Hochschule für Technik und Wirtschaft Dresden
Fakultät Informatik/ Mathematik

Abschlussarbeit zur Erlangung des akademischen Grades

Master of Science

Thema: Automatic Text Summarization
using Deep Learning and Natural Language Processing

eingereicht von: Daniel Vogel

eingereicht am: 1. Januar 2021

Erstgutachter: Prof. habil. Dr.-Ing. Hans-Joachim Böhme

Zweitgutachter: Dipl.-Kfm. Torsten Rex

Autorenreferat

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet.

Abstract

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet.

Inhaltsverzeichnis

Inhaltsverzeichnis **II**

Abbildungsverzeichnis **III**

Tabellenverzeichnis **V**

Abkürzungsverzeichnis **VII**

Formelverzeichnis **IX**

Quellcodeverzeichnis **XI**

1 Einleitung **1**
1.1 MediaInterface GmbH & SpeaKING® 1
1.2 Zielsetzung 1
1.3 Aufbau der Arbeit 2
1.4 Forschungsstand & Referenzen 2

2 Deep Learning **5**
2.1 Neuronale Netze 5
2.2 Reinforcement Learning 5
2.3 Transfer Learning 6
2.4 Architekturen 6
2.4.1 MLP 7
2.4.2 RNN 7
2.4.3 LSTM 7
2.4.4 DQN 7
2.5 Hyperparameter 7

3 Natural Language Processing **9**
3.1 Vorverarbeitung 10
3.1.1 Textbereinigung 10
3.1.2 Tokenisierung 10
3.1.3 POS-Tagging 10
3.1.4 Lemmatisierung 10
3.1.5 Entfernen von Stoppwörtern 11
3.2 Merkmalsextraktion 11
3.2.1 Übereinstimmung mit dem Titel 11

3.2.2 Satzposition 11
3.2.3 Satzähnlichkeit 11
3.2.4 Satzlänge 11
3.2.5 Domänenspezifische Wörter 11
3.2.6 Eigennamen 11
3.2.7 Numerische Daten 11

4 Datengrundlage **13**
4.1 Akquise 13
4.2 Vorverarbeitung 14
4.3 Datensatz 15

5 Abstraktiver Ansatz **17**
5.1 Architektur 17
5.2 Konfiguration 17
5.3 Training 17
5.4 Evaluation 18

6 Zusammenfassung **19**

7 Diskussion und Ausblick **21**

Literaturverzeichnis **50**

Thesen **53**

Selbstständigkeitserklärung **55**

A Erster Anhang **57**

B Zweiter Anhang **59**

Abbildungsverzeichnis

Tabellenverzeichnis

Abkürzungsverzeichnis

Formelverzeichnis

$\frac{1}{2}$ . . . . . Formel

Quellcodeverzeichnis

1 Einleitung

Notizen [Backhaus et al., 2015]:

- Zweck der Automatic Text Summarization beschreiben
- Anwendungsgebiete: Report-Generierung, Nachrichten-Zusammenfassung, Überschriften-Generierung (vgl. Paper: „Automatic Text Summarization with Machine Learning“), dadurch Reduktion der Lesezeit oder auch Entscheidungsunterstützung
- Kontext und Notwendigkeit der Arbeit im Gesundheitswesen offenlegen
- Praktischen Workflow bzw. Integration dieser Arbeit in die Praxis beschreiben
- Szenarien: Spracherkennung und Sprechererkennung auf Aufzeichnungen einer Videosprechstunde anwenden, Modelle dieser Arbeit dann für die Verdichtung der entstehenden Protokolle integrieren, Multi-Dokument-Zusammenfassung, dialogorientierte Zusammenfassungen
- Text Summarization bspw. auch als Mix aus Entity Recognition und Text Generation
- Siehe Abstract im Exposé

1.1 MediaInterface GmbH & SpeaKING®

Notizen:

- Unternehmen beschreiben
- Produkt beschreiben

1.2 Zielsetzung

Notizen:

- Ziele definieren („Automatic Single Document Summarization“, extraktiven und abstraktiven Ansatz definieren, in dieser Arbeit noch keine Zusammenfassung von Texten mit Dialogcharakter)

- Forschungsfragen formulieren

1.3 Aufbau der Arbeit

Notizen:

- Kapitel beschreiben
- Grafik als roten Faden dieser Arbeit skizzieren

1.4 Forschungsstand & Referenzen

Notizen:

