

Neben den Werbebannern wurden später auch kostenpflichtige Dienste angeboten um den Gewinn weiter zu steigern. Für ihre Suchmaschine nutzte Yahoo zunächst die Datenbestände von Altavista, Inktomi und bis 2004 sogar die von Google. Nachdem anschließend zunächst eigene Algorithmen und Indizes verwendet wurden, kooperierte Yahoo 2009 mit Microsoft. Durch diese unterscheiden sich auch die Suchergebnisse von Yahoo und Bing kaum. Für die Indizierung der Seiten und Ausgabe von Suchergebnissen nutzt Yahoo vergleichbare Rankingfaktoren wie Google. Die mobile Suche Yahoo OneSearch nutzt jedoch andere Algorithmen als die Web-Suche am PC. Hier werden an oberster Stelle Wegbeschreibungen und aktuellste Informationen zu den Suchfragen geliefert. [YaEP] [YaAb]

## **Bing**

Bing ist der Nachfolger der Suchmaschine Live Search und wurde von Microsoft Mitte 2009 das erste Mal in Betrieb genommen. Ziel war es, dem Marktführer Google Konkurrenz zu machen. [TCBi] Nach einem umfangreichen Umbau im Jahr 2012 wurde die Beta-Testphase beendet. Hierbei wurde auch erhöhter Wert auf die Integration von Social Media gelegt. [AF2012] Wie bei Google lässt sich direkt in den verschiedenen Kategorien Bildern, Videos, Karten und News suchen. Auch optisch orientiert sich Bing, abgesehen von den täglich wechselnden Hintergrundbildern auf der Startseite, stark an Google. Im direkten Vergleich zwischen Abbildung 2 und 4 ist zu erkennen, dass sich die Ergebnisseiten im Aufbau kaum unterscheiden.



**Abbildung 4: Ergebnisse bei Bing für das Suchwort „tree“ [BiTree2016]**

Beim Ranking gibt es jedoch kleinere Unterschiede. Zum einen werden Domains, welche den Suchbegriff enthalten, bei Google höher bewertet als bei Bing. Im Gegensatz dazu bevorzugt Bing kürzere Links, Startseiten gegenüber Unterseiten und legt erhöhten Wert auf die Popularität einzelner Domains sowie auf die Anzahl an Backlinks. Auch die Integration der sozialen Medien haben bei Bing einen Einfluss auf die Suchergebnisse, was bei Google nicht der Fall ist. [TS2014]

Die Funktionsweisen der drei näher betrachteten Suchmaschinen sind im Grunde identisch. Alle gehen mit Schreib- und Tippfehlern mühelos um und verbessern diese zum Teil sogar ungefragt. Auch eine automatische Vervollständigung der eingegebenen Suchanfrage findet nicht nur bei Google, sondern auch bei Yahoo und Bing statt. Ein deutlicher Unterschied lässt sich jedoch bei dem Vergleich der Anzahl an Ergebnissen erkennen. Beispielsweise für das Suchwort „tree“ liefern Yahoo und Bing mit jeweils 79.100.000 Ergebnissen eine identische Anzahl an Resultaten. Dies ist auf die Suchallianz zwischen Yahoo und Microsoft zurückzuführen. Google

hingegen liefert für das Suchwort „tree“ mit 1.260.000.000 fast die 16-fache Anzahl an Ergebnisse. An erster Stelle ist, wie auch auf den Abbildungen 2 bis 4 zu sehen, bei allen drei Suchmaschinen die Wikipedia-Seite für „tree“ zu finden. Für andere Anfragen, wie zum Beispiel „query correction“ verhält sich die Anzahl mit 2.520.000 Ergebnissen bei Yahoo und Bing und 23.200.000 Resultaten bei Google, also fast der 10-fachen Menge, ähnlich. Dies war, neben dem hohen Marktanteil von Google, ein weiterer Grund für die Nutzung dieser Suchmaschine im Verlauf der weiteren Arbeit.

## Kapitel 3

# Verwandte Arbeiten

Bereits 1965 entwickelte Wladimir Lewenstein einen Algorithmus zur automatischen Rechtschreibkorrektur. Erste Systeme waren noch auf geringen Umfang und die Nutzung eines kleinen Wörterbuches beschränkt. Später wurden für eine effektive Rechtschreibkorrektur statistisch generative Modelle mit Sprach- und Fehlermodellen genutzt. Die Rechtschreibkorrektur von Suchanfragen stellt einen Spezialfall dar und erhielt mit der Etablierung von Suchmaschinen erhöhte Aufmerksamkeit. [HD2012]

## 3.1 Vergleichbares Korpus

Für die *query spelling correction* konnte nur ein vergleichbares Korpus gefunden werden. Dieses wurde im Januar 2011 von Microsoft unter dem Namen *Speller Challenge TREC Data* im Rahmen der Microsoft Speller Challenge veröffentlicht und steht kostenlos zum Download zur Verfügung. Ziel dieses Wettbewerbs von Microsoft Research und Microsoft Bing war es, einen Algorithmus zu finden, der für jede Suchanfrage die plausibelste Alternative liefert. [EVSC2011] Für das Korpus wurde aus dem „2008 Million Query Track“ Daten-Set eine Stichprobenmenge entnommen. Von dieser Menge wurden alle Anfragen entfernt, welche eindeutig aus URLs oder E-Mail-Adressen bestanden. Die verbleibenden Anfragen wurden dann normalisiert, indem alle Buchstaben in Kleinschreibung konvertiert wurden, alle nicht-alphabetischen Schriftzeichen entfernt und die Interpunktionen durch Leerzeichen ersetzt. Die Rechtschreibung jeder einzelnen Anfrage wurde dann manuell von bis zu drei unabhängigen Experten geprüft und korrigiert. Hierbei wurden auch verschiedene Varianten der Schreibweise

zugelassen, wenn es mehrere gängige Versionen gibt. Die Datei ist wie folgt aufgebaut:

```
query <tab> suggestion1 <tab> suggestion2 ...
```

Das Korpus besteht aus insgesamt 5892 Anfragen. Davon wurden 311 als falsch geschrieben bewertet. Die Zuweisung von Vorschlägen zur Schreibweise ist wie folgt aufgebaut:

- 1122 Anfragen, die mindestens einen Vorschlag zur Schreibweise haben, welcher sich von der ursprünglichen Abfrage unterscheidet.
- 5030 Anfragen, die einen Vorschlag zur Schreibweise haben.
- 824 Anfragen, die zwei Vorschläge zur Schreibweise haben.
- 35 Anfragen, die drei Vorschläge zur Schreibweise haben.
- 3 Anfragen, die vier Vorschläge zur Schreibweise haben.

Somit wurde für fast 20% eine sich zu der ursprünglichen Anfrage unterscheidende Variante annotiert, obwohl gerade einmal etwa 5% der Anfragen als falsch bewertet wurden. [MS2011]

Der Korpus von Microsoft dient in der weiteren Arbeit als Vergleich zu dem neu entstanden *Webis Query Spelling Correction 2016 Korpus*.

## 3.2 Existierende Verfahren

Die Vielzahl an Verfahren zur automatischen Korrektur von Suchanfragen entstanden im Rahmen der Microsoft Speller Challenge. Mehr als 300 Teilnehmer beteiligten sich weltweit an diesem Wettbewerb. Neben dem in 3.1 vorgestellten Korpus wurde den Beteiligten auch der Microsoft Research Web N-gram Service, sowie ein Evaluations-Dienst zur Verfügung gestellt. Mit diesem konnte jeder seinen Algorithmus testen und direkt mit anderen Rechtschreibkorrektur-Systemen vergleichen.

[EVSC2011] Die Idee der prämierten Verfahren ist im Grunde immer übereinstimmend. Nach Eingabe einer Suchanfrage wird eine Liste von möglichen Korrekturen erstellt, welche anschließend, basierend auf ihrer Wahrscheinlichkeit, in eine Reihenfolge gebracht und ausgegeben werden. Hierfür wurde fast immer der N-gram Service von Microsoft, oder ein vergleichbarer, genutzt.

