TECHNISCHE UNIVERSITÄT CHEMNITZ

Fakultät für Informatik

Professur Medieninformatik

Hauptseminar

Automatisierte Textgenerierung

Stefan Taubert

Chemnitz, den 19. November 2016

Prüfer: Prof. Dr. Maximilian Eibl
Betreuer: Christina Lohr

Abstract

Die vorliegende Arbeit soll das Thema Automatisierte Textgenerierung bearbeiten. Anfangs werden einige theoretische Grundlagen erklärt. Dazu gehören Grundverständnisse über Korpora und die Bestandteile von geschriebener Sprache. Weiterhin wird auf Segmentierung, Tokenisierung und Normalisierung eingegangen. Nach der Vorstellung einiger Methoden zur natürlichsprachlichen Textgenerierung, folgen Einsatzgebiete und verwendete Softwarelösungen der Textgenerierung. Zum Schluss der Arbeit werden Potentiale und Grenzen diskutiert, gefolgt von einer kurzen Zusammenfassung und einem Ausblick zum Thema.

Alle benutzten Internetquellen wurden am 19. November 2016 zum letzten Mal auf Aktualität geprüft.

Inhaltsverzeichnis

1 Grundlegende Aspekte der natürlichen Sprachverarbeitung (NLP)

Die natürliche Sprachverarbeitung (Natural Language Processing, NLP) besteht aus mehreren Schritten und Methoden, die im folgenden Text grundlegend erklärt werden sollen; insbesondere wird zum Schluss veranschaulicht, wie der Prozess der natürlichen Text- und Sprachgenerierung (Natural Language Generation, NLG) abläuft. Dazu werden anfangs die theoretischen Grundlagen, sowie einige Begriffe bezüglich Sprache im Allgemeinen, erklärt. Anschließend werden die notwendigen Vorverarbeitungsschritte und Methoden gezeigt.

1.1 Textkorpus

Die Verarbeitung von natürlicher Sprache basiert auf der Sammlung von Daten. Diese Datenansammlung wird als Korpus bezeichnet und kann beispielsweise aus Texten oder Sprachansammlungen bestehen [MJ00, S. 194]. Im weiteren Verlauf soll jedoch nur das sich aus Texten zusammengesetzte Korpus, das **Textkorpus**, fokussiert werden. Dieser enthält Zeitungsartikel, Literatursammlungen und weitere Texte, die durch Personen verfasst wurden. Solche Textkorpora spielen vorallem bei großen Unternehmen ein Rolle, bei kleineren Anwendungen wird in der Regel ganz auf ein Korpus verzichtet oder nur ein Minikorpus benutzt, welcher weniger Umfang als ein normaler Korpus hat. Alternativ können hier, anstelle von Korpora, auch sogenannte „Deeplearning Algorithms“ eingesetzt werden, die mit Hilfe von neuronalen Netzen Texte erzielen [Ben09, SLY11].

Um auf Textkorpora zuzugreifen, gibt es zum Beispiel das Programm „WordSmith“[1] [ST06, S. 12]. Jedoch muss nicht zwingend ein Programm für die Verwaltung genutzt werden, es gibt diverse Korpora online. Diese bieten Zugriff auf Korpora verschiedenster Sprachen. Eine Liste an einigen Korpora bietet die HU Berlin an[2]. Ein Hauptvertreter aus dem deutschen Raum ist das DWDS[3], das Digitale Wörterbuch der deutschen Sprache, welches über 1,8 Milliarden Korpusbelege aus 15 Korpora enthält.

Weitere andere genannte Vertreter sind beispielsweise:

- Projekt Deutscher Wortschatz [Qua97]

[2] `https://www.linguistik.hu-berlin.de/de/institut/professuren/korpuslinguistik/links/korpora_links`

[3] `http://www.dwds.de/`

- Hamburg Dependency Treebank [FKBM14]
- Korpus Südtirol [AAP13]
- British National Corpus (BNC) [LR+14]
- Loyola Computer-Mediated Communication Corpus
- Corpus de Référence du Français parlé
- Banca dati dell'italiano parlato (BADIP)
- Corpus OVI dell'Italiano antico (corpus TLIO)