- NLP-SOTA beschreiben, ggf. Übergriff zu anderen interdisziplinären Anwendungsgebieten andeuten, hier genutzte Datensätze, welche als Benchmark dienen und die zur Verbesserung des Scores genutzt werden könnten, sind: `https://paperswithcode.com/task/abstractive-text-summarization`, `https://paperswithcode.com/task/text-summarization`, also CNN/ Daily-Mail, Wikihow und Gigaword
- Vergleichbare Arbeiten beschreiben (vgl. Paper: „German Abstractive Text Summarization using Deep Learning“ und „Automatic Text Summarization“)
- Kürzlich entwickelte Ansätze (vgl. Paper: „Recent Automatic Text Summarization Techniques“)
- SOTA-Modelle recherchieren (vgl. Paper: „Automatic Text Summarization“ und weitere Architekturen aus dem Internet, ggf. auch ohne zugehöriges Paper)
- Nützliche GitHub-Repo's verlinken/ referenzieren und deren hauptsächliche Herangehensweise dokumentieren, hierfür siehe GitHub-Stars und Notizen nachfolgender Kapitel
- Medizinische Zusammenfassung: `https://github.com/armancohan/long-summarization`
- Architekturen: `https://towardsdatascience.com/deep-learning-models-for-automat` `https://medium.com/analytics-vidhya/deep-reinforcement-learning-deeprl-for-a` `https://github.com/yaserkl/RLSeq2Seq#dataset`, `https://medium.com/analytics-vi` `deep-reinforcement-learning-deeprl-for-abstractive-text-summarization-made-` `https://github.com/rohithreddy024/Text-Summarizer-Pytorch`, `https://github.com/oceanypt/A-DEEP-REINFORCED-MODEL-FOR-ABSTRACTIVE-SUMMARIZATION`, `https://github.com/theamrzaki/text_summurization_abstractive_methods`

- Möglicherweise als Paper aufnehmen, oder sogar für spätere Kapitel nutzen: `https://arxiv.org/abs/1805.11080,https://arxiv.org/pdf/1705.04304v3.pdf`, gleiche Prüfung stets auch bei andere URL's in den Notizen dieser Arbeit durchführen
- Vergleich verschiedener Modelle anhand des ROUGE-Scores: `http://nlpprogress.com/english/summarization.html`, in den Ergebnissen erwähnen, Korpus-Zusammensetzung beachten

2 Deep Learning

Notizen:

- Deep Learning definieren
- Machine Learning erwähnen
- Siehe Abstract im Exposé
- ATS on GitHub: `https://github.com/mathsyouth/awesome-text-summarization#corpus`

2.1 Neuronale Netze

Notizen:

- Neuronale Netze definieren
- Historie beschreiben
- Funktionsweise und ausgewählte Komponenten beschreiben

2.2 Reinforcement Learning

Notizen:

- Reinforcement Learning definieren. auch Deep Reinforcement Learning als Kombination aus neuronalen Netzen und Reinforcement Learning, beides Unterkapitel des Deep Learning selbst, gute Zusammenfassung zu Beginn des einen Abschnittes hier: `https://medium.com/analytics-vidhya/deep-reinforcement-learning-deeprl-for-abstracti`
- Bisherige Errungenschaften und Eigenschaften erwähnen
- Funktionsweise und ausgewählte Komponenten ggf. in Unterkapiteln beschreiben
- Unsupervised Learning, ggf. in Verbindung mit der Datengrundlage und der später beschriebenen Architektur nochmal hervorheben
- `https://www.learndatasci.com/tutorials/reinforcement-q-learning-scratch-python-opena#`

2.3 Transfer Learning

Notizen:

- Transfer Learning mit BERT hier sinnvoll, sodass das Modell die Sprache nicht in einer bestimmten Domain oder mit zu wenigen Texten neu erlernen muss
- BERT zunächst in Englisch nutzen, weil SOTA, ggf. Grafik aus Oli's VL integrieren, irgendwie in Abstractive Summarization Pipelines integrieren
- BERT ist auch multilingual, d.h. englisches Modell mit deutschem Fine-Tuning vermutlich sogar brauchbar
- Modell beschreiben, d.h. Datensätze und Parametrisierung, SOTA für verschiedene NLP-Tasks, von Kapazitäten profitieren, hat viel Kontextwissen, Fine-Tuning für eigenes Problem, d.h. domainspezifisch o.ä.
- Am besten direkt ein vortrainiertes Transformer-Modell nutzen (extra für Summarization-Tasks), BERT und RL bspw. in der Pipeline integrieren, Ziel wäre dann: Verbesserung im Score erzielen
- BERT vielleicht durch andere (teils bessere und neuere) Transformer ersetzen? Transformer in NLP recherchieren, LSTM als veraltet bezeichnen