Mit einem Teilnehmer der Microsoft Speller Challenge, Peter Nalyvayko, konnte erfolgreich Kontakt hergestellt werden. Er beteiligte sich mit der auf REST basierenden Rechtschreibkontrolle „Pythia“ am Wettbewerb. REST steht für „Representational State Transfer“ und beschreibt einen fundamentalen Programmierstil, mit dem sich Programmierschnittstellen (APIs) erstellen lassen für die Kommunikation mit Webservices. [MR2015] „Pythia“ wurde implementiert als ein Java Servlet, welches über den Apache Web Server von Tomcat und unter der Nutzung von Bibliotheken von LingPipe, Apache Lucene und MT WordNet API läuft. Das Programm verarbeitet eine eingegebene Suchanfrage und gibt ein Ranking von möglichen Rechtschreibkorrekturen, geordnet nach ihrer Wahrscheinlichkeit, aus. Diese berechnet sich mit einem Noisy Channel Model, welches zuvor mit einer Liste von üblichen Schreibfehlern trainiert wurde. Um die Fehler in den Eingaben zu identifizieren wird ein Schlüssel generiert, welcher anschließend mit den Schlüsseln der Einträge im Wörterbuch auf ihre Ähnlichkeit verglichen werden. Aus der entstandenen Liste mit möglichen Kandidaten werden anschließend die plausiblen Varianten ausgewählt. Hierfür wird eine modifizierte Levenshtein-Distanz genutzt, um die Similarität der korrigierten und fehlerhaften Wörter zu berechnen. Dabei werden für die notwendigen Korrekturen verschiedene Wichtungen festgelegt. Deletion und Insertion werden mit Bearbeitungskosten von 1,2 bemessen. Substitutionen werden mit einem Wert von 2,0 und Transpositionen mit 1,0 angerechnet. In einer ersten Wertung wird die bedingte Wahrscheinlichkeit für jede Korrektur mit dem Satz von Bayes berechnet. Zuletzt greift der Algorithmus auf den Microsoft

Web N-gram Service zurück, um die berechneten Wahrscheinlichkeiten zu bewerten, zu ordnen und die besten Korrekturen auszuwählen. [PN2011]  
Dieses Programm wurde durch Peter Nalyvayko für diese Arbeit zur Verfügung gestellt und für die Evaluierung des *Webis Query Spelling Correction 2016 Korpus* genutzt.



## Kapitel 4

# Webis Query Spelling Correction 2016 Korpus

Ziel der Arbeit war es, ein umfangreiches Korpus zur Query Spelling Correction zu erschaffen, welches für jede fehlerhafte Anfrage mindestens eine Korrektur enthält. Hierfür wurde das 2010 entstandene *Webis Query Segmentation Corpus* genutzt und nochmals geprüft und verbessert. Dieses Kapitel handelt von der Entstehung und dem finalen Aufbau dieses neu entstandenen *Webis Query Spelling Correction 2016 Korpus*.

## 4.1 Korpusannotation

Im Vergleich zu dem im letzten Kapitel betrachteten *Speller Challenge TREC Data* Korpus, umfasst das hier als Grundlage genutzte *Webis Query Segmentation Corpus* mit 54.944 Anfragen fast das Zehnfache. Bei einer damals ersten Fehleranalyse durch Dr. Matthias Hagen wurde für 8.364 Anfragen genau eine Korrektur vorgenommen, was einer Fehlerquote von etwa 15% entspricht. Eine Annotation von zusätzlichen Varianten hatte zu diesem Zeitpunkt nicht stattgefunden. Im Folgenden werden die drei Schritte beschrieben, welche während der Überarbeitung des Korpus erfolgten.

### 4.1.1 Entfernung von Duplikaten

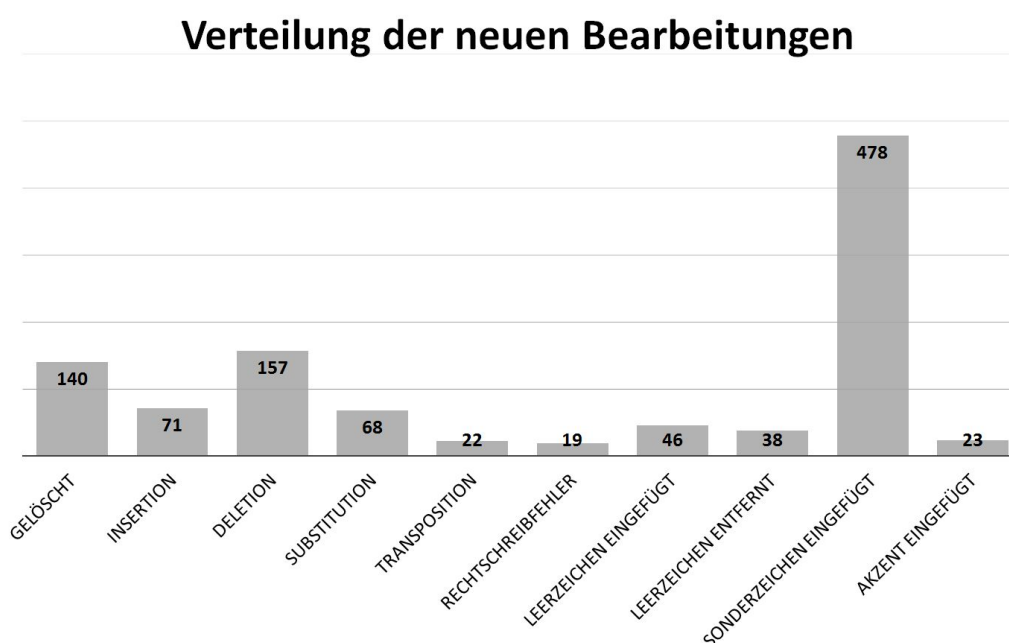
Bei der genauen Betrachtung des Korpus war aufgefallen, dass unerwünschte Dopplungen von Anfragen vorlagen. Diese galt es im ersten

Schritt zu entfernen. Hierfür wurde mit Hilfe eines Python-Skripts auf identische Zeilen geprüft, wobei für insgesamt 19 Anfragen ein Duplikat gefunden wurde. Bei diesen handelte es sich vollständig um Anfragen, bei denen eine Korrektur vorhanden war. Nach Entfernung der Duplikate umfasste das Korpus noch 54.924 Anfragen.

## 4.1.2 Semiautomatische Rechtschreibkontrolle

Ziel sollte es sein, ein möglichst fehlerfreies Korpus zu erschaffen. Aus diesem Grund wurde im zweiten Schritt nochmals eine Rechtschreibkontrolle aller verbleibenden 54.924 Anfragen vorgenommen. Unter Nutzung eines umfangreichen englischen Wörterbuchs, mit mehr als 114.000 Einträgen, und zusätzlich einer Auflistung von Marken- und Firmennamen wurden alle Anfragen erneut geprüft. Mittels Python wurde jede Zeile des Korpus durchlaufen und Wort für Wort überprüft. Ausgehend von der Editier-Distanz zwischen einem falschen Wort und einer möglichen Korrektur wird versucht, die bestmögliche Korrektur  $c$  zu finden, sodass die Wahrscheinlichkeit  $\operatorname{argmax}_c P(c|w)$ , dass  $c$  dem originalen Wort  $w$  entspricht, maximal ist. Anschließend wurde in eine Datei die Zeilennummer, gefolgt von der kompletten Anfrage, dem fehlerhaften Wort und dem Korrekturvorschlag geschrieben. Bei mehr als einem falschen Wort pro Anfrage, wurde für jeden Fehler eine neue Zeile geschrieben. Insgesamt wurden so für 20.123 Wörter Korrekturvorschläge geliefert und diese anschließend manuell geprüft. Bei Unsicherheiten, ob es sich um einen Fehler handelt und ob die Korrektur richtig ist, wurde die Suchmaschine Google zur Hilfe genommen. Fragwürdige Anfragen wurden dort nochmals geprüft und entsprechend der Suchergebnisse neu bewertet. Wenn für Anfragen bereits aus der ersten Kontrolle im Jahr 2010 Verbesserungen vorlagen, wurden diese in dem aktuellen Schritt nicht beachtet. Bei den hier annotierten Fehlern handelt es sich somit