Beispiel Die Suche nach dem Term „Informatik“ auf der Korpus Seite der Uni Leipzig[4] liefert folgende Ausgaben:

- Wort: Informatik
- Anzahl: 2193
- Häufigkeitsklasse: 12 (d.h. das Wort „der“ ist ca. 2^12 mal häufiger, als das gesuchte Wort)
- Sachgebiet: Wissenschaften
- Morphologie: in|form*at|ik
- Grammatikangaben:
  - Wortart: Substantiv
  - Geschlecht: weiblich
  - Flexion: die Informatik, der Informatik, der Informatik, die Informatik
  - Signifikante Kookkurrenzen für Informatik: Mathematik (2541.49), Technik (1575.16), Naturwissenschaften (1367.32),

1.2 Bestandteile von geschriebener Sprache

Die Sprache kann grob in drei linguistische Ebenen eingeteilt werden, welche die Bestandteile der Sprache bilden. Im folgenden sollen aufsteigend die wichtigsten Ebenen dargelegt werden.

1.2.1 Zeichen & Morphologie

Als kleinste Einheit gilt der Buchstabe, beziehungsweise das Zeichen in einer Sprache. Sie hängt direkt mit der **Morphologie** zusammen. Diese beschreibt, wie Wörter strukturiert und gebildet werden, mit zum Beispiel Silben. Das heißt sie beschreibt, wie sich Bedeutungen von Wörtern ändern, wenn einzelne Zeichen hinzugefügt oder weggelassen werden. [CEE+10, S. 12]

[4] `http://wortschatz.uni-leipzig.de/abfrage/`

1.2.2 Syntax

Die **Syntax** bildet die zweite Ebene und beschreibt, wie Sätze strukturell zusammengesetzt sind. Sie ist unerlässlich, wenn es um die Erkennung von der Grammatik eines Satzes und dessen Bedeutung geht. Sie wird in der Computerlinguistik am meisten unter die Lupe genommen, da sie dort eine der wichtigsten Rollen spielt. [CEE+10, S. 12 f]

1.2.3 Semantik

Die **Semantik** ist einer der obersten Sprachebenen, deren Ziel es ist, die Bedeutung von lexikalischen Einheiten zu beschreiben. Damit verbunden ist auch die Einordnung von ihrer Struktur und Bedeutung in einen größeren Zusammenhang. [CEE+10, S. 12 f]

Semantische Wortinformationen, die gesammelt werden, sind unter anderem:

- Anzahl der Vorkommen des Wortes im Text
- Häufigkeitsklasse des Wortes
- Synonyme des Wortes
- Morphologie des Wortes
- Sachgebiete die dem Wort zugeschrieben werden
- Grammatikangaben des Wortes:
  - Geschlecht
  - Flexion
  - Wortart

Die gesammelten Daten werden nach den durchgeführten Operationen gespeichert, damit alle diese gesammelten Daten nicht jedes Mal neu "berechnet" werden müssen. Dazu bieten sich große Datenbanken an auf welche dann mit verschiedenen Interfaces Zugriff besteht. Ein Beispiel, um auf die Informationen zuzugreifen, ist die Online Verwaltung „DWDS-Kernkorpus".

1.3 Segmentierung, Tokenization, Normalization

Bei der **Segmentierung** geht es um die Extraktion der sprachlichen relevanten Einheiten eines Dokumentes. Diese Einheiten sind beispielsweise Wörter, Absätze, Diskursabschnitte oder Sätze. Ziel ist es damit die Bearbeitung der Dokumente durch infomationstechnologische Werkzeugen zu ermöglichen. Die Aufteilung in einzelne Wörter wird dabei **Tokenization** genannt. Ein Wort ist hier eine Kombination von alphanumerischen Zeichen, die rechts und links durch ein Leerzeichen oder durch Interpunktion begrenzt wird. Ein Wort wird in diesem Zusammenhang auch **Token** genannt. Während dies erfolgt, werden die Wörter auf kanonische Formen durch

Normalisierung umgewandelt. Dabei wird zum Beispiel die Stammformbildung angewandt. [CEE+10, S. 264]

Nachstehende Textfragmente würden wie folgt umgewandelt:

- 2015/16 → 2015 und 2016
- Text-verarbeitung → Textverarbeitung
- nähmlich → nämlich

Bei diesen beiden Verfahren kann es zu einigen Problemen kommen. Beispielsweise sollte das Wort „Wort-trennung“ als „Worttrennung“, „Auf- und Abbewegung“ jedoch als „Aufbewegung“ und „Abbewegung“ erkannt werden. Genauso muss bei der Kommatrennung beachtet werden, ob es sich um eine Währungsangabe oder um ein „richtiges“ Komma handelt.