2.4 Architekturen

Notizen:

- Existenz und Notwendigkeit verschiedener Architekturen ankündigen, ggf. in spätere Kapitel verlegen, bspw. zum abstraktiven Ansatz
- Später benötigte Architekturen hier beschreiben
- Diversität der existierenden Architekturen (wie im Forschungsstand bereits erwähnt) hervorheben
- "Reinforcement Learning comes to the rescueäus `https://towardsdatascience.com/deep-learning-models-for-automatic-summarization-4c2b89f2a9ea` einbinden
- Encoder/ Decoder, Self-Attention, Seq to Seq, Transformer Model (Recherche + Vergleich)

- Transformer, bestehend aus Seq-to-Seq-Model mit Encoder-/ Decoder-Architektur, gut erklärt: https://medium.com/inside-machine-learning/what-is-a-transformer-d07dd1fbe wissenschaftliche Paper hierzu: https://arxiv.org/abs/1706.03762, https://wiki.pathmind.com/, https://nlp.stanford.edu/pubs/emnlp15_attn.pdf, Struktur ggf. überarbeiten, d.h. langsam an Seq to Seq, Encoder, Decoder heranführen

2.4.1 MLP

Notizen:

2.4.2 RNN

Notizen:

2.4.3 LSTM

Notizen:

2.4.4 DQN

Notizen:

2.5 Hyperparameter

Notizen:

- Hyperparameter vorstellen
- Notwendigkeit und Einfluss von Hyperparametern beschreiben
- Batch-Size, e.g. Mini-Batch vs. Stochastic Batch: https://stats.stackexchange.com/questions/153531/what-is-batch-size-in-neural-network

3 Natural Language Processing

Notizen:

- Natural Language Processing definieren, e.g. Natural Language Understanding?
- NLP ist Optimierungslösung, d.h. es gibt keine eindeutige und damit im mathematischen Sinne analytische Lösung, Beispiel bei der Textzusammenfassung: Selbst Menschen können Texte auf verschiedene Arten und Weisen zusammenfassen, und verschiedene Varianten können korrekt sein
- NLU ist Teilgebiet des NLP
- Umfang der Anwendungsgebiete andeuten
- Natural Language Generation bspw. zum Generieren von Texten anhand von Stichworten benutzen, sollte bereits in gutem Zustand implementierfähig sein, möglicherweise Strukturen hiervon für die Generierung der Zusammenfassung verwenden, NLP-Links: `https://www.analyticsvidhya.com/blog/2020/08/build-a-natural-langua` `https://www.analyticsvidhya.com/blog/2019/09/introduction-to-pytorch-from-scratch/?utm_source=blog&utm_medium=Natural_Language_Generation_System_using_PyTorch`, `https://courses.analyticsvidhya.com/courses/natural-language-processing-nlp?utm_source=blog&utm_medium=Natural_Language_Generation_System_using_PyTorch`
- `https://github.com/adbar/German-NLP#Data-acquisition`
- `https://github.com/JayeetaP/mlcourse_open/tree/master/jupyter_english`
- Spacy: `https://spacy.io/usage/processing-pipelines#pipelines`
- Lemmatizer: `https://github.com/Liebeck/spacy-iwnlp`
- Transfer Learning with German BERT? `https://deepset.ai/german-bert` -¿ Modell muss die deutsche Sprache nicht alleine und von neu mit den Trainingsdaten lernen, sondern erhält einen großen Vorsprung, BERT ist Modell, welches der Transformer-Architektur nachkommt, d.h. Transformer sind bestimmte Architekturen, eventuell hiermit die Struktur dieses Kapitels überarbeiten, hier für vor allem aus meinem privaten Verzeichnis das Paper "Pre-Training of Deep Bidirectional Transformers for Language Understanding using BERT"nutzen