ausschließlich um Ergänzung zu den 8.364 damaligen Korrekturen. Neben der Kontrolle von Rechtschreibfehlern wurden in diesem Schritt auch die vorhandenen Sonderzeichen geprüft und speziell bei Eigennamen im gesamten Korpus auf die richtige Schreibweise geachtet. Auch Buchstaben mit Akzenten wurden, wenn notwendig, eingefügt um eine möglichst korrekte Schreibweise garantieren zu können. Insgesamt konnten so bei weiteren 853 Anfragen Fehler gefunden und berichtigt werden. Bei mehr als der Hälfte handelte es sich dabei um die Ergänzung von Sonderzeichen. Weitere 140 Anfragen wurden gänzlich aus dem Korpus entfernt, da sie nicht der englischen Sprache angehörten, sondern beispielsweise aus dem Spanischen oder Französischen stammten. Wie in der Abbildung 5 zu sehen ist, lag für die Großzahl der restlichen hier bearbeiteten Fehler das Auslassen von Buchstaben, also eine Deletion, zugrunde.



**Abbildung 5: Fehlerverteilung der bearbeiteten Anfragen im zweiten Schritt**

Eine generelle Schwierigkeit bei der Annotation von Fehlern war die Wertung, wann eine Korrektur nicht mehr ausschließlich die Rechtschreibung, sondern auch den Intent betrifft und ob eine entsprechende Bearbeitung notwendig ist. Ein Beispiel hierfür ist die

Anfrage „think i wanna make a move lyrics“. Hier liegt kein direkter Rechtschreibfehler vor. Jedoch ist zu vermuten, dass der Nutzer auf der Suche nach den Lyrics zu dem Lied „Me & U“ von „Cassie“ war. In diesem heißt es jedoch „think i wanna make that move“. Da in diesem Fall die Unwissenheit des Nutzers über die richtige Textpassage wahrscheinlich ist, wurde diese Anfrage zu dem richtigen Inhalt „think i wanna make that move lyrics“ berichtigt.

Wie im vorgestellten Korpus der Microsoft Speller Challenge wurden jetzt auch im *Webis Query Spelling Correction 2016 Korpus* für einen Teil der Anfragen Varianten ergänzt. Dies ist zum Beispiel der Fall, wenn ausgehend von der originalen Anfrage verschiedene Korrekturen in Frage kommen. Auch wenn für ein Wort oder einen Eigennamen verschiedene Schreibweisen möglich sind, wurden diese als Varianten hinzugefügt. So wurden zum Beispiel bei Anfragen zur Band „nsync“ die ebenfalls geläufigen Schreibweisen „\*nsync“ und „‘n sync“ annotiert.

Des Weiteren wurden für Buchstabenauslassung, wie beispielsweise bei „don’t“, jeweils eine Variante mit der ausgeschriebenen Form, in diesem Fall „do not“, ergänzt. Anlage 1 zeigt alle möglichen Fälle, welche kontrolliert und wenn vorhanden im Korpus annotiert wurden. Bei dem Großteil der Anfragen fehlte das Apostroph gänzlich, sodass dieses zunächst hinzugefügt wurde. Die ausgeschriebene Variante wurde dann in der darauffolgenden Spalte angeordnet. Wenn mehr als eine ausgeschriebene Variante in Frage kam, so wurde die Anfrage individuell geprüft, welche Versionen mit dem restlichen Kontext harmonisieren. Die Tabelle in Anlage 1 zeigt außerdem für jeden Fall den regulären Ausdruck. Diese beschreiben eine Grammatik und können für eine automatisierte Annotation genutzt werden. Bei Eigennamen, wie zum Beispiel bei Titeln von Liedern, wurde entschieden, keine weitere Variante zu ergänzen, da diese nicht relevant sind.

Auch eine korrekte Annotation für die Großschreibung wurde in Betracht gezogen und begonnen. Diese erwies sich jedoch aufgrund zahlreicher Namen und Eigennamen von beispielsweise Personen, Firmen,

Kunstfiguren, Filmen, Gebäuden und Orten als außerordentlich umfangreich. Um das Wörterbuch entsprechend zu ergänzen, wurde zusätzlich eine Liste der Titel von Wikipedia-Seiten genutzt, um so möglichst viele Spezialfälle abzudecken. Eine Stichprobenkontrolle zeigte jedoch, dass eine ausschließlich automatische Korrektur so nicht möglich war. Denn bei Wikipedia werden zum Teil Eigennamen auch fälschlicherweise großgeschrieben, so wie zum Beispiel bei der Firma „adidas“. Eine fehlerfreie Annotation konnte somit ohne manuelle Kontrolle nicht garantiert werden. Daher wurde entschieden, eine Korrektur mit Großschreibung zu diesem Zeitpunkt nicht vorzunehmen.

### 4.1.3 Review der Rechtschreibkontrolle von 2010

Im letzten Schritt wurde eine erneute Kontrolle der 8.364 Korrekturen aus dem Jahr 2010 vorgenommen. Diese wurden ebenfalls unter der Nutzung von Google und der Zuhilfenahme eines englischen Wörterbuchs manuell geprüft. Bei rund 350 Anfragen konnten die damaligen Korrekturen nicht mehr nachvollzogen werden. In diesen Fällen wurde Rücksprache mit dem Verfasser der Berichtigungen, Dr. Matthias Hagen, gehalten. Für jede dieser kritischen Anfragen wurde dann individuell entschieden, ob die alte Korrektur beibehalten, bearbeitet oder gar entfernt wird. Auch hier war das Öfteren zu bewerten, ob eine Korrektur, die über die Rechtschreibkontrolle hinausgeht und den Intent der Anfrage betrifft, notwendig erscheint oder nicht. Wenn mehrere Korrekturen möglich erschienen, wurde diese als zusätzliche Variante hinzugefügt. Nach Abschluss des Reviews der Kontrolle von 2010 wurde die Bearbeitung am Korpus beendet.

## 4.2 Fertigstellung des Korpus

Im Anschluss an die abgeschlossenen Korrekturschritte wurde das Korpus fertig gestellt. Dieses ist final wie folgt in Spalten aufgebaut, wobei leere Spalten für eine bessere Übersichtlichkeit eingefügt wurden.

- **Spalte 1:** Diese enthält die originale Anfrage des Nutzers.
- **Spalte 2:** Diese enthält bei fehlerhaften Anfrage die korrigierte Variante, wobei auch fehlende Sonderzeichen hier bereits ergänzt wurden. Wenn keine Korrektur notwendig war, wird die originale Anfrage wie aus Spalte 1 übernommen, sodass in Spalte 2 immer eine korrekte Anfrage steht.
- **Spalte 3:** In dieser ist für einen Teil der Anfragen eine erste mögliche weitere Variante in der Schreibweise aufgelistet. Außerdem ist hier die ausgeschriebene Variante für Buchstabenauslassungen zu finden und bei Akzentzeichen wurden diese hier von den Vokalen entfernt. Des Weiteren wurde als eine Variante hier das Et-Zeichen „&“ durch das Wort „and“ ersetzt. Wenn es keine weitere Variante für die Anfrage gibt, so bleibt diese Spalte leer.
- **Spalte 4 und 5:** Hier sind, falls vorhanden, weitere mögliche Varianten für die Anfragen zu finden. Gibt es keine weiteren Varianten, so bleibt diese Spalte ebenfalls leer.
- **Spalte 6 und 7:** Diese wurden frei gelassen, um eine optische Trennung zu den folgenden Spalten zu schaffen.
- **Spalte 8:** In dieser Spalte wurden alle Sonderzeichen nach vorher festgelegten Regeln aus der Anfrage, ausgehen von Spalte 2, entfernt. Akzentzeichen von Vokalen wurden hier ebenfalls entfernt. In der Spalte steht somit für jede Anfrage eine Variante ohne Sonderzeichen.
- **Spalte 9, 10 und 11:** Diese Spalten folgen dem selben Ansatz wie Spalte 8, wurden jedoch ausgehen von den möglichen Varianten in

Spalte 3, 4 und 5 gefüllt. Diese Spalten können somit ebenfalls auch leer sein.