1.4 Methoden zur Generierung von natürlichsprachlichen Texten (NLG)

Nachfolgend sollen verschiedene Methoden anhand ihrer Entwicklungsgeschichte aufgezeigt werden.

1.4.1 Historisches

Bereits 1952 wurde versucht Text mit sogenannten Phrasendreschmaschinen zu generieren. Diese verknüpften Satzteile und Begriffe miteinander und korrigierten danach die Grammatik des Textes [EHR15, S. 228]. Eine weitere Methode zur Textgenerierung war das Verfertigen eines **Lückentextes** und das anschließende Einfügen von Zahlen oder Wörtern in die Lückentexte. Ein Text konnte so relativ schnell, durch Zusammenklicken, erstellt werden. Weitere Anfänge waren der **Shake 'N Bake Algorithm** von Whitelock aus dem Jahr 1988 und der **Semantic Head-Driven Generation Algorithm** von Shieber et al. aus dem Jahr 1990 [Bea92, SVNPM90]. Bei ersterem werden die einzelnen Wörter der einen Sprache in einem zweisprachigem Lexikon gesucht und übersetzt wieder semantisch zusammengefügt. Bei letzterem werden Texte mit Hilfe von semantischen nicht-monotonischen Grammatiken, Top-Down Methoden oder Links-Rekursion erzeugt. Eine andere Methode war der **Modifying A Chard For Generation Purposes Algorithm**, bei welcher Tabellen genutzt werden, um Textgenerationspfade zu schmälern, welche semantisch unvollständige Phrasen enthalten [Kay96].

1.4.2 Heutige Situation

Aus diesen historischen Anfängen haben sich unter anderem die nachfolgenden Ansätze entwickelt. Eines der wichtigsten Verfahren in der heutigen Zeit ist das **Template**

basierte Generierungsverfahren. Nachfolgend wird das Grundprinzip des Generierens auf Basis von Templates kurz erklärt.

Zuerst werden Templates erstellt, welche aus Text- oder Satzmustern bestehen und den Blickwinkel auf eine Situation festlegen. Sie sind nötig um Sätze oder Aussagen zu generieren. Um genügend Satzvarianz zu erhalten, können beliebig viele unterschiedliche Formulierungen definiert werden. Dabei sind Satzteile nicht fixiert, sondern können auch mit anderen als vertauschbar gekennzeichnet werden. Um die Texte etwas aufzubessern, können Trends, Prognosen oder weitere Einsichten im Text angereichert werden. Dies geschieht durch Verknüpfung von Analysen der Rohdaten und deren Verknüpfung mit Hintergrundinformationen. Wie detailliert und in welcher Reihenfolge Absätze oder Satzabschnitte sein sollen, kann selbst definiert werden. Durch diesen gesamten automatisierten Prozess können Texte in praktisch Echtzeit erzeugt werden [EHR15, S. 230 f].

Neben diesem Verfahren gibt es auch Verfahren, bei denen künstliche Intelligenz zum Einsatz kommt. Dabei werden neuronale Netze verwendet, die durch Simulation der menschlichen Nervenzellen möglichst „natürliche“ und abwechslungsreiche Texte generieren sollen [Nil14]. Um diese neuronalen Netze zu trainieren, werden spezielle Algorithmen verwendet [HM94]. Zudem gibt es weitere andere Algorithmen, die hier noch genannt werden sollen; zum Beispiel der **Chart Realization Algorithm For Combinatory Categorial Grammar** oder der **Handling Disjunctive Inputs Algorithm** von White aus den Jahren 2004, beziehungsweise 2006. Bei ersterem werden nacheinander 3 Methoden angewandt, welche „Argument Cluster Coordination“ und Lückenbildung benutzen [Mou06]. Bei letzterem wird das „Columbinatory Categorial Grammas“ Framework verwendet. Es wird dazu genutzt, um Umschreibungen von disjunktiven logischen Formen zu bilden. [Whi06, MW11]