- GLoVe-Embeddings nutzen, weil TF-IDF etc. nicht den Kontext eines Satzes betrachten
- Supervised Learning nutzen, aber es ist eventuell nicht genug, hier kommt bspw. Transfer Learning mit BERT zur Abhilfe, zudem bspw. semi-supervised Learning mit Auto-Encoders? Self-supervised Training
- Siehe Abstract im Exposé

3.1 Vorverarbeitung

Notizen:

- Pipeline der Vorverarbeitung als Voraussetzung hervorstellen
- Relevanz von Capitalization, Punctuation, Zeilenumbrüchen klären, auch im Negativfall begründen und belegen, Satzzeichen für die Minimierung von Zwei- oder Uneindeutigkeiten berücksichtigen

3.1.1 Textbereinigung

Notizen:

- Siehe vergangene Aufgaben in GitHub

3.1.2 Tokenisierung

Notizen:

3.1.3 POS-Tagging

Notizen:

3.1.4 Lemmatisierung

Notizen:

- Lemmatisierung eventuell irrelevant, weil Wort-Tokenisierung bei modernen Architekturen und Modellen oftmals ausreicht
- Nach erfolgreichem Aufsetzen der Pipeline kann man die Eingangsdaten testweise immer noch der Lemmatisierung oder weiteren Vorverarbeitungsschritten unterziehen, um deren Auswirkungen zu messen

3.1.5 Entfernen von Stoppwörtern

Notizen:

3.2 Merkmalsextraktion

Notizen:

- Relevanz für extraktiven Ansatz beschreiben (vgl. Paper: „Automatic Text Summarization“)
- Relevanz für abstraktiven Ansatz, falls vorhanden, beschreiben
- Metriken selbst weiterentwickeln und ausreifen
- Siehe: `https://scikit-learn.org/stable/modules/feature_extraction.html`

3.2.1 Übereinstimmung mit dem Titel

Notizen:

3.2.2 Satzposition

Notizen:

3.2.3 Satzähnlichkeit

Notizen:

3.2.4 Satzlänge

Notizen:

3.2.5 Domänenspezifische Wörter

Notizen:

3.2.6 Eigennamen

Notizen:

3.2.7 Numerische Daten

Notizen:

4 Datengrundlage

Notizen:

- Modelle erfordern keine gelabelten Daten, wohl aber gesichtete Daten
- Siehe Abstract im Exposé

4.1 Akquise

Notizen:

- Data Collection: Akquise mittels Skripten in Python, zunächst mit grober Vorverarbeitung, noch nicht entsprechend der NLP-Pipeline
- Zielform der Textdateien beschreiben, Ablagestruktur ebenfalls
- Datenquellen: Wikipedia-API (`https://pypi.org/project/Wikipedia-API/`, rekursiv für 263.000 Texte), OpenLegalData-Dumps (`https://de.openlegaldata.io/pages/api/`, `https://static.openlegaldata.io/dumps/de/2019-10-21/` für 100.000 Texte), tensorflow-datasets (use latex-boxes when using bib), also `https://www.tensorflow.org/datasets/catalog/wikihow` mit 157.252 Texten, in denen Themen beantwortet werden, `https://www.tensorflow.org/datasets/catalog/gigaword` mit 3.803.957 Sätzen, `https://zenodo.org/record/1168855#.X75WfmhKiUk` mit 3.084.410 Sätzen und `https://www.tensorflow.org/datasets/catalog/cnn_dailymail` mit 287.113 Newsartikel, jeweils mit entsprechender Zusammenfassung), Unterschiede und Dokumentation siehe Excel (bspw. EN-DE)
- Datenherkünfte beschreiben, d.h. Dateiformat, Größe, Sprache etc. beschreiben
- Von den Datenquellen wird vermutet und nach manueller Einsicht bestätigt, dass Texte dort grammatikalisch korrekt sind, außerdem allgemeinsprachlich und ausreichend lang (¿ 1000 Wörter, es wird angenommen, dass 1000 Wörtern vorliegen müssen, um eine Zusammenfassung erforderlich zu machen) sind und möglichst diversifizierten Themengebieten entstammen
- Testdaten aus anderen Domänen vorbereiten und dokumentieren

- V1: Englischsprachige Korpora aus verschiedenen Branchen aus Text-Zusammenfassung-Paaren beschaffen und übersetzen