- **Spalte 12, 13 und 14:** Diese wurden wieder frei gelassen, um eine optische Abgrenzung zu der letzten Spalte zu erzeugen.
- **Spalte 15:** Diese Kommentarspalte dient ausschließlich zur Information. Hier ist für fehlerhafte Anfragen die Art der Fehler angegeben. Auch Bemerkungen zum Grund für eine Bearbeitung, zum Beispiel bei Fällen wo der Intent der Anfrage eine Rolle spielte, können hier stehen. Diese Spalte kann somit bei richtige Anfragen ebenso ungefüllt sein.

Ziel war es, dass in Spalte 2 die möglichst richtigste Schreibweise für die Anfrage steht. Dies wurde nach dem jeweiligen Eintrag im Wörterbuch und dem Google Ranking der Schreibweisen entschieden. Wenn weitere Varianten folgen, sind diese orthografisch ebenfalls richtig, aber beispielsweise weniger gebräuchlich.

Die Kommentarspalte wurde eingefügt um späteren Nutzern des Korpus gewisse Informationen mitzuteilen. Zum einen werden hier für alle Anfragen, bei denen eine Korrektur vorliegt, die analysierten Fehler angegeben. So lässt sich für jede Anfrage sofort nachvollziehen, aus welchem Grund dieses korrigiert wurde. Liegen mehrere verschiedene Fehler für eine Anfrage vor, werden diese durch Komma getrennt. Außerdem können hier weitere Hinweise, zum Beispiel zum Grund der Bearbeitung oder der Annotationen von weiteren Varianten stehen.

Die Spalten ohne Sonderzeichen und Akzentzeichen wurden ergänzt, um eine problemlose Arbeit mit dem Korpus in verschiedensten Anwendungen zu ermöglichen. Hierbei wurde wie folgt vorgegangen:

- Bindestriche - Doppelpunkte : Schrägstriche / und die öffnende Klammer ( wurden jeweils durch ein Leerzeichen ersetzt.
- Et-Zeichen & und Plus-Zeichen + wurden durch das mit Leerzeichen eingeschlossene Wort "and" ersetzt. Wenn es dadurch zu doppelten Leerzeichen kam, wurden diese anschließend wieder entfernt.

- Das Dollar-Zeichen \$ wird durch den Buchstaben "s" ersetzt
- Alle weiteren Sonderzeichen wie Kommas , Punkte . Apostrophe ' Anführungszeichen " Ausrufezeichen ! Fragezeichen ? Prozentzeichen % schließende Klammern ) und Rauten # werden durch nichts ersetzt.
- Akzente werden entfernt und durch den entsprechenden Vokal ohne Akzentstrich ersetzt.

Der resultierende Korpus fasst insgesamt 54.772 Anfragen. Bei insgesamt 9.033 Anfragen wurden Fehler gefunden und Bearbeitungen vorgenommen. Für 643 Anfragen wurde eine weitere Variante der Schreibweise hinzugefügt. Davon erhielten 13 Anfragen eine dritte und 4 Anfragen zusätzlich noch eine vierte Variante. Kommentare wurden für insgesamt 9.044 Anfragen im Korpus eingefügt.



## Kapitel 5

# Korpusanalyse

Dieses Kapitel handelt von der Analyse des neu entstandenen Korpus. Für eine bessere Bemessung erfolgte die genaue Betrachtung im Vergleich zum unter Punkt 3.1 vorgestellten *Speller Challenge TREC Data* Korpus.

## 5.1 Korpusvergleich

Zunächst wurde ein direkter Vergleich der Anfragen zwischen dem neuen *Webis Query Spelling Correction 2016 Korpus* und dem unter 3.1 vorgestellten Korpus von Microsoft vorgenommen. Hierfür wurde für jede Anfrage aus dem kleineren Korpus von Microsoft die Anfrage mit der prozentual höchsten Ähnlichkeit gesucht. Die Berechnung erfolgte mit dem *SequenceMatcher* der *difflib* Bibliothek von Python. Dieser arbeitet mit einem von Ratcliff und Obershelp im Jahr 1988 veröffentlichten „gestalt pattern matching“-Algorithmus. Die Idee dahinter ist es, die längste übereinstimmende Sequenz von Zeichen zu finden. In den verbliebenen Zeichenketten wird dann rekursiv nach weiteren identischen Sequenzen gesucht. Die Ähnlichkeit berechnet sich dann aus der doppelten Anzahl identischer Zeichen dividiert durch die Gesamtanzahl an Zeichen von beiden Anfragen. [PSLSM]

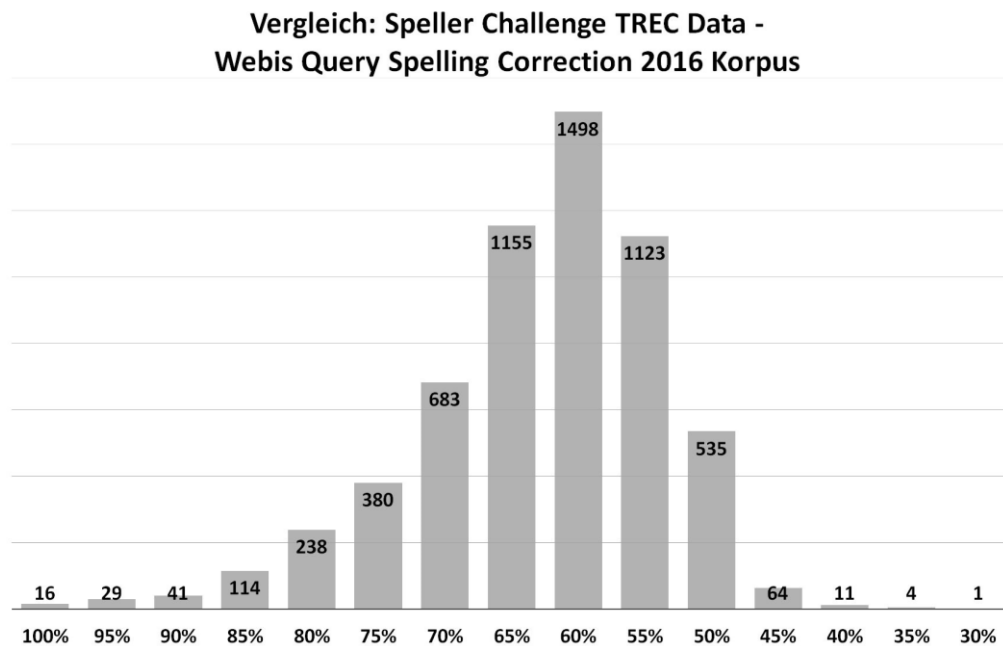
Beispiel:

Berechnung für die Ähnlichkeit der beiden Anfragen „lewisandclark“ und „lois and clark“

Übereinstimmende Zeichenketten: „clark“, „and“, „is“, „I“

$$\frac{2 * (5 + 3 + 2 + 1)}{(13 + 14)} = 0,814814 \dots \approx 81,5\%$$

Für den Vergleich der beiden Korpora ergibt sich dann die in der folgenden Abbildung 6 dargestellte Verteilung. Genau 16 Anfragen kamen in beiden Korpora gleichermaßen vor. Der Großteil der Anfragen weist eine Ähnlichkeit von etwa 60% auf.