2 Anwendungsmöglichkeiten & existierende Software

In diesem Kapitel sollen einige Anwendungsmöglichkeiten beleuchtet werden. Weiterhin werden kommerzielle Programme genannt. Andere kostenfreie NLG Tools wie ASTROGEN, CRISP, OpenCCG, FUF/SURGE [BME99] oder SimpleNLG sind von vielen kommerziellen Tools Bestandteil, sollen hier jedoch nicht näher betrachtet werden.[1]

2.1 Medienunternehmen & Journalismus

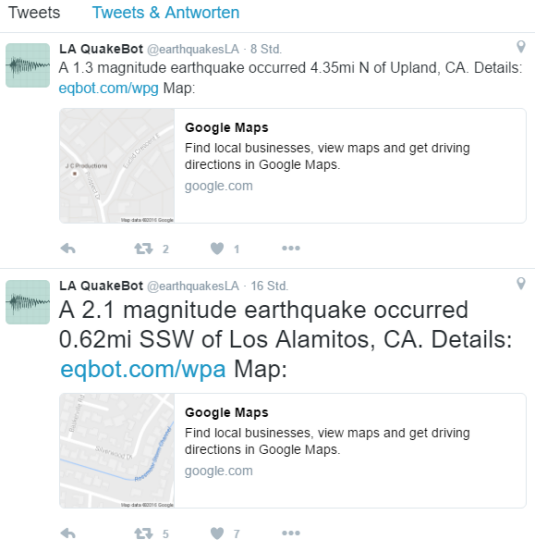
Hauptanwendung findet die automatisierte Textgenerierung in der Berichterstattung. Da die Generierung besonders bei objektiven Sachverhalten funktioniert, wird sie dort auch am meisten eingesetzt. Dazu zählen beispielsweise Berichte über Fußballergebnisse, Baseballpartien, Wetterlagen, Börsen- und Finanzgeschehen oder allgemein Sportstatistiken. Aber auch bei Patientendaten und -zusammenfassungen in Krankenhäusern, Tourismus oder bei Verkehrsmeldungen und Kurznachrichten, wie beispielsweise bei Erdbeeben der L.A. Quakebot[2] (siehe Abbildung 2.1) kommt Textgenerierung zum Einsatz.

2.2 Weitere Einsatzgebiete

Weiterhin können Artikel von E-Commerce-Anbieter in Onlineshops automatisch mit suchmaschinenoptimierten Produkttexten beschrieben werden. Ein weiterer Aspekt sind Texte über „Firmenprofilen, Personenprofilen, Veranstaltungskalendern und Veranstaltungsberichten, Immobiliendossiers“ und Arbeitszeugnisse[3] (siehe Abbildung 2.2). Weiterhin werden „Test- und Messreihen, sowie die Erstellung von Trends und Prognosen“ zusammengefasst. Aber auch Berichte über die „Auswertung des Besucherverhaltens von Internetseiten auf Basis der Zahlen, die u. a. von Google Analytics geliefert werden“. [EHR15, 228 ff] Zuletzt sei noch genannt, dass generierte Texte auch eine Vorstufe für Sprachsynthese bilden und in Dialogsystemen benutzt werden können. So könnte zum Beispiel in einem Chatbot, wie elizabot[4], nicht nur der Text

[1] `https://www.aclweb.org/aclwiki/index.php?title=Downloadable_NLG_systems`
[2] `https://twitter.com/earthquakesla`
[3] `http://www.retresco.de/automatisierung/roboterjournalismus`
[4] `http://www.masswerk.at/elizabot/`

Abbildung 2.1: Twitter Auszug des L.A. Quakebots



generiert, sondern später auch wiedergegeben werden.

2.3 Kommerzielle Software

Überall auf der Welt werden heutzutage Texte generiert. Es wird mit ungefähr 90 Millionen generierten Texten pro Tag gerechnet[5]. Dadurch haben sich viele Unternehmen auf den Bereich der Textgenerierung spezialisiert. Einige Beispiele sind Firmen wie Aexea, AX Semantics, Text-On, 2txt NLG, Retresco, Textomatic, Narrative Science, Automated Insights, Syllabs, Labsense, Arria, Tencent oder Yandex.

Die Kommunikationsagentur Aexea hat sich beispielsweise zur Aufgabe gemacht, Daten aus unterschiedlichen Quellen zu verknüpfen und daraus Beziehungen, Einflüsse, Korrelationen und Besonderheiten zu identifizieren[6].

Eine Studie zeigte, dass tausend Probanden Nachrichten aus den Bereichen Sport und Finanzen computergenerierte Texte für sachlicher und glaubwürdiger hielten als die von Hand geschriebenen. Allerdings hätten sich die von Hand verfassten Texte angenehmer lesen lassen[5].

[5] http://www.zeit.de/2015/22/roboter-journalismus-digitalisierung/komplettansicht
[6] https://www.ax-semantics.com/de/aexea/

Abbildung 2.2: Arbeitszeignis, generiert mit der „rtr textengine“ von Retresco

```
Herr Max Mustermann, geb. am 27.10.1985, war vom
   01.02.2005 bis zum 31.07.2015 als Elektroinstallateur
   in unserer Firma beschaeftigt.

Herr Mustermann verfuegt ueber ein hervorragendes und auch
    in Nebenbereichen sehr tiefes Fachwissen, das er
   unserer Firma in gewinnbringender Weise durch seine
   Arbeit zur Verfuegung stellte. Durch sein
   konzeptionelles, kreatives und logisches Denkvermoegen
   fand er fuer alle auftretenden Probleme stets
   hervorragende Loesungen. Waehrend seiner gesamten
   Beschaeftigungszeit in unserer Firma bearbeitete Herr
   Mustermann seine Aufgaben mit sehr grossem Engagement
   und beispielhaftem persoenlichem Einsatz. [...]

Wir danken Herrn Mustermann fuer die stets sehr gute,
   langjaehrige Zusammenarbeit und bedauern sein
   Ausscheiden sehr.
```

2.3.1 Generierung von natürlichsprachlichem Text

Ein Tool, um Texte zu generieren, ist **„Quill“**[7]. Quill wurde von der Firma Autmated Insights entwickelt und erstellt mit Hilfe von künstlicher Intelligenz Texte. Die künstliche Intelligenz arbeitet hier mit Hilfe von neurnalen Netzen. Das Quellkorpus der Software setzt sich zusammen aus Artikeln des Forbes Magazines und der Wall Street. Angewandt wird das Programm für die Generierung von Sportberichten, Wetterberichten oder für Berichte über Finanzmärkte. Weitere Anwendung findet das Tool für die CIA. Dies lässt sich auch aus dem Fakt schließen, dass die Firma die CIA als Hauptfinanzierer bei der Gründung hatte.

2.3.2 Kosten

Es wurden nun die verschieden angebotenen Softwarelösungen gezeigt Nun stellt sich die Frage, wie viel aktuell für einen generierten Text gezahlt werden muss. Hier lässt sich sagen, dass durchschnittlich pro 500 Wörter 10$ (ca. 9€) gezahlt wer-

[7] https://www.narrativescience.com/quill

den müssen[8]. Die oben genannte Software WordSmith kostet 78 € pro Einzellizenz. Ax-Semantics verlangt für ihre Dienstleistungen relativ hohe Preise. Sollen ca. 500 Berichte täglich generiert werden, müssen 999 € pro Monat gezahlt werden. Für Nachrichtenunternehmen bietet sich an, für 9499 € pro Monat ca. 30000 Berichte generieren zu lassen[9].

[8] https://www.welt.de/wirtschaft/article128017233/Die-Roboterjournalisten-sind-schon-unter-uns.html

[9] http://www.ax-semantics.com/de/

3 Potentiale & Grenzen

Das letzte Kapitel soll Potentiale und Grenzen der natürlichsprachlichen Textgenerierung diskutieren. Am Ende folgt eine Zusammenfassung und ein kleiner Ausblick.