- V2: Deutschsprachige Korpora aus Text-Zusammenfassung-Paaren anfragen

- V3: Englischsprachige Korpora verwenden, um Modellarchitektur zu entwickeln und Modell zu trainieren, Adaption auf die deutsche Sprache als separates anschließendes Arbeitspaket, NLP-Vorverarbeitung überarbeiten und Modell neu trainieren

- V3 nutzen, bei Erfolg auf V1 ummünzen, oder: Eigenen deutschen Korpus aus den lokalen Agenturen aufbereiten und nutzen, Herkunft und Struktur beschreiben, Vorgang der Akquise ebenfalls, dann Fine-Tuning mit diesen Daten

4.2 Vorverarbeitung

Notizen:

- Daten iterieren, jeweils die Klassen zum Data Cleaning, Tokenisierung, Lemmatisieren etc. für einen einzelnen Text aufrufen, ggf. per weiteren Exporten zwischenspeichern, zuvor alle möglichen Dateien sichten und möglichst viele Fehler im Laufe des erneuten Exportes eliminieren, Ablageorte und Textdateiversionen beschreiben, dann Train-Test-Split, Übergabe der vorverarbeiteten Daten an die Modelle, welche den Korpus von einer Klasse namens NLP-Pipeline bekommt

- Weitergehende Besonderheiten innerhalb der Texte werden toleriert, da diese auch im Praxisbetrieb auftreten könnten und somit gekannt werden sollten

- Möglicherweise Spell Checking von Google RS für die deutsche Sprache einbinden

- Interne Pipeline: Skripte zum Herunterladen erledigen Data Cleaning, NLP-Pipeline erledigt Tokenisierung und Lemmatisierung, Lemmatisierung ausschließen, mit der Vermutung, dass neuartige Verfahren ohne viele Vorverarbeitungsschritte auskommen, außerdem Notiz zu Capitalization, Punctuation, Zeilenumbrüchen: Stark modellabhängig, tiefe Modelle wie bspw. Transformer-Architekturen (BERT) kommen damit ganz gut klar, d.h. an denen orientieren, vermutlich Plain-Text reingeben, "die machen nicht mal lower-case", Annahme: Älles was ich reinstecke, kann ein potenzielles Feature sein", andere Vorverarbeitungsschritte verfälschen das Ergebnis insofern, als dass das Training anders erfolgt, als das Modell selbst trainiert wurde

4.3 Datensatz

Notizen:

- Datengrundlage besteht aus frei verfügbaren allgemeinsprachlichen, ausreichend langen und deutschsprachigen Daten, verschiedene Herkünfte
- Auf Grundlage dieser Allgemeinsprache und den eben genannten Vorhaben, sollte ein grundlegendes Modell trainiert werden und später für den Use Case eine Art Adaptive Learning betrieben werden, d.h. wenn bekannt ist, dass das Modell für medizinische Texte angewandt werden soll, sollte man vorher die Parameter des Modells finetunen
- Später dann zwecks Adaption auch unternehmensinterne fachspezifische Daten notwendig, genauer beschreiben, perspektivisch sogar fachspezifische, dialogorientierte oder auch mehrsprachige Modelle möglich, dementsprechend mehr Daten benötigt, ggf. erst im Ausblick erwähnen
- Ähneln medizinische Texte "normalen"Texten? Gefahr: Hohe Informationsdichte bei Diktaten - "Was fällt raus?"
- Ergebnisse beim Domänenübergriff? "falsch-positiv"?
- Skript zum Einlesen entwickeln, bspw. `data_loader`
- Sätze nur in geringem Anteil verwenden, d.h. knapp unter 500.000 realen Trainings- und/ oder Testdaten

5 Abstraktiver Ansatz

Notizen:

- Abgrenzung zum extraktiven Ansatz beschreiben
- Vorteile gegenüber referenzierten Modellen herausstellen
- Generierung neuer Sätze sowohl mit vorkommenden als auch mit nicht-vorkommenden Wörtern (vgl. Paper: „Automatic Text Summarization with Machine Learning“)
- Verschiedene Ansätze `https://medium.com/analytics-vidhya/deep-reinforcement-learning-de` evtl. im Forschungsstand erwähnen
- Siehe Abstract im Exposé

5.1 Architektur

Notizen:

- Netzwerk des abstraktiven Ansatzes als Pipeline skizzieren
- Seq2Seq `https://github.com/yaserkl/RLSeq2Seq#dataset`
- Deep Reinforcement Learning (DeepRL) for Abstractive Text Summarization `https://medium.com/analytics-vidhya/deep-reinforcement-learning-deeprl-for-abstractive-tex`