**Abbildung 6: Vergleich von *Speller Challenge TREC Data* und *Webis Query Spelling Correction 2016 Korpus* auf Ähnlichkeit**

## 5.2 Anfragenlängenverteilung

Vergleicht man, wie in Tabelle 2 zu sehen, die beiden Korpora auf ihre Wörteranzahl der einzelnen Anfragen zeigen sich prozentual deutliche Unterschiede. Während bei dem *Webis Query Spelling Correction 2016 Korpus* über zwei Drittel der Anfragen aus drei oder vier Wörtern bestehen, wird bei dem *Speller Challenge TREC Data* Korpus das Maximum mit etwa 20% bei der Anzahl von sechs Wörtern erreicht. Anfragen mit zwei bis fünf und sieben Wörtern liegen in etwa vergleichbarer Häufigkeit vor. Auffällig ist außerdem, dass im *Webis Query Spelling Correction 2016 Korpus* fast keine Anfragen mit nur ein oder zwei Wörtern vorhanden sind. Dies ist durch die

Vorverarbeitung aus dem Jahr 2010 zu begründen, in der bewusst alle kurzen Anfragen entfernt wurden.

	<b>Webis Query Spelling Correction 2016</b>		<b>Speller Challenge TREC Data</b>	
	Absolut	Relativ	Absolut	Relativ
<b>1 Wort</b>	0	0,000%	178	3,021%
<b>2 Wörter</b>	1	0,002%	629	10,675%
<b>3 Wörter</b>	24477	44,689%	988	16,768%
<b>4 Wörter</b>	14933	27,264%	817	13,866%
<b>5 Wörter</b>	7841	14,316%	534	9,063%
<b>6 Wörter</b>	3887	7,097%	1381	23,439%
<b>7 Wörter</b>	1884	3,440%	762	12,933%
<b>8 Wörter</b>	969	1,769%	390	6,619%
<b>9 Wörter</b>	507	0,926%	195	3,310%
<b>10 Wörter</b>	273	0,498%	18	0,305%

**Tabelle 2: Vergleich von *Speller Challenge TREC Data* und *Webis Query Spelling Correction 2016 Korpus* auf Anfragenlänge**

## 5.3 Fehleranalyse

Die in Kapitel 2.1 beschriebenen Fehlerarten galt es nun im Korpus zu analysieren und auszuwerten. Hierfür wurde in einem Python-Skript die Damerau-Levenshtein-Distanz genutzt. Die Levenshtein-Distanz wurde 1965 von dem russischen Wissenschaftler Wladimir Lewenstein definiert und gibt die Editierdistanz zwischen zwei Zeichenketten an. Diese setzt sich zusammen aus der Anzahl vom Einfügen (Insertion), Löschen (Deletion) und Ersetzen (Substitution) von Zeichen. Durch eine Erweiterung von Damerau kann auch die Vertauschung von Buchstaben (Transposition) ermittelt werden. Ist die Summe Null, so waren die Zeichenketten identisch. Der Algorithmus zur Damerau-Levenshtein-Distanz lässt sich als Matrix von linearen Differenzgleichungen wie folgt darstellen, wobei  $u$  und  $v$  den eingegebenen Zeichenketten und  $m$  und  $n$  jeweils dem Betrag dieser entspricht.

$$D_{0,0} = 0$$

$$D_{i,0} = i \quad 1 \leq i \leq m$$

$$D_{0,j} = j \quad 1 \leq j \leq n$$

$$D_{i,j} = \min \begin{cases} D_{i-1,j-1} & + 0 \text{ falls } u_i = v_j \\ D_{i-1,j-1} & + 1 \text{ (Substitution)} \\ D_{i,j-1} & + 1 \text{ (Insertion)} \\ D_{i-1,j} & + 1 \text{ (Deletion)} \end{cases}$$

$$(i = 1, 1 \leq j \leq n) \vee (1 \leq i \leq m, j = 1)$$

$$D_{i,j} = \min \begin{cases} D_{i-1,j-1} & + 0 \text{ falls } u_i = v_j \\ D_{i-1,j-1} & + 1 \text{ (Substitution)} \\ D_{i,j-1} & + 1 \text{ (Insertion)} \\ D_{i-1,j} & + 1 \text{ (Deletion)} \\ D_{i-2,j-2} & + c \text{ (Transposition), falls } u_i = v_{j-1} \wedge u_{i-1} = v_j \end{cases}$$

$$2 \leq i \leq m, 2 \leq j \leq n$$

$c$  stellt dabei die Kosten dar, welche bei der Vertauschung von zwei Zeichen entstehen. Um die einzelnen Summanden der Damerau-Levenshtein-Distanz für eine genaue Fehlerbestimmung zu erhalten, muss die berechnete Matrix anschließend in einem Backtrace rekursiv durchlaufen werden. Das Python-Skript gibt dann die Anfrage, gefolgt von der Anzahl für jeden Fehler aus. Neben den vier Grundoperationen wurde das Skript zusätzlich um die Prüfung auf das Hinzufügen oder Löschen von Leer- und Sonderzeichen erweitert. Somit wurde insgesamt auf sechs verschiedene Fehler geprüft. Mit der folgenden beispielhaften Anfrage „spelimgcorrectoin“ mit der Korrektur „spelling correction“ wird dargestellt, wie jede einzelne Anfrage durch das Skript analysiert wurde.

## 1. Prüfe Leerzeichen und Sonderzeichen

Im ersten Schritt werden die Leerzeichen in den Zeichenketten gezählt und für die weitere Verarbeitung entfernt. Für die Sonderzeichen erfolgt die Prüfung gleichermaßen.

spelimgcorrectoin → spelimgcorrectoin

spelling correction → spellingcorrection

→ Die originale Anfrage enthielt keine Leerzeichen, die Korrektur genau eins. Somit lag ein Leerzeichen-Fehler vor, welcher korrigiert wurde.

## 2. Damerau-Levenshtein-Distanz-Matrix wird erstellt

Im zweiten Schritt wird die Matrix der Damerau-Levenshtein-Distanz für die Strings „spelimgcorrectoin“ und „spelling correction“ erstellt.



		s	p	e	l	l	i	n	g	c	o	r	r	e	c	t	i	o	n
s	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
p	1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
e	2	1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
l	3	2	1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
i	4	3	2	1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
m	5	4	3	2	1	1	1	2	3	4	5	6	7	8	9	10	11	12	13
g	6	5	4	3	2	2	2	2	3	4	5	6	7	8	9	10	11	12	13
c	7	6	5	4	3	3	3	3	2	3	4	5	6	7	8	9	10	11	12
o	8	7	6	5	4	4	4	4	3	2	3	4	5	6	7	8	9	10	11
r	9	8	7	6	5	5	5	5	4	3	2	3	4	5	6	7	8	9	10
r	10	9	8	7	6	6	6	6	5	4	3	2	3	4	5	6	7	8	9
r	11	10	9	8	7	7	7	7	6	5	4	3	2	3	4	5	6	7	8
e	12	11	10	9	8	8	8	8	7	6	5	4	3	3	4	5	6	7	8
c	13	12	11	10	9	9	9	9	8	7	6	5	4	3	3	4	5	6	7
t	14	13	12	11	10	10	10	10	9	8	7	6	5	4	3	3	4	5	6
o	15	14	13	12	11	11	11	11	10	9	8	7	6	5	4	3	4	4	5
i	16	15	14	13	12	12	12	12	11	10	9	8	7	6	5	4	4	4	5
n	17	16	15	14	13	13	12	13	12	11	10	9	8	7	6	5	5	4	4

Abbildung 7: Matrix der Damerau-Levenshtein-Distanz

Die Damerau-Levenshtein-Distanz der beiden Zeichenketten beträgt somit 4. Es sind somit minimal vier Operationen notwendig, um den einen String in den anderen zu überführen.

### 3. Backtrace der Matrix

Anschließend wird die Matrix rekursiv durchlaufen um die einzelnen Fehler zu ermitteln. Dabei wird für jedes Zeichen ausgegeben ob dieses gleich e (für das englische Wort *equal*) ist, oder ob eine Insertion i (in der Matrix lila dargestellt), Deletion d (in der Matrix gelb dargestellt), Substitution s (in der Matrix orange dargestellt) oder Transposition t (in der Matrix pink dargestellt) vorliegt.

```
spelimgcorrrectoin  
spellingcorrection  
eeeeedeseeeeeeeeieeetee
```

### 4. Zählen von Fehler

Im letzten Schritt werden die Fehler (i, d, s, t) im letzten String ausgezählt. Insgesamt liegen für diese Anfrage somit 1 Leerzeichen-Fehler, 0 Sonderzeichen-Fehler, 1 Insertion-Fehler, 1 Deletion-Fehler, 1 Substitutions-Fehler und 1 Transformations-Fehler vor.

Abschließend wurde die Ausgabe für den *Webis Query Spelling Correction 2016 Korpus* nochmals manuell auf Besonderheiten untersucht. Dadurch konnte bei Anfragen, bei denen eine hohe Anzahl an Fehlern vorlag, individuell beurteilt werden, wie diese zu bewerten sind. So konnte für mehr als 100 Anfragen entschieden werden, dass es sich um offensichtliche Rechtschreibfehler und nicht um Tippfehler handelt. Hatte eine Anfrage eine große Anzahl an Deletion-Fehlern, handelte es sich des Öfteren um Abkürzungen, welche dann in der Korrektur ausgeschrieben wurden. Dies umfasst insgesamt fast 2% aller Anfragen (zu sehen in Abbildung 8).

### Webis Query Spelling Correction 2016 Korpus

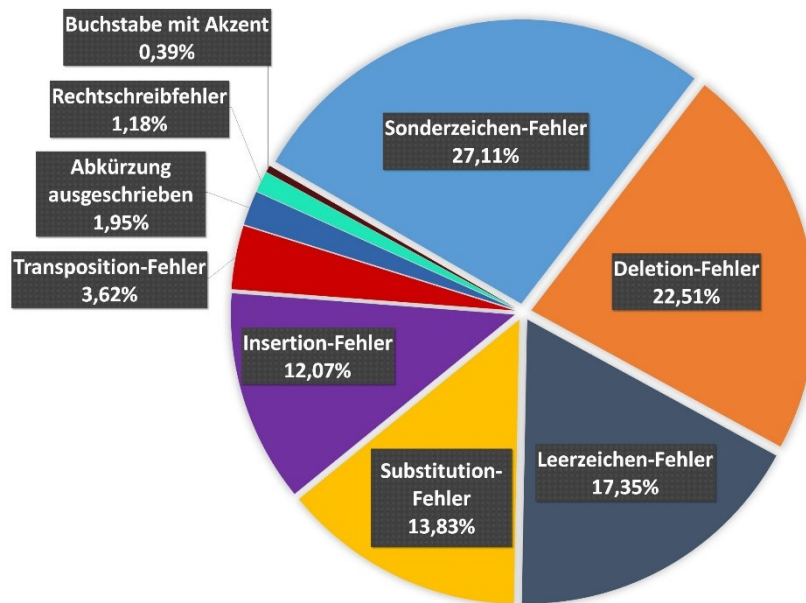


Abbildung 8: Fehlerverteilung für *Webis Query Spelling Correction 2016 Korpus*

### Speller Challenge TREC Data

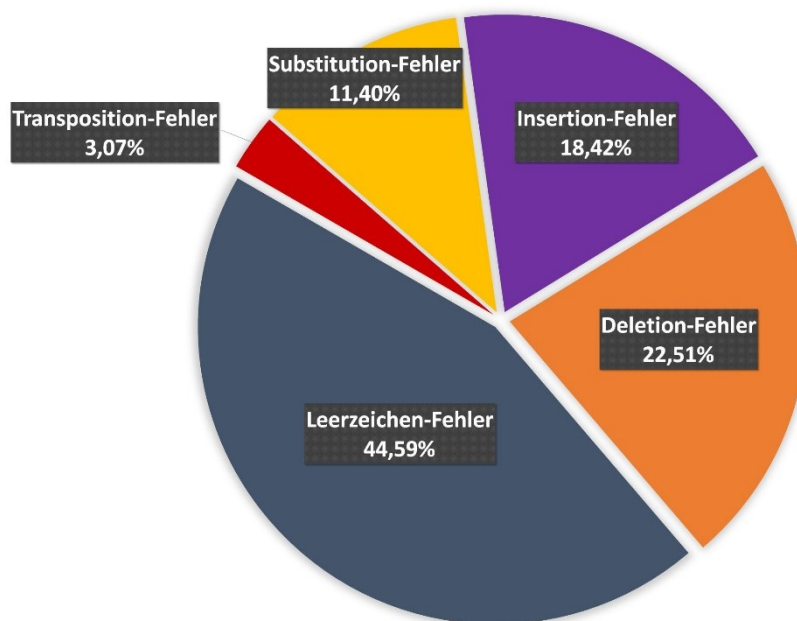


Abbildung 9: Fehlerverteilung für *Speller Challenge TREC Data Korpus*

Abbildung 8 und 9 zeigen die prozentuale Verteilung der annotierten Fehler für beide Korpora. Sonderzeichen und Akzente sind im Korpus von Microsoft nicht vorhanden. Fasst man aus diesem Grund für einen direkten Vergleich die Leer- und Sonderzeichen-Fehler im *Webis Query Spelling Correction 2016 Korpus* zusammen, so wird in beiden Korpora für diese Fehlerart ein fast identischer Anteil von rund 45% erreicht. Diese Fehler machen dementsprechend mit fast der Hälfte den größten Teil der Korrekturen aus. Deletion-Fehler traten in beiden Korpora gleichermaßen mit 22,51% auf und sind damit die häufigsten Schriftzeichen betreffenden Fehler. Die Verteilung von Substitution- und Insertion-Fehlern unterscheidet sich im *Webis Query Spelling Correction 2016 Korpus* nur minimal. Ein geringer Teil (0,39%) der Substitution-Fehler konnte durch die manuelle Nachkontrolle als das Einfügen von akzentuierten Buchstaben bewertet werden und wurde daher extra dargestellt. Im *Speller Challenge TREC Data* Korpus treten Insertion-Fehler zu etwa 2/3 öfter auf als Substitutions-Fehler. Transposition-Fehler treten in beiden Korpora mit rund 3% im gleichen Maß und vergleichsweise selten auf.



## Kapitel 6

# Evaluierung

Zuletzt wurde das neue Korpus mit dem unter Punkt 3.2 vorgestellten „Pythia“-System evaluiert. Hierfür wurden mit einem Python-Skript für jede der 54.772 Anfragen die vom System angebotenen Korrekturvorschläge mit ihrer Wahrscheinlichkeit gespeichert. Anschließend wurde verglichen, ob die im Korpus vorgenommene Korrektur auch unter den Vorschlägen des Systems zu finden ist. Allerdings werden durch „Pythia“ keine Korrekturen mit Sonderzeichen vorgenommen. Aus diesem Grund wurden alle möglichen Varianten aus dem Korpus, auch jene, bei denen die Sonderzeichen entfernt wurden, mit der Ausgabe des Systems verglichen. Die Berechnung des F-Maß, dem harmonischen Mittel zwischen Genauigkeit und Trefferquote, ermöglicht einen direkten Vergleich des *Webis Query Spelling Correction 2016 Korpus* mit dem *Speller Challenge TREC Data* Korpus.

$$F_1 = 2 * \frac{precision * recall}{precision + recall}$$

Die Genauigkeit (precision) ergibt sich als Quotient aus der Anzahl der übereinstimmenden Korrekturen dividiert durch die Anzahl an vorgeschlagenen Korrekturen durch „Pythia“. Die Trefferquote (recall) berechnet sich aus Anzahl der übereinstimmenden Korrekturen geteilt durch Anzahl der Korrekturen im Korpus.

Die folgende Tabelle 3 zeigt die Ergebnisse für das F-Maß, die Genauigkeit und die Trefferquote der beiden Korpora. Die Werte für das *Speller Challenge TREC Data* Korpus entstammen der von Peter Nalyvayko 2011 durchgeführten Evaluierung. [PN2011]

	<b>Speller Challenge TREC Data</b>	<b>Webis Query Spelling Correction 2016 Korpus</b>
<b>F-Maß</b>	0.9185730419	0,6579752269
<b>Genauigkeit</b>	0.9352770727	0,5331053872
<b>Trefferquote</b>	0.9024552090	0,8592346752

**Tabelle 3: Evaluierung von *Speller Challenge TREC Data* und *Webis Query Spelling Correction 2016 Korpus***

Ein Problem bei der Evaluierung mit „Pythia“ stellte die fehlende Korrektur von Sonderzeichen dar. Diese konnte bei der Auswertung des Korpus somit nicht berücksichtigt werden. Für eine Untersuchung, welche auch die korrekte Setzung von Sonderzeichen beachtet, ist anzunehmen, dass das neue *Webis Query Spelling Correction 2016 Korpus* im Vergleich sichtlich besser abschneiden würde. Der geringe Wert der Genauigkeit ist dadurch geschuldet, dass „Pythia“ für einen Teil der Anfragen eine hohe Anzahl an möglichen Korrekturen vorschlägt. Diese werden allerdings zu einem großen Teil als sehr unwahrscheinlich bewertet. Da jedoch die Anzahl aller möglichen Korrekturen, unabhängig ihrer Wahrscheinlichkeit, in der Berechnung dem Divisor entspricht, fallen die Werte für die Genauigkeit des *Webis Query Spelling Correction 2016 Korpus* in dieser Berechnung entsprechend geringer aus. Anhand der Trefferquote ist jedoch zu erkennen, dass das neue Korpus im Allgemeinen gute Ergebnisse erzielt. Des Weiteren wurde im Vergleich zum Korpus von Microsoft mit etwa der 36-fachen Menge an Anfragen getestet.

## Kapitel 7

# Fazit und Ausblick

Die Nutzung von Suchmaschinen ist in der heutigen Zeit allgegenwärtig. Vor allem auf Grund verschiedener Tippfehlern durch den Nutzer ist eine Rechtschreibkorrektur der Anfragen oft notwendig. Für diese Korrekturen sind aufgrund von Faktoren wie beispielsweise der Eingabe von zahlreichen Eigennamen und neomodischen Begriffen Systeme, welche ausschließlich mit klassischen Wörterbüchern arbeiten, nicht ausreichend. Die Systeme müssen sowohl mit umfangreichen Sprachmodellen als auch Fehlermodellen trainiert werden, um eine möglichst hohe Erfolgsquote zu erzielen. Hierfür sind umfangreiche Korpora notwendig. Ein Korpus dieser Art ist während dieser Arbeit entstanden. Das *Webis Query Spelling Correction 2016 Korpus* besteht aus fast 55.000 Suchanfragen, wobei alle Fehler entsprechend korrigiert wurden. Hierbei lag eine Schwierigkeit in der kontextbasierten Korrektur von Anfragen. Für jede Anfrage wurde individuell entschieden, ob eine Korrektur den Intent betreffend sinnvoll erscheint oder nicht. Zusätzlich wurden bei der Existenz von verschiedenen Schreibweisen diese als Varianten eingefügt. Das fertige *Webis Query Spelling Correction 2016 Korpus* kann nun für ein umfangreiches Training von Algorithmen zur *query spelling correction* genutzt werden.

In der Zukunft wäre zusätzlich eine korrekte Annotation von Großschreibungen wünschenswert um den Korpus noch weiter zu optimieren. Auch eine gesonderte Ergänzung von Varianten mit kontextuellen Inhalt der Anfrage wäre im Bereich des Denkbaren.

# Literaturverzeichnis

[AF2012] Andreas Floemer: *bing goes social: Microsoft will Google zeigen wie soziale Suche geht*, 11. Mai 2012, <http://t3n.de/news/bing-social-microsoft-google-387234/>, letzter Zugriff 01. Mai 2016

[BiTree2016] Bing: *tree*,  
<http://www.bing.com/search?q=tree&qs=n&form=QBRE&pq=tree&sc=10-4&sp=-1&sk=&cvid=A17A8C8F2C7C4CA194343364AD7171BE>,  
letzter Zugriff 04. Mai 2016

[BL2011] Bertrand Lisbach: *Linguistisches Identity Matching: Paradigmenwechsel in der Suche und im Abgleich von Personendaten*, 1. Auflage 2011, Vieweg+Teubner Verlag

[DMDF1936] August Dvorak, Nellie Louise Merrick, William Learned Dealey, Gertrude Catherine Ford: *Typewriting behavior: psychology applied to teaching and learning typewriting*, 1936, American book company

[DS2012] Daniel Schneider: *15 Most Misspelled Words in English*, 22. April 2012, <http://www.knetfeder.de/magazin/2012/thema/mechanische-tastaturen/>, letzter Zugriff 01. Mai 2016

[EVSC2011] Evelyne Viegas: *Speller Challenge*, 2011,  
<http://research.microsoft.com/en-us/projects/spellerchallenge/default.aspx>, letzter Zugriff 04. Mai 2016

[FAZ2014] FAZ.NET/Reuters: „*Fat Finger*“ kostet Aktienhändler 500.000 Euro, 30.01.2014, <http://www.faz.net/aktuell/finanzen/aktien/boerse-in-london-fat-finger-kostet-aktienhaendler-500-000-euro-12777839.html>, letzter Zugriff 01. Mai 2016

[GoCo] Google Company: *Google Through the Years*, <http://www.google.com/intl/en/about/company/timeline/>, letzter Zugriff 01. Mai 2016

[GoGr] Google Green: *Gesamtüberblick*, <http://www.google.com/green/bigpicture/>, letzter Zugriff 01. Mai 2016

[GoGS] Google Inc.: *Unsere zehn Grundsätze*, [https://www.google.com/intl/de\\_de/about/company/philosophy/](https://www.google.com/intl/de_de/about/company/philosophy/), letzter Zugriff 01. Mai 2016

[GoIn] Google Inside: *Alles über die Suche*, <https://www.google.de/insidesearch/howsearchworks/thestory/>, letzter Zugriff 01. Mai 2016

[GoT2015] Google Trends: *Top-Charts weltweit 2015*, <https://www.google.de/trends/topcharts#date=2015&geo>, letzter Zugriff 01. Mai 2016

[GoTree2016] Google: *tree*, [https://www.google.co.uk/search?q=tree&biw=1536&bih=731&source=lnms&sa=X&ved=0ahUKEwiJzuOs9LrMAhVBF8AKHRYdArIQ\\_AUIBigA&dpr=2.5#](https://www.google.co.uk/search?q=tree&biw=1536&bih=731&source=lnms&sa=X&ved=0ahUKEwiJzuOs9LrMAhVBF8AKHRYdArIQ_AUIBigA&dpr=2.5#), letzter Zugriff 04. Mai 2016

[GRMW] Grammar.net: *15 Most Misspelled Words in English*, 13. Januar 2011, <http://www.grammar.net/misspelledwords>, letzter Zugriff 01. Mai 2016

- [HD2012] Huizhong Duan, Yanen Li, ChengXiang Zhai, Dan Roth: *A Discriminative Model for Query Spelling Correction with Latent Structural SVM*, Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, 12.-14. Juli 2012, Korea
- [HK2015] Hope King: *How many sheets of paper would it take to print out the Internet?*, 27.04.2015, <http://money.cnn.com/2015/04/27/technology/size-of-internet-print-out/index.html/>, letzter Zugriff 01. Mai 2016
- [HPSB2011] Matthias Hagen, Martin Potthast, Benno Stein, Christof Bräutigam: *Query Segmentation Revisited*, *Proceedings of the 17th International Conference on World Wide Web*, WWW 2011, Hyderabad, India
- [MR2015] Margaret Rouse: *RESTfulAPI*, Juli 2015, <http://www.searchenterprisesoftware.de/definition/RESTful-API>, letzter Zugriff 05. Mai 2016
- [MS2011] Microsoft Research: *Speller Challenge TREC Data*, 14. Januar 2011, <http://research.microsoft.com/en-us/downloads/ff7aba09-fbb4-4201-bc98-23e2a3674e3c/>, letzter Zugriff 01. Mai 2016
- [PN2011] Peter Nalyvayko: *A REST-based Online English Spelling Checker "Pythia"*, Proceedings of the Microsoft Speller Challenge, 2011, USA
- [PSLSM] Python: *difflib — Helpers for computing deltas*, <https://docs.python.org/2/library/difflib.html>, letzter Zugriff 05. Mai 2016

[StatD2016] Statista: *Anzahl der Smartphone-Nutzer in Deutschland in den Jahren 2009 bis 2015 (in Millionen)*, 2016, <http://de.statista.com/statistik/daten/studie/198959/umfrage/anzahl-der-smartphonenuutzer-in-deutschland-seit-2010/>, letzter Zugriff 01. Mai 2016

[StatMS2016] Statista: *Marktanteile der meistgenutzten Suchmaschinen nach Page Views weltweit in ausgewählten Monaten von Januar 2013 bis Februar 2016*, 2016, <http://de.statista.com/statistik/daten/studie/225953/umfrage/die-weltweit-meistgenutzten-suchmaschinen/>, letzter Zugriff 01. Mai 2016

[StatU2016] Statista: *Anzahl der Smartphone-Nutzer in USA in den Jahren 2011 bis 2016 (in Millionen)*, 2016, <http://de.statista.com/statistik/daten/studie/285590/umfrage/anzahl-der-smartphone-nutzer-in-usa/>, letzter Zugriff 01. Mai 2016

[TCBi] TecChannel: *Microsoft startet Google-Konkurrent Bing*, 29. Mai 2009, [http://www.tecchannel.de/news/themen/business/2019307/websuche\\_microsoft\\_nimmt\\_neuen\\_anlauf\\_mit\\_bing/](http://www.tecchannel.de/news/themen/business/2019307/websuche_microsoft_nimmt_neuen_anlauf_mit_bing/), letzter Zugriff 01. Mai 2016

[TS2014] Timo Stoppacher: *SEO-Rankingfaktoren: Die Unterschiede zwischen Google, Yahoo und Bing*, 04. Februar 2014, <http://t3n.de/news/seo-rankingfaktoren-unterschiede-526070/>, letzter Zugriff 01. Mai 2016

[YaAb] Yahoo: *About*, <https://about.yahoo.com/>, letzter Zugriff 01. Mai 2016

[YaEP]            Yahoo            Presseportal:            *Unternehmensprofil*,  
http://yahoo.enpress.de/Unternehmensprofil.aspx, letzter Zugriff 01.  
Mai 2016

[YaTree2016]                            Yahoo:                            *tree*,  
https://search.yahoo.com/search;\_ylt=AwrBTzYQqCIXpJ8AAgRXNyo  
A;\_ylc=X1MDMjc2NjY3OQRfcgMyBGZyA3ImcC10LTc3NwRncHJpZ  
ANxbnU0eTJNUVRvS1ViZjJUcVF4ZHRBBG5fcnNsdAMwBG5fc3Vn  
ZwM5BG9yaWdpbgNzZWYy2gueWFob28uY29tBHBvcwMwBHBxc  
3RyAwRwcXN0cmwDBHfzdHJsAzQEcXVlcnkDdHJIZQR0X3N0bXA  
DMTQ2MjM0ODMxMw--?p=tree&fr2=sb-top-search&fr=yfp-t-777,  
letzter Zugriff 04. Mai 2016



# Anlage 1

## Tabelle für Buchstabenauslassung

Buchstaben- auslassung ohne Sonder- zeichen	Buchstaben- auslassung mit Sonder- zeichen	Ausge- schriebene Variante	weitere mögliche Variante	weitere mögliche Variante	regulärer Ausdruck
im	i'm	i am			i(?:'m  am m)
ill	i'll	i will			i(?:'ll  will ll)
ive	i've	i have			i(?:'ve  have ve)
id	i'd	i would	i had		i(?:'d  would  had d)
hes	he's	he is	he has		he(?: is's  has s)
shes	she's	she is	she has		she(?: is's  has s)
its	it's	it is	it has		it(?: is's  has s)
hed	he'd	he would	he had		he(?:'d  would  had d)
shed	she'd	she would	she had		she(?:'d  would  had d)
itd	it'd	it would	it had		it(?:'d  would  had d)
hell	he'll	he will			he(?:'ll  will ll)
shell	she'll	she will			she(?:'ll  will ll)
itll	it'll	it will			it(?:'ll  will ll)
were	we're	we are			we(?:'re  are re)
youre	you're	you are			you(?:'re  are re)
theyre	they're	they are			they(?:'re  are re)
weve	we've	we have			we(?:'ve  have ve)
youve	you've	you have			you(?:'ve  have ve)
theyve	they've	they have			they(?:'ve  have ve)
wed	we'd	we would	we had		we(?:'d  would  had d)
youd	you'd	you would	you had		you(?:'d  would  had d)
theyd	they'd	they would	they had		they(?:'d  would  had d)
well	we'll	we will			we(?:'ll  will ll)
youll	you'll	you will			you(?:'ll  will ll)
theyll	they'll	they will			they(?:'ll  will ll)
isnt	isn't	is not			is(?:'n't  not nt)
hasnt	hasn't	is not			is(?:'n't  not nt)
dont	don't	do not			do(?:'n't  not nt)
cant	can't	can not			can(?:'t  not t)
arent	aren't	are not			are(?:'n't  not nt)
aint	ain't	are not			a(?:'in't re not nt)
havent	haven't	have not			have(?:'n't  not nt)
doesnt	doesn't	does not			does(?:'n't  not nt)

couldnt	couldn't	could not			could(?:n't  not nt)
wasnt	wasn't	was not			was(?:n't  not nt)
hadnt	hadn't	had not			had(?:n't  not nt)
didnt	didn't	did not			did(?:n't  not nt)
wont	won't	will not			w(?:on't ill not ont)
werent	weren't	were not			were(?:n't  not nt)
wouldnt	wouldn't	would not			would(?:n't  not nt)
shouldnt	shouldn't	should not			should(?:n't  not nt)
mustnt	mustn't	must not			must(?:n't  not nt)
neednt	needn't	need not			need(?:n't  not nt)
mightnt	mightn't	might not			should(?:n't  not nt)
darent	daren't	dare not			dare(?:n't  not nt)
whos	who's	who is	who has	who does	who(?:'s  is  has  does s)
whod	who'd	who would	who had		who(?:'d  would  had s)
wholl	who'll	who will			who(?:'ll  will ll)
whats	what's	what is	what has	what does	what(?:'s  is  has  does s)
whatll	what'll	what will			what(?:'ll  will ll)
hows	how's	how is	how has	how does	how(?:'s  is  has  does s)
wheres	where's	where is	where has	where does	where(?:'s  is  has  does s)
whens	when's	when is	when has	when does	when(?:'s  is  has  does s)
heres	here's	here is	here has	here does	here(?:'s  is  has  does s)
theres	there's	there is	there has	there does	there(?:'s  is  has  does s)
thered	there'd	there would	there had		there(?:'d  would  had d)
therell	there'll	there will			there(?:'ll  will ll)
thats	that's	that is	that has	that does	that(?:'s  is  has  does s)