3.1 Potentiale

Die Generierung von Texten mit Hilfe von Software offenbart viele neue Möglichkeiten der Informationsverbreitung. So können beispielsweise Zielgruppen, wie der lokale Fußballverein, durch kleinere Medienunternehmen besser abgedeckt werden. Gerade in Bereich der Berichterstattung von Sportarten ist, durch die oft wiederkehrenden Wortwahl, ein perfektes Szenario für Roboterjournalismus gegeben. Durch die automatisierte Texterstellung müssen solche Art von Text nicht immer neu, aufwändig erstellt, beziehungsweise formuliert werden. Dadurch können vor allem Kosten gespart werden.

Ein weiterer Aspekt ist die dauerhafte Verfügbarkeit der Computer, da diese keinen Schlaf brauchen. Sie sind quasi immer betriebsbereit. Außerdem können Sie parallel mehrere Programme laufen lassen, die an mehreren Texten arbeiten. Ein Journalist dürfte dafür mehrere Stunden brauchen. Er wird also gleichzeitig entlastet und hat mehr Zeit für Arbeiten, die nicht so einfach automatisiert werden können. Ebenso profitieren Medienunternehmen, da sie durch mehr Abdeckung verschiedenster Bereiche ein breiteres Spektrum an Texten anbieten und liefern können. Zudem müssen weniger Autoren an größeren Texten arbeiten, da generische Textpassagen von Computern übernommen werden können.

Durch den enormen Vorteil der Maschine gegenüber des Menschen, in Bezug auf Rechengeschwindigkeit, sind beinahe Echtzeitveröffentlichungen möglich. Diese Schnelligkeit ist sogar in manchen Fällen essentiell. Ein gutes Beispiel dafür ist der Quakebot[1], welcher minutengenau auf Erdbeben reagiert und einen vorgefertigten Lückentext ausfüllt. Dieser Text muss dann nur noch von einem Berichterstatter veröffentlicht werden. Durch die Skalierbarkeit von Computersystemen, sind der Texterstellung kaum Grenzen gesetzt.

Damit verbunden ist einer weiterer Vorteil der Automatisierung: gleichbleibende Qualität. Da die Texte mit einer Vorlage erstellt werden, kommt es nur zu wenigen oder sogar gar keinen Fehlern im Text. Außerdem sind die Texte auch objektiver,

[1] `http://www.heise.de/newsticker/meldung/Quakebot-schreibt-erste-Meldung-zum-Erdbeben-in-Los-Angeles-2149156.html`

da kein Autor seine Meinung einfließen lässt. Zudem können die Texte auch auf den Benutzer zugeschnitten werden. Durch zum Beispiel beobachtetes Surfverhalten, können Texte an die individuellen Vorlieben des Lesers angepasst werden.

Ein wichtiges Potential ist außerdem, dass weniger Sprachbarrieren vorhanden sind. Dadurch ist es einfacher, Texte mehrsprachig anzubieten.

Für Händler, die Produktshops betreiben, könnte eine automatisierte Erstellung der Produktbeschreibungen auch Vorteile bieten, da mittlerweile Algorithmen entwickelt wurden, die suchmaschinenoptimierte Texte verfassen. Somit ist es möglich mehr Produkte zu verkaufen, da diese ansprechender für die potentiellen Kunden sind.

3.2 Grenzen

Leider kommt die Textgenerierung neben den vielen Potentialen auch an ihre Grenzen. Zum Beispiel kann nicht mehr nachempfunden werden, was die Autoren gefühlt hätten, wenn sie den Text mit Hand geschrieben hätten. Vernünftige Interviews können von Computer auch nicht geführt werden, da ein Computer keine guten Fragen stellen kann und kein Weltwissen besitzt. Außerdem ist keine 100%ige menschliche Semantik möglich und der Computer ist größtenteils auf zahlenbasierte, regelmäßig vorkommende Ereignisse beschränkt.

Neben den technischen Grenzen muss auch beachtet werden, dass durch den vermehrten Einsatz von Computern, Arbeitsplätze ersetzt werden. Durch immer bessere Verfahren könnte es sogar bald passieren, dass professioneller Journalismus abgesetzt werden könnte.

Damit im Zusammenhang steht auch, dass den generierten Texten keine Verantwortung zugeschrieben werden kann. Es muss immer mindestens eine Person geben, die den Text freigibt und verantwortlich für eventuelle Fehler wird.

3.3 Zusammenfassung & Ausblick

Zusammenfassend lässt sich sagen, dass die automatisierte Textgenerierung seine Daseinsberechtigung hat. Sie bietet zahlreichen Vorteile und Potentiale, die noch längst nicht ausgeschöpft sind. Ein Ablösen des professionellen Journalismus' ist nicht zu erwarten, da es wahrscheinlich noch einige Jahre dauern wird, bis es möglich ist ein menschliches Gehirn zu simulieren. Durch den rasanten Anstieg an technischem Fortschritt, wird es aber viele Verbesserungen an den Algorithmen und Methoden geben. Und vielleicht wird es, selbst bei komplexeren Texten, fast nicht mehr möglich sein, einen generierten von einem handgeschriebenen Text zu unterscheiden.

Literaturverzeichnis

[AAP13] Andrea Abel, Stefanie Anstein und Stefanos Petrakis: Die Initiative Korpus Südtirol, Linguistik online, Bd. 38(2), 2013.

[Bea92] John L Beaven: Shake-and-bake machine translation, in Proceedings of the 14th conference on Computational linguistics-Volume 2, S. 602–609, Association for Computational Linguistics, 1992.

[Ben09] Yoshua Bengio: Learning deep architectures for AI, Foundations and trends® in Machine Learning, Bd. 2(1):S. 1–127, 2009.

[BME99] Regina Barzilay, Kathleen R McKeown und Michael Elhadad: Information fusion in the context of multi-document summarization, in Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics, S. 550–557, Association for Computational Linguistics, 1999.

[CEE+10] Kai-Uwe Carstensen, Christian Ebert, Cornelia Ebert, Susanne Jekat, Hagen Langer und Ralf Klabunde: Computerlinguistik und Sprachtechnologie: Eine Einführung, Springer-Verlag, 2010.

[EHR15] Börteçin Ege, Bernhard Humm und Anatol Reibold: Corporate Semantic Web: Wie semantische Anwendungen in Unternehmen Nutzen stiften, Springer-Verlag, 2015.

[FKBM14] Kilian Foth, Arne Köhn, Niels Beuck und Wolfgang Menzel: Because size does matter: The Hamburg dependency treebank, 2014.

[HM94] Martin T Hagan und Mohammad B Menhaj: Training feedforward networks with the Marquardt algorithm, IEEE transactions on Neural Networks, Bd. 5(6):S. 989–993, 1994.

[Kay96] Martin Kay: Chart generation, in Proceedings of the 34th annual meeting on Association for Computational Linguistics, S. 200–204, Association for Computational Linguistics, 1996.

[LR+14] Geoffrey Leech, Paul Rayson et al.: Word frequencies in written and spoken English: Based on the British National Corpus, Routledge, 2014.

[MJ00] James H Martin und Daniel Jurafsky: Speech and language processing, International Edition, Bd. 710, 2000.

[Mou06] François Mouret: A phrase structure approach to argument cluster coordination, in The Proceedings of the 13th International Conference on Head-Driven Phrase Structure Grammar, S. 247–267, 2006.

[MW11] Scott Martin und Michael White: Creating disjunctive logical forms from aligned sentences for grammar-based paraphrase generation, in Proceedings of the Workshop on Monolingual Text-To-Text Generation, S. 74–83, Association for Computational Linguistics, 2011.

[Nil14] Nils J Nilsson: Principles of artificial intelligence, Morgan Kaufmann, 2014.

[Qua97] Uwe Quasthoff: Projekt Der Deutsche Wortschatz, in GLDV-Jahrestagung, S. 93–99, 1997.

[SLY11] Frank Seide, Gang Li und Dong Yu: Conversational Speech Transcription Using Context-Dependent Deep Neural Networks., in Interspeech, S. 437–440, 2011.

[ST06] Mike Scott und Christopher Tribble: Textual patterns: Key words and corpus analysis in language education, Bd. 22, John Benjamins Publishing, 2006.

[SVNPM90] Stuart M Shieber, Gertjan Van Noord, Fernando CN Pereira und Robert C Moore: Semantic-head-driven generation, Computational Linguistics, Bd. 16(1):S. 30–42, 1990.

[Whi06] Michael White: Efficient realization of coordinate structures in Combinatory Categorial Grammar, Research on Language and Computation, Bd. 4(1):S. 39–75, 2006.

Abbildungsverzeichnis