5.2 Konfiguration

Notizen:

5.3 Training

Notizen:

- Training verschiedener Modelle

5.4 Evaluation

Notizen:

- Kompressionsrate messen
- Qualität der Zusammenfassung messen (BLEU `https://en.wikipedia.org/wiki/BLEU`, ROUGE `https://en.wikipedia.org/wiki/ROUGE_(metric)`, evtl. Funktionen fusionieren)
- Evaluation verschiedener Modelle mit geeigneter Vergleichstabelle
- Vergleich mit SOTA-Modellen
- Praktische Nutzung durch Implementation eines vortrainierten Modells in ein Skript oder eine Software
- Es muss eine Metrik existieren, mit der man die Genauigkeit bzw. Qualität der Zusammenfassung messen kann, d.h. man möchte die Texte nicht mit menschlich generierten Zusammenfassungen vergleichen, sondern automatisiert lernen, ggf. sollte man auch Grammatik und Inhalt separat prüfen
- For a given document there is no summary which is objectively the best. As a general rule, many of them that would be judged equally good by a human. It is hard to define precisely what a good summary is and what score we should use for its evaluation. Good training data has long been scarce and expensive to collect. Human evaluation of a summary is subjective and involves judgments like style, coherence, completeness and readability. Unfortunately no score is currently known which is both easy to compute and faithful to human judgment. The ROUGE score [6] is the best we have but it has obvious shortcomings as we shall see. ROUGE simply counts the number of words, or n-grams, that are common to the summary produced by a machine and a reference summary written by a human. `https://towardsdatascience.com/deep-learning-models-for-automatic-summarization-4c2b`
- Bei der Anwendung einer Architektur, in der das Modell durch Reinforcement Learning trainiert wird, braucht man keine massenhaft menschlich generierten Referenztexte, sondern eine wohlbedachte Kostenfunktion, der ein entsprechender Aufwand entgegen gebracht werden muss, d.h. die Herausforderung liegt beim RL eher darin, eine Umwelt und eine geeignete Funktion zum Belohnen und Bestrafen zu konstruieren, hier sind bspw. auch Evaluationsmetriken notwendig
- Rouge-Score in Python: `https://pypi.org/project/rouge-score/`
- Typisches Diagramm zur Visualisierung des Trainingsprozesses anfügen

6 Zusammenfassung

Notizen:

- Methoden und Ergebnisse zusammenfassen
- Bewertung der Zielerreichung
- Beantwortung der Forschungsfragen
- Lösung eingangs beschriebener Szenarien
- Welche Domäne wird am besten erkannt? Funktionieren Modelle mit gemischten Korpora?
- Siehe Abstract im Exposé

7 Diskussion und Ausblick

Notizen:

- Adaptive Learning für die Modelle ansatzweise vorstellen
- Modelle für mehrere Sprachen trainieren
- Modell auf Dialogcharakter adaptieren, um es in der Verdichtung von Protokollen einer Videosprechstunde zu nutzen, bzw. generell bspw. Meetings zusammenzufassen
- Forschungsstand und SOTA-Modelle hierfür im einleitenden Kapitel beschreiben, hier aufgreifen (vgl. Paper: „Abstractive Dialogue Summarization with Sentence-Gated Modeling Optimized by Dialogue Acts“ und “Using a KG-Copy Network for Non-Goal Oriented Dialogues”), bereits Architekturen vorstellen (vgl. „Automatic Dialogue Summary Generation of Customer Service“ und „Dialogue Response Generation using Neural Networks and Background Knowledge“ und „Global Summarization of Medical Dialogue by Exploiting Local Structures”)
- Modell ohne Anpassungen auf Konversationen anwenden: `https://www.isi.edu/natural-language/people/hovy/papers/05ACL-email_thread_summ.pdf`
- Modell nutzen, um Zusammenfassungen für Texte zu generieren und damit neue Datensätze für neue Modelle zu generieren, aber stark von der Qualität abhängig
- Gelb markierte Literatur sichten und verwenden, Datumsangaben aktualisieren
- Ausblick: Zusammenfassungen formatieren, also Großschreibung nach Satzenden oder auch Leerzeichenentfernung vor Punkten
- Siehe Abstract im Exposé

Literaturverzeichnis

[Backhaus et al., 2015] Backhaus, Klaus & Erichson, Bernd & Plinke, Wulff & Weiber, Rolf: Multivariate Analysemethoden, Verlag Springer Gabler, Berlin, Deutschland, 2015.

[Bird et al., 2009] Bird, Steven & Klein, Ewan & Loper, Edward: Natural Language Processing with Python, Verlag O'Reilly, Sebastopol, Vereinigte Staaten, 2009.

[Chaudhuri et al., 2019] Chaudhuri, Debanjan & Al Hasan Rony, Rashad & Jordan, Simon & Lehmann, Jens: Using a KG-Copy Network for Non-Goal Oriented Dialogues, Fraunhofer IAIS, Dresden, 2019.

[Gambhir et al., 2016] Gambhir, Mahak & Gupta, Vishal: Recent Automatic Text Summarization Techniques, University of Panjab in Chandigargh, 2016.

[Goncalves, 2020] Goncalves, Luis: Automatic Text Summarization with Machine Learning, in: https://medium.com/luisfredgs/automatic-text-summarization-with-machine-learning-an-overview-68ded5717a25, Aufruf am 01.01.2021.

[Goo et al., 2018] Goo, Chih-Wen & Chen, Yun-Nung: Abstractive Dialogue Summarization with Sentence-Gated Modeling Optimized by Dialogue Acts, University of Taiwan in Hsinchu, 2018.

[Joshi et al., 2020] Joshi, Anirudh & Katariya, Namit & Amatriain, Xavier & Kannan, Anitha: Global Summarization of Medical Dialogue by Exploiting Local Structures, in: `https://arxiv.org/pdf/2009.08666.pdf`, Aufruf am 01.01.2021.

[Kiani, 2017] Kiani, Farzad: Automatic Text Summarization, University of Arel in Istanbul, 2017.

[Kosovan et al., 2017] Kosovan, Sofia & Lehmann, Jens & Fischer, Asja: Dialogue Response Generation using Neural Networks with Attention and Background Knowledge, University of Bonn, 2017.

[Kriesel, 2005] Kriesel, David: Ein kleiner Überblick über neuronale Netze, in: `http://www.dkriesel.com/science/neural_networks`, Aufruf am 01.01.2021.

[Liu et al., 2019] Liu, Chunyi & Wang, Peng & Xu, Jiang & Li, Zang & Ye, Jieping: Automatic Dialogue Summary Generation for Customer Service, International Conference on Knowledge Discovery & Data Mining, Anchorage, Vereinigte Staaten, 2019.

[Nitsche, 2019] Nitsche, Matthias: Towards German Abstractive Text Summarization using Deep Learning, HAW Hamburg, 2019.

[Paulus et al., 2017] Paulus, Romain & Xiong, Caiming & Socher, Richard: A Deep Reinfored Model for Abstractive Summarization, in: `https://arxiv.org/pdf/1705.04304v3.pdf`, Aufruf am 01.01.2021.

[Rashid, 2017] Rashid, Tariq: Neuronale Netze selbst programmieren, Verlag O'Reilly, Sebastopol, Vereinigte Staaten, 2017.

[Raschka et al., 2019] Raschka, Sebastian & Mirjalili, Vahid: Machine Learning and Deep Learning with Python, Verlag Packt, Birmingham, Vereinigtes Königreich, 2019.

[Sinha et al., 2018] Sinha, Akash & Abhishek, Yadav & Gahlot, Akshay: Extractive Text Summarization using Neural Networks, Indian Institute of Technology in Delhi, 2018.

[Zhang et al., 2020] Zhang, Aston & Lipton, Zachary & Li, Mu & Smola, Alexander: Dive into Deep Learning, in: `https://d2l.ai/`, Aufruf am 01.01.2021.

Thesen

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet.

Selbstständigkeitserklärung

Hiermit erkläre ich, Daniel Vogel, die vorliegende Masterarbeit selbstständig und nur unter Verwendung der von mir angegebenen Literatur verfasst zu haben. Die aus fremden Quellen direkt oder indirekt übernommenen Gedanken sind als solche kenntlich gemacht. Diese Arbeit hat in gleicher oder ähnlicher Form keiner anderen Prüfungsbehörde vorgelegen.

Dresden, den 1. Januar 2021

Daniel Vogel

A Erster Anhang

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet.

B Zweiter Anhang

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet.