Fachhochschule Graubünden
University of Applied Sciences

Churer Schriften
zur Informationswissenschaft

Herausgegeben von
Wolfgang Semar Bernard Bekavac, Ivo Macek, Armando Schär

Arbeitsbereich Master of Science
in Business Administration, Information and Data Management

Schrift 156

Deep Learning für Part-of-Speech-Tagging

Vergleich eines auf Transformers basierenden POS-Taggers mit bestehenden Modellen

Marina Lea Schürmann

Chur 2023

Churer Schriften zur Informationswissenschaft

Herausgegeben von Wolfgang Semar,
Bernard Bekavac, Ivo Macek, Armando Schär

Schrift 156

Deep Learning für Part-of-Speech-Tagging

Vergleich eines auf Transformers basierenden POS-Taggers mit bestehenden Modellen

Marina Lea Schürmann

Diese Publikation entstand im Rahmen einer Thesis zum Master of Science FHGR in Business Administration, Information and Data Management.

Referent: Prof. Dr. Albert Weichselbaum

Korreferent: Prof. Dr. Rolf Assfalg

Verlag: Fachhochschule Graubünden

ISSN: 1660-945X

Ort, Datum: Chur, Februar 2023

Abstract

POS-Tagger bestimmen die Wortarten in einem Text. Verschiedene Verfahren konnten die Genauigkeit immer weiter verbessern, so dass der Benchmark heute bei 97% korrekt klassierter Wörter liegt. Der Wert vollständig korrekter Sätze ist jedoch um einiges tiefer, was Folgen für die Qualität darauf aufbauender Aufgaben hat.

Im Deep Learning entstanden vielversprechende Algorithmen, die Sequenzen effektiv modellieren können. Transformers-Modelle konnten in jüngster Zeit mit ihrem Aufmerksamkeitsmechanismus grosse Fortschritte in der natürlichen Sprachverarbeitung erzielen.

Ziel dieser Arbeit war es herauszufinden, ob Transformers das POS-Tagging auf Satzebene verbessern kann. Dafür wurde ein auf dem vortrainierten DistilBERT-Modell basierender POS-Tagger in drei Sprachen implementiert und mit bestehenden Taggern verglichen. Die Ergebnisse zeigten, dass die Genauigkeit auf Satzebene durch ein Transformers-Modell stark ansteigt, es aber Unterschiede zwischen den Wortarten und Sprachen gibt.

Schlagwörter: POS-Tagging, NLP, Deep Learning, Transformers, Sequence Labeling

POS taggers determine the part of speech in a text. Different methods have improved the accuracy so that the benchmark today is 97% of correctly classified words. However, the accuracy of completely correct sentences is much lower, which has consequences for the quality of tasks based on this.

In Deep Learning, promising algorithms emerged which are capable of effectively modeling sequences. Transformers models have recently made great progress in natural language processing with their at- tention mechanism.

The goal of this thesis was to find out whether Transformers can improve POS tagging at the sentence level. For this purpose, a POS tagger based on the pre-trained DistilBERT model was implemented in three languages and compared with existing taggers. The results showed that accuracy at the sentence level increase greatly with a Transformers model, but there are differences between word types and languages.

Les étiqueteurs grammaticaux déterminent la nature grammaticale des mots dans un texte. Différentes méthodes ont permis d'améliorer la précision, de sorte que l'indice de référence est aujourd'hui de 97% de mots correctement classés. La performance des phrases entièrement correctes est toutefois bien plus basse, ce qui entraîne des conséquences sur la qualité des tâches qui en découlent.

L’apprentissage en profondeur a donné naissance à des algorithmes prometteurs capables de modéliser efficacement des séquences. Les modèles Transformers ont récemment fait de grands progrès dans le traitement du langage naturel grâce à leur mécanisme d'attention.

L'objectif de ce travail était de déterminer si Transformers pouvait améliorer l'étiquetage morpho-syn- taxique au niveau de la phrase. Pour cela, un étiqueteur grammatical basé sur le modèle pré-entraîné

« DistilBERT » a été implémenté dans trois langues et comparé avec des étiqueteurs existants. Les résultats ont montré que la précision au niveau de la phrase est fortement améliorée par un modèle Transformers, mais qu'il y a des différences entre les types de mots et les langues.

Danksagung

Ich möchte mich bei allen Personen, die mich unterstützt haben, herzlichst bedanken. Insbesondere möchte ich Herrn Albert Weichselbraun für seine Betreuung, die guten Hinweise und nützlichen Tipps danken, die das Schreiben dieser Arbeit erleichtert haben.

Ein grosses Merci gilt auch denjenigen, die diese Arbeit kritisch gegengelesen und so zu ihrer Verbesserung beigetragen haben.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

1 Einführung

Wortarten (englisch: *Part-of-Speech, POS*) bestimmen die grammatikalische Zugehörigkeit eines Wortes und helfen, einen Satz inhaltlich zu verstehen oder die Satzstruktur zu analysieren. Die Bestimmung der Wortarten in einem Text ist ein wichtiger Vorbereitungsschritt für verschiedene Aufgaben der natürlichen Sprachverarbeitung. Zum Beispiel für das automatische Zusammenfassen von Texten oder die maschinelle Übersetzung. POS-Tagger sind Algorithmen, die den Wörtern automatisiert ihre wahrscheinlichste Wortart zuweisen. (Jurafsky & Martin, 2022, S. 162–163)

Seit Brill (1992) einen der ersten automatischen POS-Tagger vorgestellt hat, wurde eine Vielfalt an Modellen basierend auf unterschiedlichen Techniken, entwickelt. In den letzten fünf Jahren wurde insbesondere an POS-Taggern basierend auf Deep Learning geforscht. Trotzdem erreicht der auf einem statistischen Verfahren basierende Stanford-Tagger im Vergleich mit anderen Taggern auf dem heutigen Forschungsstand Höchstwerte («POS Tagging (State of the art)», 2019). In diesem Zusammenhang sollen die nachfolgenden Unterkapitel das Forschungsproblem und das Ziel dieser Masterarbeit näher erläutern.

1.1 Forschungsproblem

Heutzutage erreichen POS-Tagger Accuracy-Werte von über 97% («POS Tagging (State of the art)», 2019). Diese Zahl ist jedoch mit Vorsicht zu geniessen, da sie den Durchschnitt über alle Wörter abbildet (Manning, 2011). Der Anteil vollständig richtig getaggter Sätze ist deutlich tiefer (ebd.). Je nach darauf aufbauender Aufgabe kann dies zu gravierenden Einbussen in deren Ergebnis führen (ebd.). Zu beachten ist zudem, dass ein POS-Tagger, der Tags rein nach der statistischen Häufigkeit einer Wort-Tag-Kombination zuteilt, für Englisch bereits eine Accuracy von 90% für bekannte (d.h. im Trainingsdatensatz vorhandene) Wörter erzielt (Manning, 2021). Streng genommen konnte somit seit dem Brill-Tagger (Brill, 1992), als einem der ersten transformationsbasierten POS-Tagger, erst eine Verbesserung von 13% erzielt werden.

Language Models beschreiben die Wahrscheinlichkeit einer bestimmten Abfolge von Wörtern. In herkömmlichen POS-Taggern kommen meist n-gram-Modelle zur Anwendung. Diese berechnen die Wahrscheinlichkeit von n aufeinanderfolgenden Wörtern, berücksichtigen aber keinen Kontext wie die Position im Satz oder die Verwandtschaft zu ähnlichen Wörtern. (Jurafsky & Martin, 2022)

Der Stanford-Tagger, der mit einer Genauigkeit von 97.3% als Benchmark-Tagger gilt («POS Tagging (State of the art)», 2019), basiert auf einem statistischen Verfahren (Maximum-Entropie-Methode), das die Abhängigkeiten innerhalb einer Sequenz von beiden Seiten her berücksichtigt (zyklisches Abhängigkeitsnetz) (Toutanova et al., 2003). Seit der Stanford-Tagger 2003 vorgestellt wurde, hat sich im Machine Learning und in der natürlichen Sprachverarbeitung viel getan.

Insbesondere im Deep Learning-Bereich entstanden vielversprechende Algorithmen, die Sequenzen effektiver verarbeiten können. Somit können sie die Position eines Wortes berücksichtigen und Sprachmodelle besser als statistische Machine Learning-Verfahren abbilden. Embedding-Algorithmen bilden die Bedeutung eines Wortes als Vektor in einem Vektorraum ab (Jurafsky & Martin, 2022). Dem Sprachmodell kann damit zusätzlich zur Wortsequenz eine tiefere, semantische Bedeutung der einzelnen Sequenzelemente mitgegeben und Wortfeatures können besser abgebildet werden.

Der Stanford-Tagger kann durch seine statistische Methode jedoch Language Models und Word Embeddings nur unzureichend abbilden. In letzter Zeit wurde vor allem auf die von Hochreiter und Schmidhuber (1997) entwickelten Long-Short-Term-Memory-Netzwerke (LSTM) für die Verbesserung von POS-Tagging-Problemen gesetzt. Die Zellen dieses Netzwerks können Information für später speichern. Allerdings schwindet dieses «Gedächtnis» je länger die Sequenz wird und je weiter die voneinander abhängigen Elemente auseinander liegen (Culurciello, 2019; Dirac, 2019). Zudem wird eine enorme Rechenleistung für das Trainieren verlangt und für jede konkrete Aufgabe wird ein genügend grosser, eigens dafür gelabelter Trainingsdatensatz benötigt (ebd.).

1.2 Forschungsziel

Der aktuelle State-of-the-Art der POS-Tagger erreicht nur scheinbar einen hohen Wert, da einzelne Wörter und nicht ganze Sätze für die Berechnung der Genauigkeit berücksichtigt werden (Manning, 2011). Mittels Deep Learning können Sequenzen effektiver modelliert und Language Models besser abgebildet werden. Zudem können mit Word oder Subword Embeddings Wörter, bzw. Teilwörter mit mehr semantischem Kontext in einen Vektor überführt werden. All dies könnte zu einer Verbesserung der Tagging-Genauigkeit für einzelne Wörter, aber insbesondere auch für ganze Sätze führen. Basierend auf Forschungsergebnissen aus dem Jahr 2003, berücksichtigt der Stanford-Tagger (Toutanova et al., 2003) diese Elemente erst unzureichend. Ziel dieser Arbeit war es daher, einen POS-Tagger zu bauen, der die neuen Forschungserkenntnisse miteinbezieht und so die

Tagging-Genauigkeit über einzelne Wörter, insbesondere jedoch über ganze Sätze, verbessert.

In den letzten Jahren konnten vor allem die Transformers-Modelle grosse Fortschritte in *Natural Language Processing* (NLP) Themen erzielen. Transformers basiert auf dem Aufmerksamkeitsmechanismus, was sie für Sprachsequenzen, in denen Wörter voneinander abhängen, besonders geeignet macht (Vaswani et al., 2017). Anstatt jedes Wort isoliert in der Sequenz zu betrachten, wird es im Kontext aller relevanten Wörter verarbeitet (ebd.). Transfer-Lernen erlaubt es zudem, ein vortrainiertes Transformers-Modell auch mit einem kleinen Datensatz für die gewünschte Anwendung zu adjustieren (Tunstall et al., 2022, Kapitel 1).

Somit folgt diese Masterarbeit folgender Forschungsfrage:

Welche Ergebnisse erreicht ein auf Transformers basierender POS-Tagger für Deutsch, Französisch und Englisch im Vergleich zu bestehenden Taggern?

Um dieses Ziel zu erreichen, strebt die Masterarbeit folgende Unterziele an:

1. Es soll ein State-of-the-Art zu den in den letzten fünf Jahren verwendeten POS-Tagging-Methoden erstellt werden.
2. Es soll ein Überblick über die für POS-Tagging geeigneten Deep Learning-Methoden gegeben werden.
3. Mittels der Transformers-Library soll ein auf Transformers basierender POS-Tagger entwickelt und implementiert werden.
4. Der implementierte POS-Tagger soll für Deutsch, Französisch und Englisch mittels der Textkorpora der Universal Dependencies evaluiert werden
    a. Dabei soll die Accuracy für den neuen POS-Tagger auf Wort- und Satzebene mit dem Stanford-Tagger, dem SpaCy-Tagger und dem empfohlenen Tagger von NTLK verglichen werden,
    b. sowie eine qualitative Evaluierung über konkrete Beispiele, die besser und schlechter als vom Stanford-, SpaCy- und NLTK-Tagger getaggt werden, durchgeführt werden.
5. Es soll ein Ausblick gegeben werden, wie die Ergebnisse des vorgestellten POS-Taggers noch weiter verbessert werden könnten.

2 Methodik

Die Erarbeitung der Masterarbeit setzte sich aus den folgenden drei Etappen zusammen:

Abb. 1: Forschungsdesign (eigene Grafik)

Die Literaturrecherche erlaubte es, einen Überblick über bestehende POS-Tagging-Methoden zu erhalten und deren Unterschiede zu kennen. Aufgrund dieser Erkenntnisse konnte die Entwicklung des eigenen POS-Taggers begründet werden, so dass dieser in einem nächsten Schritt implementiert werden konnte. Der entwickelte POS-Tagger wurde zum Schluss evaluiert, um seine Leistung zu verstehen und sein Potenzial in Zusammenhang mit bisher angewandten Tagging-Methoden situieren zu können.

In diesem Zusammenhang wurde die Programmiersprache Python in der Jupyter Notebook-Umgebung mit verschiedenen Bibliotheken verwendet, sowie Datensätze der Universal Dependencies für das Trainieren und Evaluieren des implementierten Taggers.

2.1 Literaturrecherche

Mit der Literaturrecherche wurde der aktuelle Forschungsstand aufgearbeitet und die relevante Literatur systematisch zusammengetragen. Folgende Bereiche sollte die Literaturrecherche abdecken:

- State-of the-Art zu den in den letzten fünf Jahren verwendeten POS-Tagging-Methoden
- Für POS-Tagging relevante Deep Learning-Methoden
- Methoden zur effektiven Feature-Repräsentation

Die Literaturrecherche zu Beginn ermöglichte es, eine konkrete Vorstellung zu bereits existierenden POS-Tagging-Methoden und deren Vor- und Nachteile zu erhalten. Somit konnte die Verwendung der Transformers-Technologie für die Entwicklung des eigenen POS-Taggers differenzierter begründet werden sowie das konkrete Modell gewählt werden.

2.2 Entwicklung und Implementierung eines eigenen POS-Taggers

Der POS-Tagger wurde basierend auf der Transformers-Technologie entwickelt und implementiert. Bisherige auf Transformers basierende POS-Tagger nutzten häufig das BERT-Modell von Devlin et al. (2019) (Kondratyuk & Straka, 2019; Xue & Zhang, 2021; Zhang et al., 2020). DistilBERT baut auf BERT auf, verwendet bei gleicher Leistung jedoch weniger Parameter und benötigt weniger Rechenressourcen (Sanh et al., 2020). Aus diesem Grund wurde der POS-Tagger in dieser Arbeit basierend auf dem vortrainierten DistilBERT-Modell implementiert. Genauer fiel die Wahl auf das Modell «DistilBERT base multilingual model (cased)», da der POS-Tagger in drei Sprachen evaluiert wurde («multilingual») und im Deutschen die Gross-/Kleinschreibung eine wichtige Bedeutung bei der Zuteilung der Wortart hat («cased»).

Mit der Programmiersprache Python wurde der entwickelte POS-Tagger anschliessend implementiert. Dabei wurde die Transformers-Library verwendet sowie auf weitere, frei verfügbare Python-Bibliotheken zurückgegriffen. Um das vortrainierte Distil-BERT-Modell für die POS-Tagging-Aufgabe feinzustimmen und die Evaluierung vorzunehmen, wurden die annotierten Datensätze der Universal Dependencies genutzt, die eine einheitliche grammatikalische Annotation in einer Vielzahl von Sprachen anbieten (*Universal Dependencies*, 2021q). Die Datensätze der Universal Dependencies sind bereits in Trainings-, Validierungs- und Testdaten aufgeteilt und wurden in dieser Aufteilung unverändert übernommen.

Für die Implementierung des eigenen POS-Taggers wurde das Kapitel 7 des Hugging-Face-Kurses (Hugging Face, o. J.-d) als Grundlage genommen und auf die Aufgabe des POS-Taggings sowie die vorhandenen Daten angepasst.

2.3 Evaluierung des POS-Taggers

Um die Leistung des entwickelten POS-Taggers messen zu können, wurde dieser in drei Sprachen evaluiert und mit drei existierenden POS-Taggern verglichen. Aufgrund der Sprachkenntnisse der Autorin wurden die Sprachen Deutsch, Französisch und Englisch gewählt.

Für die Evaluierung wurde hauptsächlich der Accuracy-Wert verwendet. Dieser wurde über alle Wörter und über ganze Sätze der Testdatensätze ermittelt. Die trainierten Modelle berechnete zudem für jede Epoche die Precision, den Recall sowie den F1-Wert. Diese Metriken waren auch im Klassifikationsreport für die einzelnen im Datensatz vorhandenen POS-Klassen ersichtlich. Zudem wurde eine Konfusionsmatrix in absoluten

Zahlen erstellt, um besser zu verstehen, welche Falschzuteilungen der POS-Tagger vornahm.

Der Performance-Vergleich des eigenen Taggers wurde mit drei bestehenden POS-Taggern, die auf verschiedenen Methoden basieren, durchgeführt. Zum Vergleich wurden der Stanford-Tagger, der von NLTK empfohlene Perceptron-Tagger und der SpaCy-Tagger ausgesucht. Diese drei Tagger wurden auf den gleichen Testdaten angewandt, wie der neu entwickelte Tagger.

Anbieter	**Methode**	**URL**
Stanford University	Maximale Entropie mit zyklischem Abhängigkeitsnetz (Toutanova et al., 2003)	https://nlp.stanford.edu/software/tagger.shtml
NLTK	Perceptron-Tagger (basierend auf HMM) (Honnibal, 2013; NLTK Project, 2022a)[1]	https://www.nltk.org/
SpaCy	Deep Learning, basierend auf CNN und dem Aufmerksamkeitsmechanismus (Honnibal & Montani, 2017, zitiert nach Partalidou et al., 2019)	https://spacy.io/usage/linguistic-features#pos-tagging

Tabelle 1: Beschreibung der zum Vergleich verwendeten POS-Tagger (eigene Tabelle)

Im quantitativen Vergleich wurde für die Vergleichstagger ebenfalls die Accuracy auf Wort- und Satzebene berechnet sowie eine prozentuale Konfusionsmatrix erstellt. Daneben wurde der F1-Wert für den Vergleich der Performance der einzelnen Klassen verwendet.

Zudem wurde eine qualitative Evaluation vorgenommen. Anhand konkreter Beispiele wurde aufgezeigt, wo der neue POS-Tagger bessere und schlechtere Leistungen als die Vergleichstagger erbringt. Neben beispielhaften Sätzen aus dem Testdatensatz wurden eigene Sätze erstellt, die Homographen enthalten, die nicht der gleichen Wortart entsprechen. Bei Homographen handelt es sich um Wörter, die gleich geschrieben werden, aber unterschiedliche Bedeutungen und oftmals unterschiedliche Aussprachen haben (Bibliographisches Institut GmbH, 2022).

[1] NLTK bietet mehrere POS-Tagger an, der Perceptron Tagger ist der empfohlene Tagger (McCoy, 2016).

2.4 Verwendete Infrastruktur und Ressourcen

Pycharm ist eine Entwicklungsumgebung für Python, die von JetBrains angeboten wird (JetBrains, 2022). Ursprünglich war geplant, den POS-Tagger in dieser Umgebung zu entwickeln. Es wurde dann jedoch auf Jupyter Notebooks ausgewichen, eine browserbasierte Applikation, die auf interaktiven Notebooks basiert und für verschiedene Programmiersprachen, unter anderem Python, verwendet werden kann (*Project Jupyter*, 2022). Grund dafür war, dass die Angleichung der Labelsequenzen (siehe Kapitel 4.1.3) in Pycharm über 24 Stunden dauerte, wohingegen dieser Prozess in Jupyter Notebooks nur wenige Sekunden benötigte. Das Trainieren der Modelle wurde in Colab ausgeführt, da dort auf GPU-Leistung zugegriffen werden konnte, die sich für die parallele Verarbeitung, wie es von Transformers vorgesehen ist, besser eignet. So dauerte das Trainieren statt mehreren Stunden nur einige Minuten. Colab wird von Google angeboten und basiert auf Jupyter Notebooks (Google, o. J.). Es ermöglicht den kostenlosen Zugriff auf verschiedene Arten von Rechenleistung (ebd.).

Um die Standard-Python-Funktionen zu ergänzen, wurden folgende Bibliotheken verwendet:

Name	Anbieter und Zweck	Verwendung
Conllu	Stenström (2022); Conllu-Format lesen und in ein verschachteltes Python-Dictionary überführen.	Einlesen von conllu-Datensätzen (Format der UD-Daten-sätze).
Pandas	The pandas development Team (2022); Datenanalyse und -verarbeitung.	Datensätze in leicht lesbarer Tabellenform darstellen für die Entwicklung sowie Erstellung der Konfusionsmatrix.
Datasets	Hugging Face; Datenvorverarbeitung und Laden von öffentlich verfügbaren Datensätzen des Hugging Face Dataset Hub (Lhoest et al., 2021).	Erstellen, Verarbeitung, Speicherung und Einlesen von Hugging Face-Datensätzen.
Transformers	Hugging Face; vortrainierte Modelle für verschiedene Aufgaben auf Texten, Audiosignalen und Bildern (Wolf et al., 2020).	Aufbau und Training des POS-Tagging-Modells.
Sequeval	Nakayama (2018); Evaluierung von Sequenz-Labeling-Aufgaben.	Messen verschiedener Performance-Werte des Modells nach jeder trainierten Epoche.
Numpy	NumPy Developers (2022); wissenschaftliches Rechnen mit Python.	Mathematische Berechnungen, konkret wurde die Durchschnittsfunktion verwendet.
Scikit-learn	Scikit-learn; maschinelles Lernen mit Python (Pedregosa et al., 2011).	Berechnung der Accuracy-Werte und Erstellen des Klassifikations-Reports.
Seaborn	Waskom (2021); statistische Werte in Python grafisch darstellen.	Visualisierung der Konfusionsmatrizen.

Tabelle 2: Verwendete Bibliotheken (eigene Tabelle)

2.4.1 Hugging Face

Hugging Face nennt sich selbst «the AI community building the future» und ist eine Organisation, die frei verfügbare Ressourcen für Machine Learning entwickelt (*Hugging Face*, 2021b).

Neben standardisierten Transformers-Modellen bietet Hugging Face Bibliotheken und Werkzeuge an, um diese Modelle für eigene Aufgaben anzupassen (Tunstall et al., 2022, Kapitel 1). Die folgende Grafik stellt das «Hugging Face Ecosystem» dar:

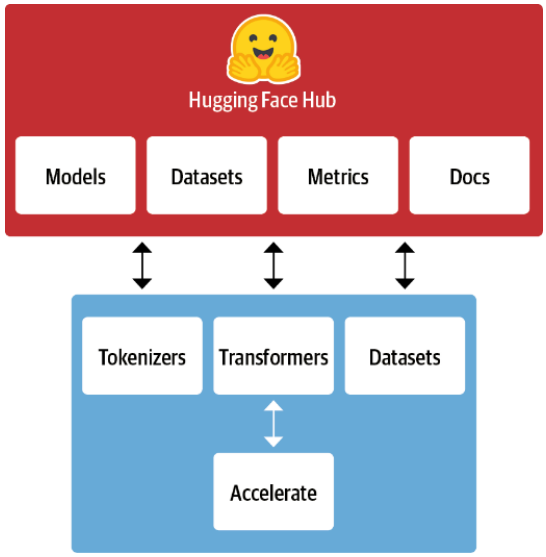

Abb. 2: "Hugging Face ecosystem" (Tunstall et al., 2022, S. 15)

Über 20'000 vortrainierte Transformers-Modelle werden im «Hugging Face Hub» zur Verfügung gestellt. Um diese direkt ausprobieren zu können, werden Datensätze und Code zum Berechnen von Metriken angeboten. Neben dem «Hub» stellt Hugging Face Python-Bibliotheken zur Verfügung, die es ermöglichen, ohne vertiefte Programmierkenntnisse eigene Modelle aufzubauen. Die vortrainierten Modelle sind als «Pipeline» konzipiert. Zu dieser gehört neben dem eigentlichen Transformers-Modell ein Tokenisierer, der die Ein- und Ausgabe normalisiert und in das geforderte Format überführt, bzw. wieder zurück in Wörter und Sätze transformiert. Über eine Schnittstelle kann zudem direkt auf öffentlich verfügbare Datensätze zugegriffen werden, um ein Modell feinzutunen. Dies ermöglicht es, ohne grossen Aufwand eigene Transformers-Modelle für verschiedene Aufgaben zu konzipieren. (Tunstall et al., 2022, Kapitel 1)

2.4.1.1 DistilBERT

2019 wurde die «Bidirectional Encoder Representations from Transformers», kurz BERT von Devlin et al. vorgestellt. BERT ist ein vortrainiertes, bidirektionales Sprachmodell und kann somit im Gegensatz zu anderen Sprachmodellen gleichzeitig den linken und den rechten Kontext in allen Schichten des neuronalen Netzwerks miteinbeziehen. Dies ermöglicht es, BERT nur mit einer zusätzlichen Ausgabeschicht für eine Vielzahl von NLP-Aufgaben feinzutunen. Devlin et al. (2019) haben ihr vortrainiertes Modell für elf Aufgaben ausprobiert und konnten neue State-of-the-Art-Resultate erreichen. (Devlin et al., 2019)

BERT ist ein sogenanntes «Masked Language Model» (Devlin et al., 2019). Während dem Training werden zufällig Tokens des Inputs verborgen und das Modell muss lernen, diese aufgrund des Kontextes vorherzusagen (ebd.). Es eignet sich daher nur für Aufgaben, die den ganzen Satz für Vorhersagen verwenden und nicht beispielsweise für die automatische Textgenerierung (Hugging Face, o. J.-a).

Ein Jahr nach BERT haben Sanh et al. (2020) auf BERT aufbauend eine verdichtete Version von BERT – DistilBERT – entworfen. Die Grösse des originalen BERT-Modells wurde dabei um 40% reduziert (134 Millionen Parameter vs. 177 Millionen). Seine Leistung blieb bei 97% der von BERT erreichten, während gleichzeitig die Geschwindigkeit des Modells um 60% erhöht werden konnte. Dies erlaubt es sogar, das Modell auf mobilen Geräten laufen zu lassen. (Sanh et al., 2020)

Die allgemeine Architektur von DistilBERT entspricht der von BERT, ebenso ist der Trainingskorpus derselbe. Um die Grösse zu reduzieren, wurde mit der sogenannten «Knowledge distillation» (Wissensverdichtung) gearbeitet. Im Training soll das kompakte Modell lernen, das Verhalten des grösseren Modells nachzubilden. (Sanh et al., 2020)

«Distilbert-base-multilingual-cased», das konkrete Modell, das in dieser Arbeit verwendet wurde, ist mit Wikipedia in 104 Sprachen trainiert worden. Es berücksichtigt die Gross-Kleinschreibung (Hugging Face, o. J.-b)

2.4.2 Universal Dependencies

Universal Dependencies ist ein offenes Gemeinschaftsprojekt, das zum Ziel hat, Sprachen einheitlich grammatikalisch zu annotieren. Neben Wortarten werden auch morphologische Eigenschaften und semantische Abhängigkeiten beschrieben. Somit können die Datensätze zur Forschung in verschiedenen NLP-Gebieten und zur Entwicklung von Instrumenten in Zusammenhang mit der natürlichen Sprache verwendet werden. Derzeit sind einheitlich annotierte Datensätze in 130 Sprachen verfügbar, die alle Kontinente abdecken. (*Universal Dependencies*, 2021q)

2.4.2.1 Verwendete Daten

Um das Modell zu trainieren, wurden Daten der Universal Dependencies-Datenbank (https://universaldependencies.org/) verwendet, da diese in verschiedenen Sprachen existieren und manuell annotierte POS-Tags enthalten.

Die deutschen Daten kamen aus dem German GSD-Korpus (McDonald et al., 2013; Universal Dependencies, 2022b), der Texte aus den Nachrichten, aus Bewertungen und aus Wikis enthält (Universal Dependencies, 2022b). Die POS-Tags wurden manuell annotiert

und maschinell in die Universal POS-Tags umgewandelt (ebd.). Nach der Vorverarbeitung enthielt der Trainingsdatensatz 13'579 Sätze, der Validierungsdatensatz 790 und der Testdatensatz 966 Sätze.

Für Englisch wurde der Georgetown University Multilayer Corpus (GUM) verwendet (Zeldes, 2017). Dieser besteht aus vielfältigen Texten, die händisch mit POS-Tags annotiert und automatisiert in die Universal POS-Tags konvertiert wurden (ebd.). Die englischen Daten enthielten nach der Vorverarbeitung 6690 Sätze für das Training, 1096 für die Validierung und 1064 für das Testen.

Der französische Korpus stammte wie der Deutsche aus dem GSD-Korpus (McDonald et al., 2013; Universal Dependencies, 2022a) und enthält ebenfalls manuell annotierte POS-Tags, die maschinell in UPOS-Tags überführt wurden. Da der Validierungsdatensatz viel weniger Daten als der Testdatensatz enthält, wurden diese beiden umgekehrt genutzt. Somit enthielt der Trainingsdatensatz 13'843 Sätze, der Validierungsdatensatz 387 Sätze und der Testdatensatz 1424 Sätze.

Nicht alle POS-Tags waren in den Daten gleich repräsentiert. Seltene Wortarten, wie Interjektionen beispielsweise, waren generell wenig vorhanden. Somit konnte nicht für alle Klassen eine repräsentative Auswertung gemacht werden.

2.4.2.2 Universal POS-Tags

Die Universal POS-Tags entsprechen den Wortarten, die für das Annotieren der Universal Dependencies-Datensätze genutzt werden. In den drei gewählten Datensätzen sind alle POS-Tags vertreten.

Folgende Wortarten werden in den Universal POS-Tags unterschieden:

UPOS-Klasse	Wortart	Definition
ADJ	Adjektiv	Adjektive beschreiben Eigenschaften von Nomen. Ordinalzahlen gelten als Adjektive, Kardinalzahlen hingegen als Zahlwörter. **Beispiel**: gross (in: *grosser* Tiger) (Universal Dependencies, 2021a)
ADV	Adverb	Adverbien beschreiben Verben und verändern Adjektive und andere Adverbien. Die englische Negierung «not» gilt als Partikel. Verbale Präfixe im Deutschen (runterfallen → fiel runter) gelten als Adverbien oder Adpositionen. **Beispiel**: schnell (in: *schnell* rennen) (Universal

		Dependencies, 2021d)
INTJ	Interjektion	Interjektionen drücken eine Emotion aus, meist in Form oder als Teil eines Ausrufs. Wörter, die sowohl als Interjektion, als auch als andere Wortart genutzt werden können, werden nicht als Interjektion getaggt (z.B. *Gott*). **Beispiel**: Psst (Universal Dependencies, 2021i)
NOUN	Nomen	Nomen stehen für Dinge, Personen, Orte, Tiere, Ideen etc. Eigennamen sind nicht in dieser Kategorie enthalten. **Beispiel**: Katze (Universal Dependencies, 2021j)
PROPN	Eigenname	Als Eigennamen werden Nomen bezeichnet, die als Name für ein spezifisches Nomen verwendet werden. Bilden mehrere Wörter einen Eigennamen (z.B. ein Filmtitel), so werden die einzelnen Wörter daraus entsprechend ihren Wortarten getaggt. Im Französischen werden zudem Namen von Personen, die an einem Ort leben (z.B. Franzosen in *die Franzosen*), als Nomen und nicht als Eigennamen getaggt. **Beispiel**: Jacinda Ardern (Universal Dependencies, 2021o)
VERB	Verb	Verben beschreiben Zustände. In dieser Kategorie nicht eingeschlossen sind Hilfsverben und je nach Sprache Modalverben. Dies ist der Fall für zum Beispiel Englisch. **Beispiel**: kochen (Universal Dependencies, 2021s)
ADP	Adposition	Die bekanntesten Vertreter der Adpositionen sind die Präpositionen. Sie beschreiben die grammatikalische oder semantische Zusammengehörigkeit zweier Teilsätze. Auf Deutsch können diese auch als Partikel (Präfix) vorkommen (aufgeben → gab auf) und werden dann trotzdem als Adposition getaggt. Sogenannte «Pseudo-Präpositionen», Verben, die als Präpositionen genutzt werden (z.B. betreffend), werden als Verben getaggt. Bestehen Adpositionen aus mehreren Wörtern (z.B. grâce à), werden sie einzeln entsprechend ihren Wortarten klassiert. **Beispiel**: während (Universal Dependencies, 2021b)

AUX	Hilfsverb	Hilfsverben begleiten ein lexikalisches Verb. In einigen Sprachen gelten auch Modalverben als Hilfsverben. Dies ist im Englischen und Deutschen der Fall, jedoch nicht im Französischen. **Beispiel**: ist (in: *ist* gegangen) (Universal Dependencies, 2021e)
CCONJ	nebenordnende Konjunktion	Bei unterordnenden Konjunktionen handelt es sich um eine Wortgruppe, die Wörter und Teilsätze verbindet, die einander gleichgestellt sind. **Beispiel**: und (Universal Dependencies, 2021g)
DET	Determinativ	Determinative drücken ein Nomen in einem Kontext aus. Artikel werden immer als Determinative klassiert, bei den anderen Untergruppen ist dies sprachabhängig. Auf Französisch und Deutsch gelten zum Beispiel Possessivpronomen als Artikel. **Beispiel**: der (Universal Dependencies, 2021h)
NUM	Zahlwort	Zahlwörter drücken eine Zahl aus. In den U-POS-Tags gelten Zahlwörter auch dann als Kategorie «NUM», wenn sie als Determinativ genutzt werden. Zudem gehören dieser Kategorie sowohl ausge-schriebene Zahlen als auch Ziffern an. Sind die Zahlen mit Punktationen gemischt (z.B. Daten), gelten sie ebenfalls als Zahlwort. Nicht jedoch, wenn sie mit Buchstaben gemischt sind (z.B. 10er), ausser es handelt sich um römische Zahlen. Je nach Syntax kann ein Zahlwort auch als Adjektiv gelten (*erste* in «der *erste* Kuss»). **Beispiel**: 2022 (Universal Dependencies, 2021k)
PART	Partikel	Bei Partikeln handelt es sich um Funktionswörter. Sie können nicht alleinstehend vorkommen. Deutsche Präfixe (aufgeben → gab auf) gelten je nachdem als Adposition oder Adverb. Auf Französisch und Deutsch wird die Wortart kaum genutzt. **Beispiel**: ‘s als Genitiv-Partikel im Englischen (Universal Dependencies, 2021l)
PRON	Pronomen	Als Pronomen gelten Wörter, die anstelle von Nomen stehen können, wenn der Kontext bekannt ist. Possessivpronomen gelten je nach Sprache als Determinativ (z.B. auf Deutsch, Französisch) oder Pronomen (z.B. Englisch). **Beispiel**: alle (Universal Dependencies, 2021n)

SCONJ	unterordnende Konjunktion	Im Gegensatz zu nebenordnenden Konjunktionen, geben unterordnende Konjunktionen den Teilsätzen, die sie verbinden, eine Rangordnung. **Beispiel**: weil (Universal Dependencies, 2021p)
PUNCT	Punktation	Punktationen werden in dieser Masterarbeit in der Vorverarbeitung entfernt und daher nicht verwendet.
SYM	Symbol	Symbole werden in dieser Masterarbeit in der Vorverarbeitung entfernt und daher nicht verwendet.
X	andere	Kann ein Wort keiner Klasse zugewiesen werden, wird es mit «X» getaggt. Dies gilt zum Beispiel für fremdsprachige Wörter, die in der Zielsprache nicht als Fremdwort geläufig sind oder falsch geschriebene Wörter, die nicht erkannt werden können, sowie Fantasiewörter. (Universal Dependencies, 2021t)

Tabelle 3: Universal POS-Tags und ihre Definition (eigene Tabelle)

2.4.3 Verwendete Metriken

Um die Tagging-Ergebnisse zu evaluieren und untereinander zu vergleichen, wurde hauptsächlich mit der Accuracy (Genauigkeit) gearbeitet. Diese bildet das Verhältnis der korrekt gelabelten Daten im Verhältnis zu allen vorhandenen Daten ab (Patterson & Gibson, 2017, Kapitel 1). Die Error Rate ist das Gegenteil, also das Verhältnis der fehlerhaften Klassifikationen zu allen vorhandenen Daten (ebd.).

Die Precision (Präzision) zeigt an, wie viel korrekte Zuteilungen das Modell für eine bestimmte Klasse, im Verhältnis zu allen Zuteilungen zu dieser Klasse, vorgenommen hat (Memari, 2021). Der Recall (Trefferquote), auch «Sensitivity» genannt, misst das Verhältnis zwischen der für eine Klasse korrekt zugeordneten Datenpunkte und der totalen Anzahl Datenpunkte, die dieser Klasse hätten zugeordnet werden müssen (Memari, 2021; Patterson & Gibson, 2017, Kapitel 1).

Der F1-Wert ist der harmonische Mittelwert zwischen Precision und Recall (Memari, 2021). Precision und Recall werden im Zähler multipliziert, im Nenner addiert und anschliessend verdoppelt (ebd.).

Der Durchschnitt wurde für die Anzahl Fehler in Sätzen mit mindestens einem Fehler berechnet. Es handelt sich dabei um einen Mittelwert, bei dem alle Fehler zusammengezählt und dann durch die Anzahl falscher Sätze geteilt wurden.

3 State-of-the-Art

«Part-of-Speech-Tagging» (POS-Tagging) bezeichnet den Prozess, jedem Wort die korrekte Wortart automatisch zuzuordnen (Jurafsky & Martin, 2022). Es handelt sich dabei um eine Sequenzlabeling-Aufgabe, da die Länge des Inputs der Länge des Outputs entspricht (ebd.).

Dieses Kapitel soll den aktuellen Forschungsstand zu Part-of-Speech-Tagging-Methoden aufzeigen sowie geeignete Deep Learning-Anwendungen näher erläutern. Zudem sollen aktuelle Methoden zur Feature-Repräsentation vorgestellt werden.

3.1 POS-Tagging-Methoden ohne Einsatz von Deep-Learning

Im Folgenden wird ein Überblick über die wichtigsten Part-of-Speech-Tagging-Methoden gegeben, die in den letzten fünf Jahren erprobt wurden. Es werden dabei nur Forschungsergebnisse berücksichtigt, die sich auf Part-of-Speech-Tagging für Deutsch, Französisch oder Englisch beschränken, oder multilingual sind.

3.1.1 Lexikon- und regelbasierte Ansätze

Die einfachste Art und Weise, wie ein Part-of-Speech-Tagger programmiert werden kann, ist, dass ihm entweder ein Wörterbuch mit jeweils einem Wort und dem zugehörigen Tag angeboten wird oder grammatikalische Regeln programmiert werden. Diese Tagger haben jedoch zwei entscheidende Nachteile: Sie sind sehr aufwändig zu programmieren und Homographen können nicht berücksichtigt werden. Eine der ältesten automatischen Part-of-Speech-Tagging-Techniken ist der Brill-Tagger, der von Eric Brill (1992) entwickelt wurde und regelbasiert arbeitet. In einem ersten Schritt weist er einem Wort den statistisch gesehen wahrscheinlichsten Tag aufgrund eines annotierten Korpus zu. Unbekannten Wörter, die nicht im Korpus gefunden werden können, werden aufgrund einfacher Regeln, wie zum Beispiel Wortendungen, einem Tag zugewiesen. Mit weiteren, kontextuellen Regeln werden die zugewiesenen Tags allenfalls korrigiert, um so die Fehlerrate insbesondere bei Homographen zu verringern. Gegenüber den stochastischen Taggern, die zu dieser Zeit die besten Performance-Werte erreichten, braucht der Brill-Tagger viel weniger gespeicherte Information, da nur kleine Sets von einfachen Regeln gelernt werden müssen. Da die Regeln einfach zu verstehen sind, ist die Entwicklung und Verbesserung des Taggers nicht kompliziert. Zudem kann ein regelbasierter Tagger besser von einem Tag-Set oder Korpus-Genre auf ein anderes übertragen werden. (Brill, 1992)

Heutzutage spielt lexikon- und regelbasiertes Tagging eine untergeordnete Rolle. Bei Zhou et al. (2018), Farrah et al. (2018) und Li et al. (2021) wurden Wörterbücher bzw. Regeln in hybriden Verfahren genutzt und dienten einzig als Vorbereitung für die anschliessend angewandte Methode. Verschiedene Nachteile des lexikon- und regelbasierten Taggings, unter anderem der hohe Programmier- und Rechenaufwand, führten dazu, dass es zunehmend von anderen Methoden verdrängt wurde. Banga und Mehnidratta (2017) haben die Accuracy und die Geschwindigkeit mehrere Tagger aus NTLK miteinander verglichen, unter anderem den enthaltenen Brill-Tagger. In der Geschwindigkeit konnte er mit den anderen Taggern zwar mithalten, doch in der Accuracy waren ihm alle anderen getesteten Tagger (TnT, Cpos, Perceptron, CRF) überlegen (Banga & Mehndiratta, 2017).

Sadredini et al. (2018) haben ein regelbasiertes Modell vorgestellt, das die grammatikalischen Regeln zur Bestimmung der Wortart in reguläre Ausdrücke umwandelt. Durch den Einsatz von Hardwarebeschleunigern sind viel mehr Regeln und eine höhere Komplexität möglich (Sadredini et al., 2018). Das Modell erzielte eine vergleichbare Accuracy zu statistischen Verfahren und Deep Learning Methoden, war jedoch bis zu 253-mal schneller (ebd.).

Plank und Agic (2018) nutzen lexikalische Information als zusätzliches Feature. Diese werden als «lexicon embeddings» in ein Deep Learning-Modell integriert (Plank & Agić, 2018). So kann der POS-Tagger insbesondere für Sprachen verbessert werden, für die es nur kleine oder gar keine annotierten Korpora gibt, mit dem ein Tagging-Algorithmus für die Zielsprache von Grund auf trainiert werde könnte (ebd.).

Eine mit dem Wörterbuch-Ansatz verwandte Methode, ist der Aufbau eines POS-Taggers mit einer Ontologie. Togatorop et al. (2020) haben einen solchen für Indonesisch entwickelt. Zur Erstellung der Ontologie wurden die Wörter in einen Vektor überführt und darauf basierend klassifiziert (Togatorop et al., 2020). Der Ontologie-Tagger kann jedoch Wörter, die nicht in der Ontologie vorhanden sind, nicht klassieren und hat Schwierigkeiten mit Homographen (ebd.). Dies sind ähnliche Nachteile, mit denen auch wörterbuchbasierte Tagger zu kämpfen haben (ebd.).

3.1.2 Hidden Markov Modell (HMM)

Das Hidden Markov Modell (HMM) ist ein Spezialfall der Bayesschen Inferenz, das im POS-Tagging die wahrscheinlichste Tag-Sequenz finden soll, die eine gegebene Wortsequenz generieren kann (Cutting et al., 1992; Rabiner, 1989). Ein HMM hat zwei Arten von Zuständen: Die Beobachtbaren – im Fall von POS-Tagging sind das die Wörter– und

die Verborgenen («hidden») – im Fall von POS-Tagging die Tags (ebd.). Neben der Wahrscheinlichkeit, dass ein bestimmtes Wort einen bestimmten Tag hat, wird die Übergangswahrscheinlichkeit berücksichtigt, d.h. wie hoch die Wahrscheinlichkeit ist, dass ein bestimmter Tag auf einen anderen folgt (ebd.).

In der untenstehenden Grafik ist das Modell vereinfacht dargestellt mit einer Sequenz von zwei Elementen (1 und 2) und zwei möglichen Zuständen pro Element (H und T).

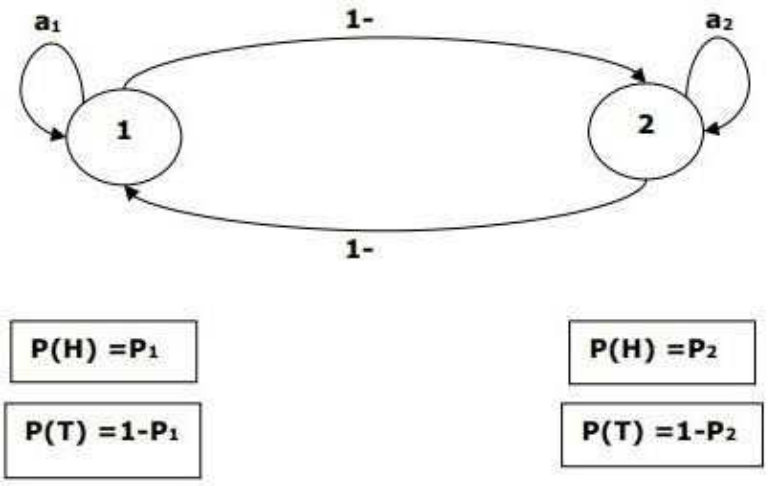

Abb. 3: HMM-Modell (Tutorials Point, 2022)

Die Komplexität eines HMM steigt exponentiell an, je länger die Sequenzen sind und je mehr mögliche Tags es gibt. Deshalb wird der Viterbi-Algorithmus angewandt. Dieser dynamische Algorithmus sorgt dafür, dass nur die wahrscheinlichste Sequenz der verborgenen Zustände – also der Tags – berechnet wird. So wird das Rechenvolumen verringert und die Rechengeschwindigkeit erhöht. (Baishya & Baruah, 2021; Cutting et al., 1992; Liu, 2017)

Im Laufe der Zeit wurde der klassische HMM-Tagger von anderen Taggern überholt (Banga & Mehndiratta, 2017; Khan et al., 2019). Verschiedene Autoren/-innen haben ihn darum weiterentwickelt, zuletzt u.a. Ankita und Abdul Nazeer (2018), Azeraf et al. (2020) sowie Baishya und Baruah (2021). NLTK, das verschiedene Bibliotheken für die natürliche Sprachverarbeitung in Python anbietet, bietet den Perceptron-Tagger und den TnT-Tagger an (Banga & Mehndiratta, 2017; NLTK Project, 2022b). Beide Tagger basieren auf Hidden Markov Models (ebd.).

Der Perceptron-Tagger wurde von Collins (2002) vorgestellt. Bei dieser Art des HMM-Modells hängt die Vorhersage des nächsten Tags in einer Sequenz nur vom aktuellen Tag und nicht von den vorherigen ab (Banga & Mehndiratta, 2017). Der TnT-Tagger («Trigrams'n'Tags») ist ein «Second Order Markov Model». Anstatt wie bei einem

klassischen Markov-Model nur den vorherigen Zustand zu berücksichtigen, werden die zwei vorherigen Zustände berücksichtigt, d.h. Trigramme statt Bigramme (Brants, 2000). zudem werden für unbekannte Wörter zusätzliche Features integriert (ebd.). Beim NLTK-Tagger-Vergleich von Banga und Mendiratta (2017) erreichte der Perceptron-Tagger die besten Werte in Geschwindigkeit und Accuracy, während der TnT-Tagger für das Training und Testen vier Mal mehr Zeit als alle anderen Tagger benötigte. Griebenouw et al. (2019) haben den TnT-Tagger mit anderen Taggern kombiniert und untersucht und konnten so die Error Rate teilweise bis fast 15% verringern.

Bărbulescu und Morariu (2020) haben in ihrer Evaluation von vorwärtsgerichteten, rückwärtsgerichteten und bidirektionalen Bigram- und Trigram-HMMs festgestellt, dass das bidirektionale Trigram-Modell zwar fast State-of-the-Art-Werte erreicht, allerdings viel Zeit für die Dekodierung benötigt. Das rückwärtsgerichtete Trigram-Modell erreicht fast dieselben Resultate mit jedoch viel besseren Geschwindigkeitswerten (ebd.). Das Hidden Markov Modell von Ankita und Nazeer (2018) nutzt 4-gramme und reduziert die Anzahl der Vergleiche, um die Geschwindigkeit zu erhöhen und die Effizienz des Taggers zu verbessern (ebd.). Dafür wird jeder Satz vom Algorithmus in Chunks à je zwei Wörtern aufgeteilt und die die Wahrscheinlichkeit eines Tags für jeden Chunk wird berechnet (ebd.).

Ein klassisches HMM verarbeitet ausschliesslich Wörter als Ganzes. Um auch andere Eigenschaften, wie zum Beispiel Suffixe und Präfixe, verarbeiten zu können, haben Azeraf et al. (2020) eine neue Art der Berechnung vorgestellt. Die Verwendung von vorwärts- und rückwärtsgerichteten Entropie-Wahrscheinlichkeiten führt zu einer Verbesserung der Tagging-Ergebnisse (Azeraf et al., 2020). Laut den Autoren/-innen kann dies als eine Alternative zu rekurrenten neuronalen Netzwerken angesehen werden, insbesondere auch weil sich das Lernen durch weniger Schichten einfacher gestaltet (ebd.).

Um die wahrscheinlichste Sequenz sinnvoll schätzen zu können, benötigt ein HMM sehr viele Trainingsdaten. Zudem haben Hidden Markov Model-Tagger Mühe, im Trainingsdatensatz nicht gesehene Wörter richtig zu klassieren. Da für viele Sprachen nicht genügend Daten vorhanden sind, ist die Tagging-Genauigkeit mit HMMs für diese Sprachen tief. Um dieser Herausforderung Rechnung zu tragen, haben Baishya und Baruah (2021) ein HMM angepasst: In den Trainingsdaten enthaltene Zeichen-Bi- und -Trigramme werden genutzt, um den wahrscheinlichsten Tag für unbekannte Wörter zu ermitteln (Baishya & Baruah, 2021). Die Accuracy für unbekannte Wörter konnte so erhöht werden, was einen Einfluss auf die allgemeine Accuracy hat (ebd.).

Markov-Modelle wurden mit Conditional Random-Fields (CRF) erweitert und ermöglichen so einem Zustand, über eine Zeitspanne bestehen zu bleiben (Sarawagi & Cohen, 2004). Kemos et al. (2020) haben ein solches «Semi-Markov Conditional Random Fields»-Modell mit neuronalen Netzen zu einem «Neural Semi-Markov Conditional Random Fields»-Modell weiterentwickelt. Diese POS-Tagging-Methode basiert auf Zeichen, so dass Texte mit Wörtern, die fälschlicherweise getrennt oder zusammengeschrieben wurden, korrekt erkannt und getaggt werden können (Kemos et al., 2020). Zudem kann der Tagger für Sprachen genutzt werden, die in ihrer Schreibweise keine eindeutigen Worttrennungen benutzen (ebd.).

3.1.3 Maximum Entropy Model (MEM)

Ein anderes Verfahren für die Wahrscheinlichkeitsverteilung, ist die Methode der maximalen Entropie. Ratnaprakhi (1996) hat diese Methode erstmals für POS-Tagging angewandt und konnte damit die Benchmark-Accuracy erhöhen. Das Modell nutzt mehr und diversifiziertere Features als ein HMM, um den korrekten POS-Tag vorherzusagen (Ratnaparkhi, 1996). Neben Wörtern und Tags ein und zwei Positionen vor und nach dem zu taggenden Wort, werden grammatikalische Eigenheiten berücksichtigt, wie zum Beispiel Suffixe und Präfixe oder Grossbuchstaben (ebd.). Griebenouw et al. (2019) hat das MEM-Modell in Kombination mit anderen Tagging-Techniken verwendet und konnte so die Tagging-Ergebnisse noch weiter verbessern.

Condition	Features	
w_i is not rare	$w_i = X$	& $t_i = T$
w_i is rare	X is prefix of w_i, $\|X\| \leq 4$	& $t_i = T$
	X is suffix of w_i, $\|X\| \leq 4$	& $t_i = T$
	w_i contains number	& $t_i = T$
	w_i contains uppercase character	& $t_i = T$
	w_i contains hyphen	& $t_i = T$
$\forall\, w_i$	$t_{i-1} = X$	& $t_i = T$
	$t_{i-2}t_{i-1} = XY$	& $t_i = T$
	$w_{i-1} = X$	& $t_i = T$
	$w_{i-2} = X$	& $t_i = T$
	$w_{i+1} = X$	& $t_i = T$
	$w_{i+2} = X$	& $t_i = T$

Abb. 4: Features für Ratnaparkhis MEM-POS-Tagging-Modell (Ratnaparkhi, 1996, S. 135)

Toutanova und Manning (2000) haben dieses Modell vier Jahre später verfeinert präsentiert und die Grundlage für den Stanford-Tagger geschaffen (The Stanford Natural Language Processing Group, 2020). Zusätzlich zum ursprünglichen Modell haben sie weitere Features hinzugefügt, damit insbesondere auch unbekannte Wörter genauer getaggt

werden können. Mit dem «Cyclic Dependency Network» haben Toutanova et al. (2003) nicht nur weiter verfeinerte Wortfeatures für unbekannte Wörter implementiert, sondern sich auch dem Problem angenommen, dass sich gängige Algorithmen nur in einer Richtung durch die Sequenz durcharbeiten. So kann ausschliesslich der bereits generierte Kontext in den Tagging-Entscheid miteinbezogen werden. Toutanova et al. (2003) haben das Prinzip der Conditional Random Fields von Lafferty et al. (2001) in ihr MEM integriert, das im nachfolgenden Kapitel näher beschrieben wird. Neben Wortfeatures werden die zwei vorherigen und nachfolgenden Tags sowie das vorherige und nachfolgende Wort berücksichtigt (Toutanova et al., 2003). Jedes Wort in der Sequenz wird von diesen zwei Nachbarswörtern und den vier Nachbartags beeinflusst (ebd.). Gleichzeitig wird jedes Wort in Isolation betrachtet, um ein lokales Modell zu trainieren, das die konditionelle Wahrscheinlichkeit eines Tags für dieses Wort maximiert (ebd.). Somit können Accuracy-Werte bis 97.3% erreicht werden (ebd.).

Heid, Wever & Hüllermeier (2021) haben den Stanford-Tagger mit der «Set-valued Prediction» ergänzt, um historische Texte besser taggen zu können. Statt einem einzigen POS-Tag, schlägt der Tagger mit dieser Ergänzung alle möglichen Kandidaten vor (ebd.). Somit können alte, nicht mehr gängige Satzstrukturen und Wörter genauer klassifiziert werden (ebd.). Gleichzeitig wird jedoch eine manuelle Endzuteilung des korrekten Tags notwendig.

3.1.4 Conditional Random Fields (CRF)

Conditional Random Fields wurden im vorherigen Kapitel bereits angesprochen, da sie von einigen Autoren/-innen mit HMMs kombiniert wurden. Tagger, die rein auf Conditional Random Fields aufbauen, existieren ebenfalls und bieten einige Vorteile gegenüber HMMs.

Ein CRF-Modell ist ein ungerichtetes grafisches Modell, das eine loglineare Verteilung definiert und auf konditionellen Wahrscheinlichkeiten basiert (Lafferty et al., 2001; Wallach, 2004). Komplexe Abhängigkeiten in Sequenzen über einen längeren Zeitraum können besser berücksichtigt werden (ebd.). Denn die Wahrscheinlichkeit, dass ein Tag zu einem anderen führt, hängt nur vom aktuellen Wort ab, sondern auch von früheren oder späteren Wörtern im Satz (ebd.). Ein CRF-Modell kann an jeder Stelle auf die ganze Eingabesequenz zurückgreifen und zudem weitere Eigenschaften als die Wörter allein berücksichtigen (ebd.). Anstatt die Sätze aufwändig zu modellieren, wird die Labelsequenz ausgesucht, die die konditionelle Wahrscheinlichkeit des Satzes maximiert (ebd.).

Durch ihre Eignung für die Segmentierung von Sequenzen, sind CRF-Modelle mächtige POS-Tagger. Bei Banga und Mehndiratta (2017) unterlag der CRF-Tagger von NLTK nur knapp dem Perceptron-Tagger, bei Khan et al. (2019) ergab der CRF-Tagger für Urdu[2] sogar bessere Ergebnisse als Deep Learning-Methoden. Fanoon & Uwantika (2019) haben einen CRF-POS-Tagger für Twitter entwickelt, erreichten jedoch keine State-of-the-Art-Ergebnisse. Gemäss den Autoren/-innen könnte dies am kleinen Trainingsdatensatz gelegen haben (Fanoon & Uwanthika, 2019).

Conditional Random Fields kommen in neueren Verfahren in Kombination mit Deep Learning zum Einsatz. Dort werden sie zum Beispiel für das Dekodieren der Tag-Sequenz genutzt, nachdem eines oder mehrere neuronale Netze durchlaufen wurden (Chaudhary et al., 2021; Wang et al., 2020).

3.1.5 Support Vector Machines (SVM)

Support Vector Machine-Modelle versuchen in einem mehrdimensionalen Vektorraum Trennflächen zu finden, die einen Raum von Datenpunkten in zwei oder mehr Klassen trennt. Jede Eigenschaft eines Datenpunkts repräsentiert eine Dimension. Die im Vektorraum repräsentierten Daten werden so oft in eine höhere Hyperebene transformiert, bis eine klare Trennung möglich ist. Der Bereich zwischen den Trennflächen und den Datenpunkten, soll dabei maximal gross sein. (Awad & Khanna, 2015)

Gimenez und Màrquez (2004) haben einen POS-Tagger vorgestellt, der auf Support Vector Machines basiert und eine Accuracy von 97.05% erreicht. Er gehört bis heute zu den State-of-the-Art-Taggern, neuere Tagger können nur einen marginal besseren Wert vorweisen («POS Tagging (State of the art)», 2019). Trotzdem wird das Verfahren in der neueren Forschung nicht mehr gross beachtet; es wird höchstens als Vergleich herangezogen, wie bei Khan et al. (2019), wo die CRF-Methode jedoch besser abschnitt. Griebenouw et al. (2019) haben SVM-Modelle in Kombination mit anderen Methoden verwendet, um so die Fehlerrate zu verringern.

3.1.6 Metaheuristische Verfahren

Metaheuristische Algorithmen suchen mittels statistischer Information und Transformationsregeln die beste Tagsequenz für eine Wortsequenz (Solano Jiménez et al., 2020). Im

[2] Eine Sprache, die in Indien und Pakistan gesprochen wird, vgl. https://www.britannica.com/topic/Urdu-language

Gegensatz zu statistischen Methoden sind sie einfacher, effizienter und robuster (Sierra Martínez et al., 2017). Memetische Algorithmen gelten als eine der effektivsten und flexibelsten Vertreter von metaheuristischen Algorithmen (Cotta et al., 2017). Sie verwenden alle verfügbaren Quellen, um ein Optimierungsproblem zu lösen und wenden verschiedene metaheuristische Verfahren gleichzeitig an (ebd.).

Solano-Jiménez et al. (2020) haben vier metaheuristische Algorithmen untersucht und deren Parameter verbessert. Für Englisch und Spanisch erzielte der durch die Autoren/-innen angepasste memetische «Global-Best Harmony Search»-Algorithmus «GBHS4» mit einer Precision von 97.54% die besten Werte (Solano Jiménez et al., 2020) und übertrifft somit die Ergebnisse des originalen Global-Best Harmony-Taggers von Sierra Martínez et al. (2017).

3.1.7 Entscheidungsbäume

Ein auf Entscheidungsbäumen basierender POS-Tagger wurde 1994 von Helmut Schmid (1994) vorgestellt. Im Gegensatz zu den Hidden Markov-Modellen, müssen weniger Parameter berechnet werden, um die Übergangswahrscheinlichkeit zu bestimmen. Schmids TreeTagger nutzt ausschliesslich diejenigen Parameter, die relevant sind, um das Wort dem einen oder anderen Ast zuzuweisen (ebd.). Der Kontext muss sich dabei nicht auf Unigramme, Bigramme und Trigramme beschränken. Viel mehr können verschiedene Arten von Kontext gleichzeitig miteinbezogen werden – je nachdem was für die Unterscheidung relevant ist (ebd.). Zudem können auch niedrige Wahrscheinlichkeiten akkurat mit kleinen Trainingsdatensätzen berechnet werden (ebd.).

Heid et al. (2021) haben den TreeTagger von Helmut Schmid (1994) mit einer «Set-valued Prediction» ergänzt, so dass er statt einem einzigen POS-Tag verschiedene mögliche Tags vorschlägt. Ihr Ziel dabei war, historische Texte besser taggen zu können, da diese viele veraltete und somit in den Trainingsdatensätzen nicht vorhandene Ausdrücke enthalten (ebd.). Ihre Ergebnisse konnten durch den Einbezug der «Set-valued Prediction» verbessert werden (ebd.). Eine Auswahl an Tags statt einem einzigen könnte auch bei Sprachen, für die POS-Tagger nur eine niedrige Genauigkeit erreichen, von Nutzen sein.

3.1.8 Error-Correcting Output Codes

Die meisten POS-Tagging-Verfahren benötigen einen grossen annotierten Korpus, um gute Ergebnisse zu erzielen. Nicht so die Lösung von Zhou et al. (2018), bei der es sich um ein halbüberwachtes Lernverfahren handelt. Der Ansatz des «Error Correction Code

ECOC» erzielte sehr hohe Werte für Englisch, Italienisch und Madagassisch, ohne dass dabei manuelle Disambiguierung nötig wäre. Features für das Trainieren und Testen werden durch neuronale Sprachmodellierung generiert, wodurch auch das manuelle Feature Engineering wegfällt. (Zhou et al., 2018)

3.1.9 Kombinierte Tagger

Um die Performance von Taggern zu verbessern und die Fehlerrate zu vermindern, werden nicht nur bestehende Tagger verbessert und neue Techniken angewandt, sondern auch bestehende Verfahren miteinander kombiniert. Dies ist insbesondere bei Deep-Learning-Verfahren der Fall (u.a. Chaudhary et al., 2021; Farrah et al., 2018; Wang et al., 2020), kommt jedoch auch bei Nicht-Deep-Learning-Verfahren zur Anwendung. Z.B. bei Bölücü und Can (2021), die ein Cascade Framework aus zwei Modellen gebaut haben oder Griebenouw et al. (2019), die über eine gewichtete Abstimmung die beste Tagger-Kombination ermittelt haben.

Bölücü & Can (2021) haben eine Kombination zweier unüberwachter Modelle, ein sogenanntes «cascade framework», gebaut. Zuerst wird ein Bayessches Modell trainiert. Mit den so vorhergesagten POS-Tags werden anschliessend die Parameter des loglinearen Modells initialisiert. Durch die Kombination dieser zwei unüberwachten Modelle haben sich die Resultate für türkisches und englisches POS-Tagging um bis zu 30% verbessert, als wenn sie einzeln verwendet worden wären. (Bölücü & Can, 2021)

Um dem Problem von ressourcenarmen Sprachen zu begegnen, haben Griebenouw et al. (2019) ebenfalls eine Kombination verschiedener Techniken vorgeschlagen und diese auf südafrikanische Sprachen angewandt. Mit ihrem Ansatz sollen verfügbare Ressourcen optimiert werden, um so die Genauigkeit des Taggers zu verbessern. Es wurden dabei verschiedene Kombinationen bestehender Klassifikationsmodelle (Memory-based tagger, Support Vector Machine, MXPOST, Trigrams'n'Tags) in einer gewichteten Abstimmung untersucht. Verschiedene Messwerte zu Precision und Recall wurden verwendet, um das Gewicht zu bestimmen, das ein Tagger dabei in einer Kombination erhält. Die Fehlerrate wurde durch die Kombination der Tagger nicht immer vermindert, in einigen Fällen ermöglichte dies aber eine Verbesserung der Genauigkeit. (Griebenouw et al., 2019)

3.2 Deep Learning-Verfahren für POS-Tagging

Deep Learning ist eine Machine Learning-Methode, die auf einer Hierarchie von Konzepten basiert. Der Computer setzt komplexe Konzepte aus einfacheren zusammen und

erhält so eine grosse Flexibilität und Leistungsfähigkeit für verschiedenste Aufgaben. Diese Konzepthierarchie geht Schicht für Schicht tief hinunter. Daher kommt der Begriff «Deep Learning» – tiefes Lernen. Insbesondere für Menschen intuitive, aber für den Computer schwierige Aufgaben, können so gelöst werden, zum Beispiel Handschriftenerkennung oder das Erkennen eines bestimmten Wortes in einer gesprochenen Sprachsequenz. (Goodfellow et al., 2018, Kapitel 1)

Machine Learning wiederum ist ein Gebiet der künstlichen Intelligenz, das Computern ermöglicht, aus Erfahrung zu lernen. Das Wissen muss so nicht explizit programmiert werden. Die künstliche Intelligenz ist ein Prinzip, das versucht, Computer in eine Denkrichtung zu bewegen, damit sie «intelligente» Schlussfolgerungen ziehen können. (Goodfellow et al., 2018, Kapitel 1)

Nachfolgend sind diese drei Begriffe und ihre Beziehung zueinander illustriert:

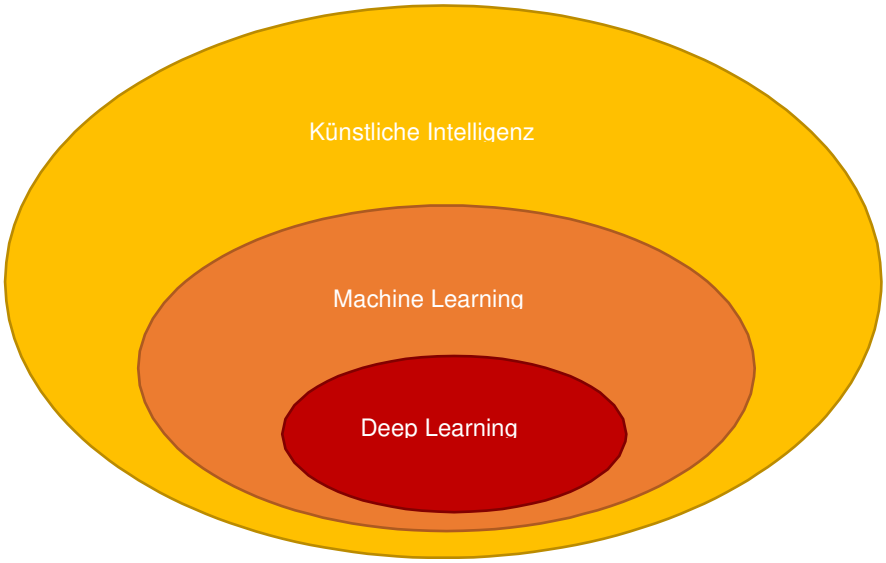

Abb. 5: Beziehung KI, ML, DL (eigene Grafik in Anlehnung an Goodfellow et al., 2018, S. 10)

Die POS-Tagging-Forschung in den letzten fünf Jahren zielte zu einem grossen Teil auf Deep-Learning-Verfahren oder Kombinationen mit dieser ab. Das folgende Kapitel soll einen Überblick über Deep-Learning-Verfahren geben, die in den letzten fünf Jahren (2017-2022) für POS-Tagging verwendet wurden.

3.2.1 Neuronale Feedforward-Netze

Deep Learning basiert auf künstlichen neuronalen Netzen, deren Inspiration die Funktionsweise des menschlichen Gehirns ist (Jurafsky & Martin, 2022, Kapitel 7). Das einfachste Deep-Learning-Modell ist ein neuronales Feedforward-Netz. Es besteht aus einer Eingabeschicht mit einer Anzahl Eingabeneuronen, je nach Komplexität des Problems

einer oder mehrerer verdeckter Schichten und zum Schluss einer Ausgabeschicht. Diese gibt eine Wahrscheinlichkeitsverteilung über die Ausgabeneuronen (im Falle einer Klassifikation) oder eine Zahl auf ein Ausgabeneuron (im Falle einer Regression) aus. Die Information wird über die Neuronen der verschiedenen Schichten von der Eingabe- zur Ausgabeschicht weitergegeben («forward»:vorwärts). (Goodfellow et al., 2018, Kapitel 6; Patterson & Gibson, 2017, Kapitel 2)

Ein Deep-Learning-Modell versucht, die Parameter in den verdeckten Schichten, d.h. die Gewichte der einzelnen Neuronen, so zu optimieren, dass die Kostenfunktion minimiert wird. Die Neuronen der einzelnen Schichten feuern ein Aktivierungssignal an die jeweils nächste Schicht ab. Dieses Aktivierungssignal ist das Ergebnis der Aktivierungsfunktion, die den Wert aller Neuronen einer Schicht, deren Gewichte sowie den Bias kombiniert. Beim Feedforward-Netz gibt es keine Rück- oder Zwischenverbindungen zwischen den Neuronen und Schichten. (Goodfellow et al., 2018, Kapitel 6; Patterson & Gibson, 2017, Kapitel 2)

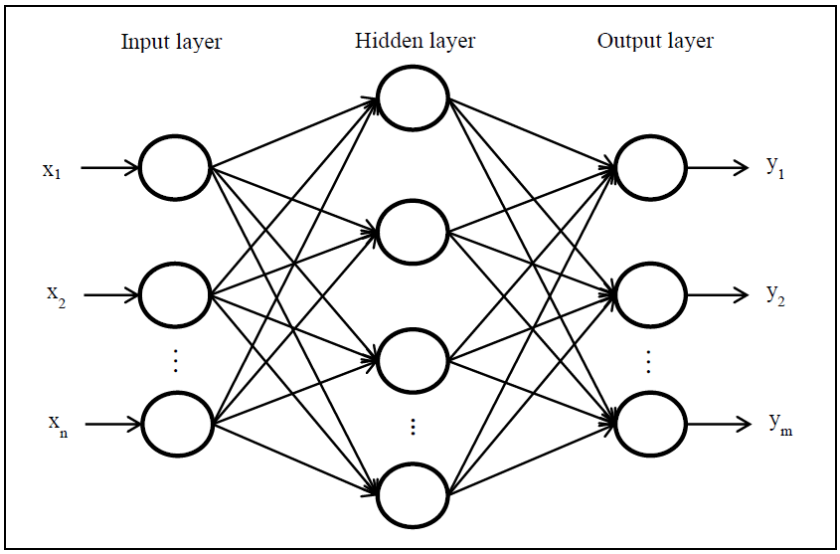

Abb. 6: Architektur eines neuronalen Feedforward-Netzes (Ahmadian & Khanteymoori, 2015)

Die Optimierung eines neuronalen Netzes basiert auf dem Gradientenabstiegsverfahren. Nachdem das Modell mit zufällig gewählten Gewichten initialisiert wurde, soll der Gradient schrittweise in die steilste Richtung absteigen, in der die Kostenfunktion am stärksten minimiert werden kann, um so das globale Minimum zu finden. Beim Backpropagation-Algorithmus wird für die Gradientenberechnung bei jedem Durchlauf das Ergebnis des Modells mit dem erwarteten Ergebnis abgeglichen, um die Gradienten neu zu berechnen. (Goodfellow et al., 2018, Kapitel 6)

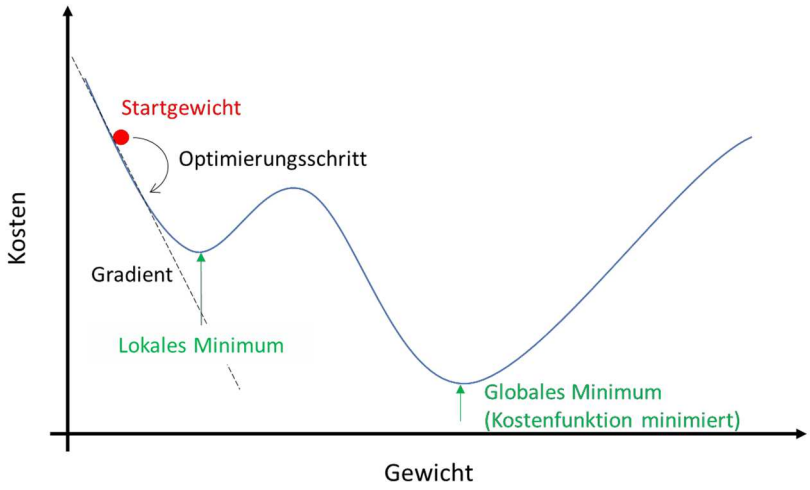

Abb. 7: Gradientenabstiegsverfahren (eigene Abbildung in Anlehnung an A. Kumar, 2020)

3.2.1.1 Nutzung von neuronalen Feedforward-Netzen im POS-Tagging

Im POS-Tagging sind reine Feedforward-Netze weniger geeignet, da keine Abhängigkeiten innerhalb einer Sequenz dargestellt werden können und kein Kontext miteinbezogen werden kann. Bei Kolesau et al. (2018) wurde POS-Tagging jedoch als reine Klassifizierungsaufgabe interpretiert und daher ein neuronales Feedforward-Netz verwendet, das ohne semantische Information auskommt. Anstelle von Word Embeddings wurden Character Embeddings in das Modell eingegeben; d.h. das Modell sollte die Klassifizierung rein aufgrund von Zeichen lernen (Kolesau et al., 2018). Damit sollte erreicht werden, dass Wörter, die nicht im Trainingsdatensatz vorhanden sind, besser erkannt werden können (ebd.).

Farrah et al. (2018) benutzen in ihrem hybriden Ansatz für Arabisch ein neuronales Feedforward-Netz, nachdem die Wörter ein regelbasiertes Modell durchlaufen haben. So sollten mehrdeutige Wörter genauer getaggt werden. Jedoch konnte der für Arabisch beste Accuracy-Wert (96%) nicht erreicht werden (ebd.).

3.2.2 Convolutional Neural Networks (CNN)

Convolutional Neural Networks wurden erstmals von Lecun et al. (1998) vorgestellt und haben mit der Bilderkennung ihren Durchbruch erlangt (Krizhevsky et al., 2012). Gemäss Patterson & Gibson (2017, S. 125) sei sogar «die Effizienz der CNNs bei der

Bilderkennung einer der Hauptgründe, weshalb die Welt die Macht von Deep Learning anerkenne»[3].

Dieser Form von künstlichen neuronalen Netzwerken ist eine Faltungsschicht vorgelagert, die lokale Teilmuster erkennt (Chollet, 2021, Kapitel 8). Der Vorteil liegt darin, dass das Modell das Muster erkennen kann, auch wenn es an einer anderen Stelle im Bild liegt und somit weniger Trainingsdaten benötigt werden (ebd.). Mehrere Convolutional-Schichten können zudem eine Hierarchie von lokalen Teilmustern lernen und so die Bilderkennung sehr effizient gestalten (ebd.).

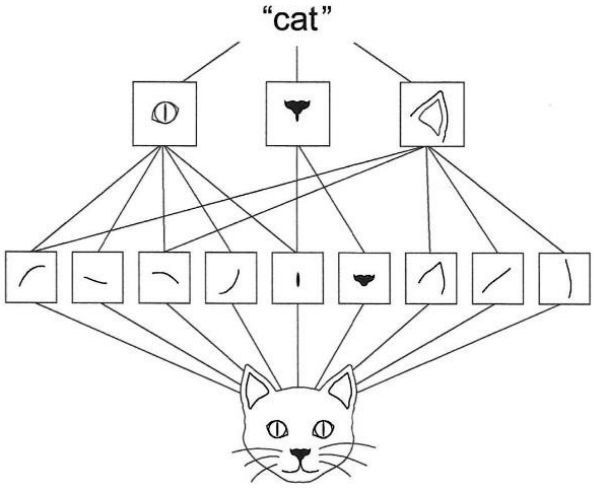

Abb. 8: Beispiel für eine Hierarchie an Teilmustern in einem CNN (Chollet, 2021)

CNNs haben in der maschinellen Bilderkennung viele Erfolge gefeiert. Sie können aber auch in anderen Gebieten, wie der natürlichen Sprachverarbeitung, zum Einsatz kommen. Kalchbrenner et al. (2017) haben Faltungsschichten bei der Verarbeitung von Textsequenzen für die automatische Übersetzung verwendet und konnten so die Performance von rekurrenten neuronalen Netzen für diese Aufgabe übertreffen.

3.2.2.1 Nutzung von CNN im POS-Tagging

Im POS-Tagging kommen Convolutional Neural Networks vor allem für das Feature Engineering zum Einsatz (Kim et al., 2015; Ma & Hovy, 2016; Mundotiya et al., 2021). Dabei werden aus Texten Zeichen- oder Wortfeatures gebildet (ebd.). Diese Features sind allerdings positionsunabhängig, was für eine Sequenz-Aufgabe suboptimal ist (Lee et al., 2018).

[3] Übersetzung der Autorin, Original: «The efficacy of CNNs in image recognition is one of the main reasons why the world recognizes the power of deep learning» (Patterson & Gibson, 2017, S. 125)

3.2.3 Rekurrente neuronale Netze (RNN)

Rekurrente neuronale Netze sind besonders für die Sequenzmodellierung geeignet, da sie – im Gegensatz zu Feedforward-Netzwerken – Schleifen beinhalten, die es erlauben, Sequenzinformation über Zeitschritte zu bearbeiten. Durch diese Schleifen kann das Modell auf die Information des vorherigen Zeitschrittes zurückgreifen. So können komplexe und unterschiedlich lange Sequenzen verarbeitet werden. Dieses Vorgehen wird «Parameter Sharing» genannt und ermöglicht es, eine Gewichtung über mehrere Zeitschritte hinweg beizubehalten. So kann das Modell für Information, die an mehreren Stellen in der Sequenz vorkommt, die gleichen Parameter nutzen. (Goodfellow et al., 2018, Kapitel 10)

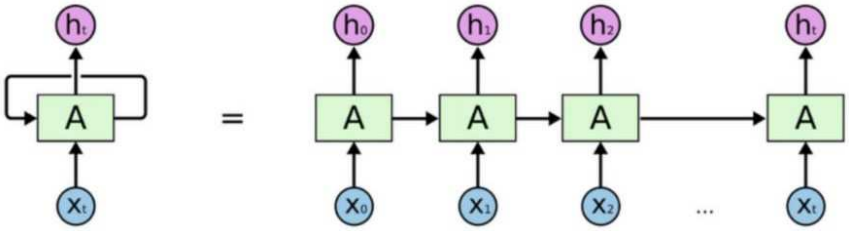

Abb. 9: Aufgefaltete RNN-Schleife (Olah, 2015)

Ein RNN kann also Sequenzen über die Zeit verarbeiten und hat dank der «Rückkoppelung» gewissermassen ein «Gedächtnis», um Information zu speichern (Chollet, 2021, Kapitel 10; Patterson & Gibson, 2017, Kapitel 4). Das Modell kann allerdings nur auf vorherige Information zurückgreifen, solange die Sequenzen kurz und die Abhängigkeiten darin nah beieinander sind. Wird die Sequenz zu lange, ereignet sich das sogenannte «Vanishing Gradient Problem» der «verschwindenden Gradienten» und das Modell kann nicht mehr trainiert werden (ebd.). Um diesem Problem zu begegnen, haben Hochreiter und Schmidhuber (1997) die Long-Short-Term-Memory-Architektur entwickelt.

Eine Sonderform des RNN sind Bidirektionale Recurrent Neural Networks (Bi-RNNs). Chollet (2021, S. 304) nennt sie «das Schweizer Taschenmesser des Deep Learnings für die natürliche Sprachverarbeitung»[4]. Sie erzielen häufig eine bessere Leistung, da die Information in beide Richtungen der Sequenz verarbeitet wird und der POS-Tag daher von Information davor und danach beeinflusst werden kann. (Chollet, 2021, Kapitel 10; Goodfellow et al., 2018, Kapitel 10)

[4] Übersetzung der Autorin, Original: «[…] the Swiss Army Knife of deep learning for natural language processing» (Chollet, 2021, S. 304)

3.2.3.1 Nutzung von RNN im POS-Tagging

Aufgrund des erwähnten *Vanishing Gradient* Problems, eignet sich die einfache Form der rekurrenten neuronalen Netzen nicht für das POS-Tagging. Das zeigen auch die Studien von Premjith et al. (2018) und Gopalakrishnan et al. (2019). Bereits ein Bidirektionales RNN zeigt bessere Ergebnisse, da der Kontext stärker miteinbezogen werden kann, kommt aber nicht an die im folgenden vorgestellten Spezialformen der RNNs heran (Gopalakrishnan et al., 2019; Premjith et al., 2018).

3.2.4 Long-Short-Term-Memory (LSTM)

Das von Hochreiter und Schmidhuber (1997) entwickelte und von Gers (2001) in seiner Doktorarbeit an der ETH Lausanne verbesserte Long-Short-Term-Memory, kann Information über einen Zeitraum von bis zu tausend Schritten speichern, ohne dass sie schwindet, und nicht (mehr) benötigte Information auch wieder vergessen (Chollet, 2021). Ein LSTM ist ein RNN, in dessen Zellen durch Tore reguliert wird, welche Information dem Zellstatus hinzugefügt, weggenommen und weitergegeben wird (Patterson & Gibson, 2017). Die von Gers (2001) eingeführten «Peepholes» ermöglichen es, den Zustand einer Zelle einzusehen. Graves und Schmidhuber (2005) haben das Bi-LSTM vorgestellt, das wie Bi-RNN Sequenzen von beiden Seiten her verarbeitet und somit sowohl zukünftigen als auch vergangenen Kontext miteinbeziehen kann für die Ausgabe.

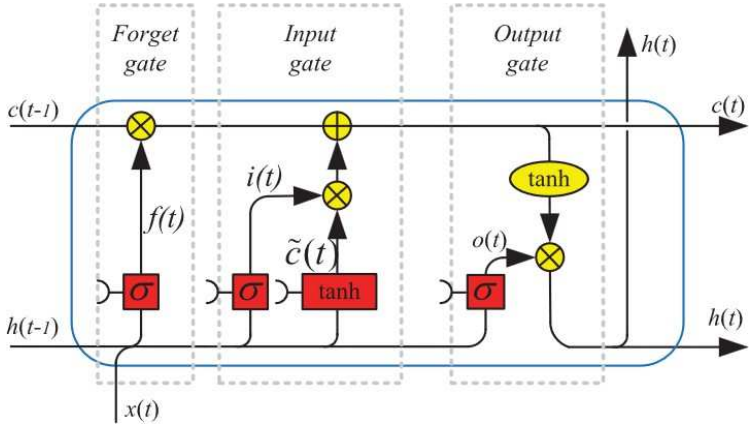

Abb. 10: Architektur einer LSTM-Zelle (Yu et al., 2019, S. 1239)

Dieses kontrollierte Erhalten von Information über einen längeren Zeitraum hat LSTM in vielen Bereichen der natürlichen Sprachverarbeitung sehr erfolgreich gemacht (Goodfellow et al., 2018; Yu et al., 2019). Durch die Zellenstruktur können LSTM zwar weiter auseinanderliegende Abhängigkeiten verarbeiten als RNN, doch ab einer gewissen Distanz kann auch ein LSTM die Information nicht mehr komplett speichern. Wie ein RNN wird es für jeden Zeitschritt erneut aufgerufen, was zu langen Gradientenwegen führt. Zudem

funktioniert das Transferlernen kaum und für jede konkrete Aufgabe wird somit ein spezifisch gelabelter Datensatz benötigt. Nicht zuletzt sind LSTM-Modelle schwierig zu trainieren, da sie viel Rechenleistung benötigen. (Culurciello, 2019; Dirac, 2019; Ye, 2020; Yu et al., 2019)

3.2.4.1 Nutzung von LSTM im POS-Tagging

Dadurch, dass Information über einen längeren Zeitraum und kontrollierter als bei einfachen RNNs gespeichert werden kann, ist das Long-Short-Term-Memory bis heute eine sehr beliebte Methode für POS-Tagging. Sie wird zum Vergleich genutzt (Dhumal Deshmukh & Kiwelekar, 2020; Khan et al., 2019; Kumar et al., 2019), weiterentwickelt (Ali et al., 2021; Anastasyev et al., 2018; Baishya & Baruah, 2021; Plank et al., 2018; Zhang et al., 2018), kombiniert (Chaudhary et al., 2021; Wang et al., 2020; Warjri et al., 2021; Xue & Zhang, 2021) oder für neue Sprachen angewandt (Alrajhi & A ELAffendi, 2019; Dhumal Deshmukh & Kiwelekar, 2020; Otman et al., 2021).

LSTMs werden auch zur gleichzeitigen Lösung von mit POS-Tagging verwandten Aufgaben, wie die Named Entity Recognition (Ali et al., 2021), das Dependency-Parsing (Yang et al., 2018), das Semantic Role Labeling (Shen et al., 2020), oder für die Extrahierung von Lemmas, POS-Tags und morphologischen Eigenschaften zusammen (Arakelyan et al., 2018), verwendet.

Bi-LSTMs wurden u.a. von Srivastava et al. (2018), Plank und Agic (2018) und Anastasyev, Gusev und Indenbom (2018) genutzt, die mit unterschiedlichen Feature Embeddings experimentiert haben. Huang et al. (2015) haben die Bi-LSTM-Architektur weiterentwickelt und mit Conditional Random Fields kombiniert. Die CRF-Schicht ermöglicht den Einbezug von Tag-Information auf Satzebene (Huang et al., 2015). Mit dieser Weiterentwicklung wird das Modell robuster und Sprachsequenzen können mit weniger Abhängigkeit zu Word Embeddings durchgeführt werden (ebd.). Das erlaubt es, mehrdeutige Wörter, oder Wörter mit Rechtschreibefehler besser zu taggen (Warjri et al., 2021). Bi-LSTM-CRF-Modelle gehören zu den Architekturen, die die aktuell höchsten Accuracy-Werte erreichen («POS Tagging (State of the art)», 2019) und wurden in den vergangenen Jahren von vielen Autoren/-innen genutzt (u.a. Akbik et al., 2018; Kumar et al., 2019; Warjri et al., 2021).

Nicht wenige Autoren/-innen haben die LSTM- oder Bi-LSTM-POS-Tagging-Struktur weiterentwickelt. Zhang et al. (2018) haben beispielsweise das Multi-Channel-Modell vorgeschlagen. Dieses berücksichtigt die Token-Tag-Abhängigkeit und die Beziehung der Tags untereinander gleichermassen (Zhang et al., 2018).

Meftah et al. (2019) haben die PretRand-Methode vorgestellt. Diese Joint-Learning Methode nutzt sowohl vortrainierte als auch domänenspezifische Daten, um ein Bi-LSTM feinzutunen. So wird Transferlernen für andere Themengebiete möglich (Meftah et al., 2019). Mit Transferlernen haben sich auch Plank und Agic (2018) beschäftigt. Ihr DsDs-Modell «Distant Supervision from Disparate Sources» ermöglicht es, Bi-LSTM auf Sprachen ohne Gold-Standard-Trainingsdaten anzuwenden (Plank & Agić, 2018). Der mehrsprachige Tagger lernt aus unterschiedlichen Quellen und integriert lexikalische Features (ebd.)

Andere Autoren/-innen, wie Chaudhary et al. (2021) und Wang et al. (2020) nutzen Bi-LSTM für das Feature Embedding und führen das POS-Tagging anschliessend mit einer Kombination von neuronalen Netzwerken und/oder anderen Architekturen durch.

3.2.5 Gated Reccurrent Units (GRU)

Gated Recurrent Units sind eine weitere Unterform der RNNs und eine Spezialform der LSTMs. Sie benötigen weniger Rechenleistung als LSTMs, da das «Vergesstor» und das Eingangstor in ein gemeinsames «Update-Tor» zusammengeführt werden. (Goodfellow et al., 2018, Kapitel 10; Yu et al., 2019, S. 1241)

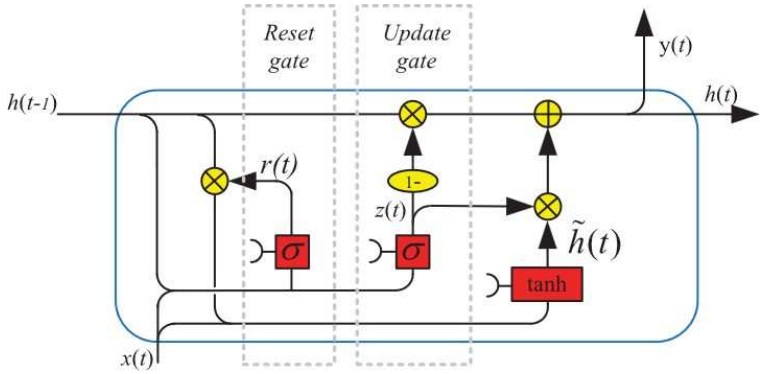

Abb. 11: Architektur einer GRU-Zelle (Yu et al., 2019, S. 1241)

Dadurch, dass die GRU-Zelle nur zwei Tore hat, rechnet das Modell zwar schneller, ist aber auch weniger mächtig als ein LSTM (Goodfellow et al., 2018, Kapitel 10; Yu et al., 2019, S. 1241).

3.2.5.1 Nutzung von GRU im POS-Tagging

Obwohl GRUs weniger mächtig als LSTM-Architekturen sind, konnten Otman et al. (2021) für eine Berbersprache mit dem GRU-Netzwerk sogar minimal bessere Accuracy-Werte erzielen als mit dem LSTM-Netzwerk. Auch Kumar und Sonans (2019) GRU-POS-Tagger zeigte, verglichen mit einem Bi-LSTM, kompetitive Ergebnisse. Muñoz-Valero et

al. (2020) haben in ihrer Arbeit explizit aus dem Grund, dass bei ähnlichen Ergebnissen GRUs weniger Rechenleistung benötigen, eine Architektur mit zwei GRU-Netzwerken aufgebaut. Zuerst wird eine Verbindung zwischen einem individuellen Wort und einem POS-Tag hergestellt, anschliessend werden diese Tags als Inputs genommen und relevanten Satzelementen, wie Subjekt und Prädikat, zugewiesen (Muñoz-Valero et al., 2020). Das Modell kann so die Regeln für Subjekt- und Prädikat-Klassifikation besser lernen, da die Anzahl der POS-Tags kleiner als die Anzahl Wörter im Vokabular ist (ebd.).

Bei Wang et al. (2020) und Mundotiya et al. (2021) wurden GRU-Modelle als Teil einer Kombination von mehreren kombinierten Deep-Learning-Architekturen verwendet.

3.2.6 Transformers

2017 wurde mit dem Artikel «Attention Is All You Need» (Vaswani et al., 2017) von Google-Ingenieuren die Transformers-Architektur erstmals für die Sprachübersetzung vorgestellt. Transformers basiert auf dem Aufmerksamkeitsmechanismus und war ein Meilenstein in der automatischen Verarbeitung natürlicher Sprache. Der bisherige State-of-the-Art, der auf verschiedenen Arten von RNNs basierte, konnte mit der Qualität und den Trainingskosten dieser neuen Architektur nicht mehr mithalten. (Tunstall et al., 2022, Kapitel 1)

Im Gegensatz zu RNNs greifen Transformers auf die gesamte Sequenz zu und bearbeiten die enthaltenen Elemente parallel. Der Aufmerksamkeitsmechanismus kann die Beziehung eines Wortes zu jedem anderen Wort in der Sequenz mit Gewichten modellieren. Somit kann das Modell seine «Aufmerksamkeit» auf die relevanten Wörter der Gesamtsequenz für das zu bearbeitende Wort beschränken. (Rothman, 2021, Kapitel 1; Tunstall et al., 2022, Kapitel 1)

In den folgenden zwei Grafiken kann die Gewichts-, bzw. Aufmerksamkeitsverteilung beobachtet werden. Abb. 10 visualisiert die Aufmerksamkeit in der Sequenz für das Wort «its». In Abb.11 kann das Modell durch die Gewichtsverteilung korrekt «the European Economic Area» in «la zone économique européenne» übersetzen, obwohl die Wortreihenfolge nicht identisch ist.

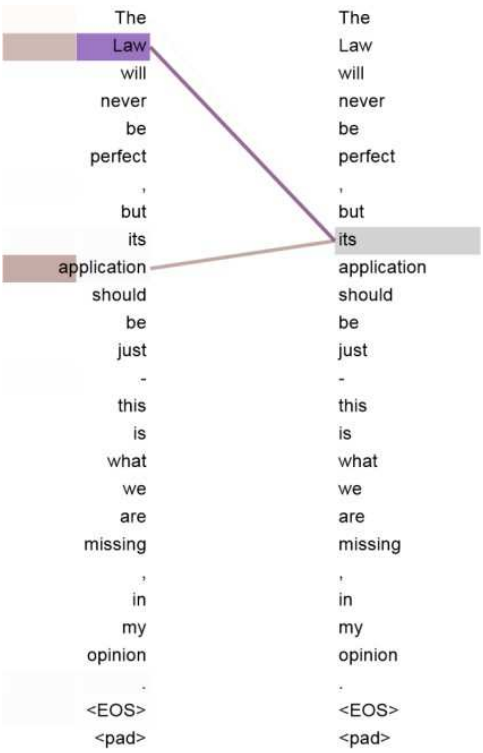

Abb. 12: Aufmerksamkeitsverteilung für das Wort "its" in einer Sequenz (Vaswani et al., 2017, S. 14)

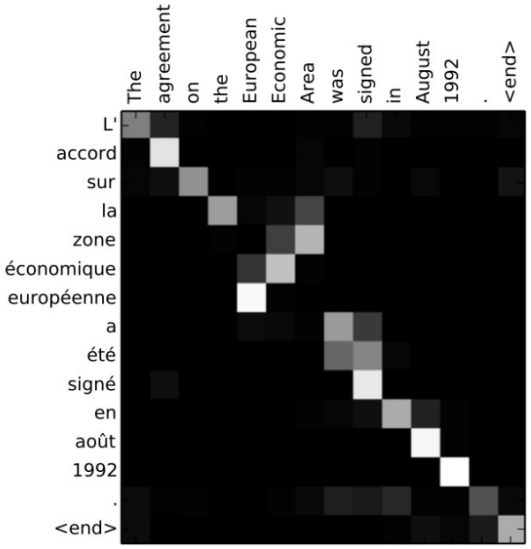

Abb. 13: Gewichtsverteilung bei einer Englisch-Französisch-Übersetzung (Tunstall et al., 2022, S. 5)

Transferlernen ermöglich es, Deep-Learning auch auf Anwendungen mit sehr kleinen Trainingsdatensätzen anzuwenden. Dafür wird ein Modell in einem ersten Schritt mit einem grossen annotierten Trainingsdatensatz vortrainiert. Anschliessend werden mit den Daten der gewünschten Anwendung die Feineinstellungen für dieses vortrainierte Modell vorgenommen. Das Modell lernt die Gewichte für die Basisfeatures und kann damit für die Zielaufgabe initialisiert werden, ohne dass ein grosser gelabelter Datensatz notwendig wäre. Die Transformers-Modelle «GPT» (Radford et al., 2018) und «BERT» (Devlin et al., 2019) waren die ersten, die den Aufmerksamkeitsmechanismus mit dem Transferlernen kombiniert haben und alle bisherigen Ergebnisse übertreffen konnten. (Tunstall et al., 2022, Kapitel 1)

Die Transformers-Architektur besteht aus einem Encoder- und einem Decoder-Block (Carrigan et al., o. J.). Der Encoder-Block optimiert das Modell, damit es den Input verstehen kann (ebd.). Der Decoder nutzt die Features des Encoders gemeinsam mit anderen Features, um eine Sequenz zu generieren (ebd.).

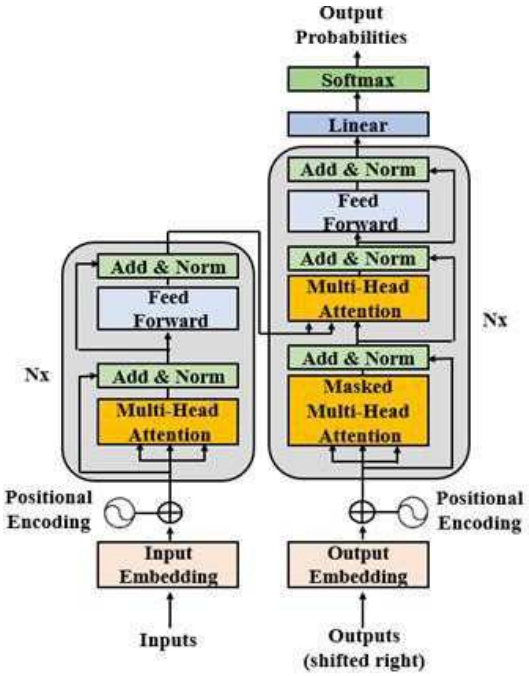

Abb. 14: Transformers-Architektur (Rothman, 2021, S. 5)

Der Encoder- und der Decoder-Block enthalten sechs verschiedene Schichten, die jeweils ein eigenes neuronales Modell darstellen.

- **Inputs:** Die Eingabe der zu bearbeitenden Sequenz
- **Input-Embedding:** Die Sequenz wird tokenisiert und in einen kontextuellen Zahlenraum überführt, damit sie von der Maschine verarbeitet werden kann.
- **Positional Encoding:** Dem Input-Vektor wird mit der Position der einzelnen Wörter mehr Kontext hinzugefügt.
- **Multi-Head Attention:** Hier kommt der Aufmerksamkeitsmechanismus zum Tragen und für jedes Wort wird bestimmt, wie relevant die anderen Wörter der Sequenz für dieses sind.
- **Add & Norm:** Die Verbindung zum Input stellt sicher, dass keine relevante Information verloren geht. Für die Weitergabe an die nächste Schicht wird eine Normalisierung durchgeführt.
- **Feed-Forward:** Ein neuronales Feedforward-Netz wird für jeden Aufmerksamkeitsvektor angewandt, um ihn in ein Format zu überführen, das vom nächsten Block verarbeitet werden kann.

(CodeEmporium, 2020)

Der Decoder-Block funktioniert nach dem gleichen Prinzip wie der Encoder-Block. Im Beispiel einer Übersetzung bearbeitet der Encoder-Block die Originalsprache und der Decoder-Block lernt, die korrekte Sequenz in der Zielsprache vorauszusagen. Für das

POS-Tagging ist nur der Encoder-Block relevant, da es darum geht, die Sequenz zu verstehen. (Carrigan et al., o. J.; Rothman, 2021, Kapitel 1)

3.2.6.1 Nutzung von Transformers im POS-Tagging

Das Transformers-Modell hat nicht nur in der Übersetzung und der Textgenerierung hervorragende Ergebnisse erzielt. Kondratyuk und Straka (2019) haben mit UDify ein Transformers-Modell vorgestellt, das auf alle 75 Sprachen der Universal Dependencies POS-Tags, morphologische Eigenschaften, Lemmas und Dependency-Trees gleichzeitig vorhersagen kann. Als Grundlage wurde das vortrainierte BERT-Modell (Devlin et al., 2019) verwendet, das anschliessend mit den Universal Dependency-Datensätzen feingetuned wurde (Kondratyuk & Straka, 2019). Sogar für Sprachen, auf die weder BERT noch UDify je trainiert worden waren, konnten gute Ergebnisse erzielt werden (ebd.).

Neben Kondratyuk & Straka (2019) nutzen auch Zhang et al. (2020) und Xue & Zhang (2021) BERT als Basismodell. Das Modell von Zhang et al. (2020) wird mit dem Model-Agnostic Meta-Learning Algorithm von Finn et al. (2017) feingetuned. Dieser Algorithmus eignet sich insbesondere für Sprachen mit wenig vorhandenen Daten. POS-Tagging wird als Meta-Lernproblem modelliert und das Modell kann sich aufgrund von datenreichen Sprachen an datenarme Sprachen anpassen. So können mit nur wenigen Trainingsbeispielen bessere Ergebnisse als mit mehrsprachigen Joint-Learning-Ansätzen erzielt werden. (Zhang et al., 2020)

Li et al. (2021) haben ein regelbasiertes Preprocessing angewandt und so für viele Wörter die möglichen POS-Tags auf einen einzigen Tag eingeschränkt. Die POS-Tags der verbleibenden Wörter wurden mit einem Transformers-Modell vorhergesagt (Li et al., 2021). Dadurch kann der bidirektionale Kontext genutzt werden, was zu einer besseren Performance als mit Bi-LSTMs führt (ebd.).

Die originale Transformers-Struktur haben Maksutov et al. (2021) für ihren russischen POS-Tagger genutzt. Um Word Embeddings zu erhalten, nutzten sie das vortrainierte fastText-Modell (Bojanowski et al., 2017), mussten die so erhaltenen Word Embeddings jedoch in die von Transformers vorgesehene Dimensionalität transformieren (Maksutov et al., 2021).

Ali et al. (2021) und Mundotiya et al. (2021) haben den Aufmerksamkeitsmechanismus wie bei Transformers verwendet, jedoch in eine eigene Deep Learning-Architektur integriert.

3.3 Language Models und Feature Engineering

Language Models, auf Deutsch «Sprachmodelle», beschreiben die Wahrscheinlichkeit einer bestimmten Abfolge von Wörtern (Jurafsky & Martin, 2022). In statistischen POS-Taggern kommen meist n-gram-Modelle zur Anwendung (ebd.). Bei n-grams handelt sich um eine Sequenz, die aus n zusammenhängenden Elementen (Wörter, Buchstaben, Silben etc.) aus einem Text besteht (Sarkar, 2019, Kapitel 3). Neuronale Language Models sind jedoch genauer in der Vorhersage, da sie längere Wortfolgen verarbeiten und bei ähnlichen Wörtern besser durch Kontext verallgemeinern können (Jurafsky & Martin, 2022, Kapitel 7).

Features sind die Eigenschaften eines jeden Datenpunkts in einem Datensatz. In einer Tabelle entsprächen sie den einzelnen Spalten jeder Zeile. Den Prozess, diese Features aus den Daten zu extrahieren, nennt man «Feature Engineering» (Sarkar, 2019, Kapitel 4).

Damit ein Algorithmus eine Textsequenz verarbeiten kann, muss diese in einen Zahlenvektor überführt werden. Dafür muss der Text zuerst in einzelne Feature-Elemente unterteilt werden, die «Tokens» genannt werden. (Tunstall et al., 2022)

3.3.1 Tokenisierung

Sarkar (2019, S. 108) umschreibt Tokens als «[…] unabhängige und minimale textuelle Komponenten, die eine gewisse definierte Syntax und Semantik haben»[5]. Die Tokenisierung ist in der Folge der Prozess, der einen Text in Tokens umwandelt. Nach der Definition von Tunstall et al. (2022, S. 29) handelt es sich bei dabei um den «Schritt der Zerlegung einer Zeichenkette in die im Modell verwendeten atomaren Einheiten.»[6]

Tokens können wortbasiert, zeichenbasiert oder teilwortbasiert sein. Daneben gibt es andere Techniken, wofür spezifische Modelle existieren. Die Art der Tokenisierung hängt von der Aufgabe und dem zu verwendenden Modell ab. Die Tokens sollen eine so kleine Einheit wie möglich bilden und gleichzeitig möglichst repräsentativ sein. Für das gleiche Modell muss zwingend immer dieselbe Tokenisierung und dieselbe Umwandlung in den

[5] Übersetzung der Autorin. Original: «[…] independent and minimal textual components that have some definite syntax and semantics» (Sarkar, 2019, S. 108)

[6] Übersetzung der Autorin. Original: «Tokenization is the step of breaking down a string into the atomic units used in the model» (Tunstall et al., 2022, S. 29)

Zahlenraum vorgenommen werden, da ansonsten die Ergebnisse falsch berechnet werden und nicht mehr signifikativ sind. (Carrigan et al., o. J.)

Bei zeichenbasierten Tokens wird der Text in die einzelnen Zeichen, die ihn zusammensetzen, zerlegt. Zeichenbasierte Tokenisierung hat den Vorteil, dass Rechtschreibefehler und seltene Wörter gut verarbeitet werden können. Nachteilig ist jedoch der hohe Rechen- und Speicheraufwand, da das Modell die Wörter erst aus den Daten lernen muss. (Tunstall et al., 2022)

Kumar et al. (2019), Anastasyev et al. (2018), Premjith et al. (2018) und Srivastava et al. (2018) haben mit zeichenbasierten Embeddings gute POS-Tagging-Resultate erzielt. Das Ergebnis von Kolesau et al. (2018) zeigte, dass je nach Sprache die Zeichen allein genügend Information liefern, um Wörtern den korrekten POS-Tags zuzuweisen. Für Sanskrit beispielsweise, ist ein zeichenbasierter Ansatz sogar besser geeignet (Premjith et al., 2018). Der Vorteil einer auf Zeichen basierenden Architektur für POS-Tagging ist, dass das Modell relativ einfach für verschiedene Sprachen angewandt werden kann, da ausschliesslich das Alphabet bekannt sein muss (Kolesau et al., 2018). In Sprachen mit vielen Homographen hingegen, wird zusätzliche Information benötigt (ebd.).

Wortbasierte Tokenisierung teilt den Text in einzelne Wörter und das Modell muss deren linguistische Struktur somit nicht mehr lernen. Da bei der Eingabe in ein Modell jedes unterschiedliche Token eine Dimension repräsentiert, führt dies bei einem grossen Vokabular zu einem enormen Vektorraum für die Eingabe. Zudem können Wörter, die nicht in den Trainingsdaten vorgekommen sind, keiner Dimensionalität zugewiesen werden. (Tunstall et al., 2022)

Teilwort-Tokens bilden einen Kompromiss zwischen den Zeichen- und Wort-Tokens. Häufig vorkommende Wörter werden dabei als Ganzes betrachtet, während seltene Wörter in Teilwörter unterteilt werden (Carrigan et al., o. J.; Tunstall et al., 2022). Um eine auf Teilwörtern basierende Tokenisierung durchzuführen, werden mathematische Algorithmen und statistische Verfahren angewandt (ebd.). Nach diesem Prinzip kann «schlüsselloch» als seltenes Wort betrachtet werden und in «schlüssel» und «loch» aufgeteilt werden. Diese Teilwörter haben beide eine Bedeutung und kommen im Korpus häufiger vor als ihre zusammengesetzte Form.

Chaudhary et al. (2021) haben beispielsweise Teilworttokens für ihre Arbeit genutzt. Akbik et al. (2018) haben mit Zeichensequenzen gearbeitet, um den Kontext, der ein Wort umgibt, in das Modell miteinbeziehen zu können. Während die Ergebnisse für die Named Entity Recognition verbessert werden konnte, bewegen sie sich beim POS-Tagging im State-of-the-Art-Bereich (Akbik et al., 2018). Auch das Modell von Wang et al. (2020)

sollte keine bedeutungslosen Teilsequenzen lernen, sondern vielmehr Zeichenfeatures und Satzfeatures gleichzeitig, um seine Leistung zu verbessern. Baishya & Baruah (2021) haben eine Kombination von Wörtern, Zeichen, Zeichen-Bigrammen und Zeichen-Trigrammen als Embeddings verwendet. Sowohl die gesamte Accuracy als auch die Accuracy für unbekannte Wörter konnte verbessert werden (Baishya & Baruah, 2021).

3.3.2 Embeddings

Machine Learning-Modelle erwarten einen Eingabevektor als Input, dessen Elemente sie gewichten und weiterverarbeiten, um anschliessend ein Ergebnis zu berechnen. Um die Tokens dem Modell übergeben zu können, müssen diese daher in einen Zahlenraum überführt werden. Analog der Tokenisierung, muss zwingend immer dasselbe Feature Embedding für ein trainiertes Modell verwendet werden, um die Kohärenz der Ergebnisse zu gewährleisten. (Carrigan et al., o. J.)

Eine relativ einfache und trotzdem mächtige Technik, um Text-Features in einen Vektorraum zu überführen, ist das «Bag-of-Words-Modell» (Sarkar, 2019, Kapitel 4). Bei dieser Methode wird jedes Dokument in einen Vektor umgewandelt, der die Häufigkeit aller unterschiedlichen Wörter dieses Dokuments repräsentiert (ebd.). Somit widerspiegelt das Gewicht eines Worts dessen Häufigkeit in einem Dokument (ebd.). Ein Dokument muss in diesem Fall nicht zwingend ein «Dokument» im klassischen Sinne sein, sondern definiert eine abgeschlossene Textsequenz, was auch ein Satz in einer Liste von Sätzen sein kann.

Die einfachste Art des Token Embeddings ist es, jedem unterschiedlichen Token eine numerische ID zuzuweisen. Die Anzahl der Tokens bestimmt die Dimensionalität der Vektoren (Tunstall et al., 2022). Mit dieser Dimensionalität werden dann die Eingabevektoren gebildet (ebd.). Dies ist das gleiche Prinzip wie bei «Bag-of-Words», nur dass statt Wort-Token jegliche Art von Token verwendet werden kann. Der Nachteil dieser Methode: Die Position eines Tokens im Satz geht komplett verloren (Sarkar, 2019, Kapitel 4). Die zwei Sätze «Ich liebe Wandern und ich hasse Schwimmen» und «Ich liebe Schwimmen und ich hasse Wandern» hätten die gleiche Bedeutung. Bag of n-grams wirken dem entgegen, indem jeweils eine Sequenz von n aufeinanderfolgenden Wörtern eine Dimension bilden (Sarkar, 2019, Kapitel 4).

Vortrainierte Feature Embedding-Modelle funktionieren mit komplexeren Algorithmen und können den Tokens beim Umwandeln in einen Vektor eine tiefere Bedeutung mitgeben (Sarkar, 2019). Häufig verwendete Word Embedding-Algorithmen sind beispielsweise Word2Vec von Google (Google, 2013; Mikolov et al., 2013), der unter anderem von

Srivastava et al. (2018) verwendet wurde, und FastText von Facebook (Bojanowski et al., 2017; Facebook Inc., 2022), den beispielsweise Ali et al. (2021) und Maksutov et al. (2021) verwendet haben, oder GloVe der Stanford University (Pennington et al., 2014b, 2014a), z.B. verwendet von Muñoz-Valero et al. (2020). Transformers hat für seine vortrainierten Modelle auch vortrainierte Tokenisierer, die direkt eine Umwandlung in den Vektorraum vornehmen (Carrigan et al., o. J.).

4 Entwicklung und Implementierung eines DistilBERT-POS-Taggers

Bei der Vorverarbeitung der Daten und dem Aufbau des Modells wurde das Kapitel 7 des HuggingFace-Kurses (Hugging Face, o. J.-d) als Inspiration und Grundlage genommen. Es wurde ein vortrainiertes DistilBERT-Modell als Basis genommen.

4.1 Daten-Vorverarbeitung

Universal Dependencies bietet seine Korpora bereits in Trainings-, Validierungs- und Testdaten unterteilte Datensätze an, die als conllu-Dateien heruntergeladen werden können. Die Sätze enthalten neben UPOS weitere annotierte Features. Um das Modell mit diesen Daten trainieren zu können, mussten sie in einem ersten Schritt vorverarbeitet werden.

4.1.1 Daten vorbereiten

Aus den Daten der Universal Dependencies wurden als allererstes die Wörter und POS-Tags extrahiert und in jeweils eigene Listen überführt, da nur diese Features relevant für das spätere Modell sein würden. Transformers-Modelle erwarten ein Hugging Face-Datenformat als Eingabe. Aus den extrahierten Listen wurde deshalb ein Dictionary erstellt, das dann in ein Hugging Face-Dataset umgewandelt wurde.

Erst sollten die Satzzeichen im Datensatz verbleiben, da diese einen Einfluss auf vorhergehende oder nachfolgende Wörter haben können. Jedoch führte dies bei der Tokenisierung zu Schwierigkeiten, weshalb sie mittels Regex entfernt wurden. Auch alle entsprechenden POS-Tags wurden entfernt, damit die Länge der Wort- und POS-Sequenzen übereinstimmte. Dabei fiel auf, dass es Fehler im Datensatz gab und mindestens ein Wort fälschlicherweise als Satzzeichen getaggt worden war. Sätze, bei denen die Länge der Wort- und POS-Tag-Sequenz nach dem Entfernen der Satzzeichen und entsprechender Tags nicht mehr übereinstimmte, wurden daher aus dem Datensatz entfernt.

```
['Das', 'alte', 'Posthaus', 'wurde', '1870', 'zum', 'zu', 'dem', 'Pfarrhaus', '', 'einige', 'der', 'zur', 'zu', 'der', 'Post
station', 'gehörenden', 'Grundstücke', 'wurden', '1871', '', '73', 'mit', 'der', 'neogotischen', 'evangelischen', 'Kirche',
'überbaut', '']
['DET', 'ADJ', 'NOUN', 'PUNCT', 'NUM', '_', 'ADP', 'DET', 'NOUN', 'PUNCT', 'PRON', 'DET', '_', 'ADP', 'DET', 'NOUN', 'ADJ',
'NOUN', 'AUX', 'NUM', 'PUNCT', 'NUM', 'ADP', 'DET', 'ADJ', 'ADJ', 'NOUN', 'VERB', 'PUNCT']
```

Abb. 15: Fehlerhaft getaggter Satz; das Wort "wurde" wird fälschlicherweise mit "PUNCT" gelabelt (eigene Grafik)

Somit ergab sich folgende Funktion, die die Wörter und UPOS-Tags der Universal Dependencies-Datensätze aus dem conllu-Datensatz in ein Hugging Face-Datenset überführen kann:

```
def prepare_data(data):
    """
    Öffnet ein .conllu-file und wandelt es in die von Transformers geforderte Form um.
    Satzzeichen werden gelöscht und der entsprechende POS-Tag ebenfalls.
    Elemente, deren Länge zwischen den Wörtern und den Tags nicht übereinstimmt, werden nicht berücksichtigt.
    :param data: .connlu-file der UD
    :return: Transformers-dataset für POS-Tagging mit features "word", "pos", "idx"
    """
    data_file = open(data, "r", encoding="utf-8")
    saetze=[]
    for tokenlist in parse_incr(data_file):
        saetze.append(tokenlist)

    data={'idx':[], 'words': [], 'pos': []}
    idx = 0
    for element in saetze:
            woerter=[]
            postagswoerter=[]
            for e in element:
                woerter.append(e['form'])
                postagswoerter.append(e['upos'])
            #Satzzeichen rausnehmen
            pattern = r"[^a-zA-Z0-9ßëäöüïéèîôûâàùçËÄÖÜÏÈÉÀÙÂÎÔÛ]"
            woerter = [re.sub(pattern, '', wort) for wort in woerter]
            #POS-Tag Punktuation entfernen, da entsprechende Elemente aus den Wortlisten entfernen werden
            while 'PUNCT' in postagswoerter: postagswoerter.remove('PUNCT')
            while '' in woerter: woerter.remove('')
            #überprüfen, ob Anzahl Elemente mit Anz Pos-Tags übereinstimmt und nur dann dem Datensatz hinzufügen
            if len(woerter) == len(postagswoerter):
                data['idx'].append(idx)
                data['words'].append(woerter)
                data['pos'].append(postagswoerter)
                idx+=1

    #Überführen des erstellten Dict in HuggingFace-Datensatz
    dataset = Dataset.from_dict(data)
    return dataset
```

Abb. 16: Funktion, die UD-Daten einliest und für Transformers transformiert (eigene Grafik)

4.1.2 Tokenisieren der Daten

Um die Sätze zu tokenisieren, wurde der zum später verwendeten Modell «distilbert base-multilingual-cased» zugehörige Tokenisierer verwendet. Da die Sätze bereits in Wörter unterteilt waren, musste dies dem Tokenisierer mit dem Befehl «is_split_into_words=True» präzisiert werden. Die Tokenisierungs-Funktion ergänzt den Datensatz mit Input-IDs sowie einer Aufmerksamkeitsschicht. Die Input-IDs repräsentieren dabei Teilwort-Token Embeddings.

```
def tokenize(batch):
    """
    Tokeniziert den Input-Text und überführt ihn in das geforderte Format
    :param batch: Datensatz im Transformers Dataset-format
    :return: Tokenisierter Text
    """
    checkpoint = "distilbert-base-multilingual-cased"
    tokenizer = AutoTokenizer.from_pretrained(checkpoint)
    tokens = tokenizer(batch['words'], truncation=True, is_split_into_words=True)

    return tokens
```

Abb. 17: Funktion zur Tokenisierung der Wörter (eigene Grafik)

Weil Transformers-Tokenisierer mit Teilwort-Tokens arbeiten, wurden durch das Tokenisieren Wörter teilweise in zwei oder mehr Tokens unterteilt. Diese können durch ein Rautezeichen (#) zu Beginn des Tokens erkannt werden. Besteht ein Wort aus mehreren Satzzeichen werden diese hingegen getrennt, ohne sie entsprechend zu kennzeichnen. Doppelte Gedankenstriche, Smileys und ähnliche Satzzeichenkombinationen führten somit zu Schwierigkeiten in der nachfolgenden Angleichung der POS-Sequenz an die Token-Sequenz. Aus diesem Grund wurden alle Satzzeichen, wie im vorherigen Kapitel beschrieben, entfernt.

4.1.3 Angleichen der Labelsequenz an die Input-Sequenz

Da durch das Tokenisieren die Input-Sequenz länger als die POS-Tag-Sequenz wurde, mussten die POS-Tags an die Tokens angepasst werden.

Die Input-IDs wurden hierfür mit der Funktion «tokenizer.convert_ids_to_token» in Tokens konvertiert. Anschliessend wurde für jedes Token überprüft, ob es a) sich um die Bezeichnung für den Anfang ([CLS]), das Ende eines Satzes ([SEP]) oder ein Fülltoken zur Angleichung an die längste Sequenz ([PAD]) handelt. Diesen Token wurde das Label -100 zugewiesen, da dieser Index vom Modell bei der Berechnung der Loss-Funktion ignoriert wird (Hugging Face, o. J.-d). Wenn das Token b) mit einem Rautezeichen beginnt, gehört es zum vorherigen Wort und das zugehörige Label wurde erneut in die Liste aufgenommen.

```
def align_labels_with_tokens(labels, word_ids):
    """
    angepasst von https://huggingface.co/course/chapter7/2?fw=pt
    Ergänzt die Labelsequenz, so dass sie die gleiche Länge wie die word_id-Sequenz aufweist
    :param labels: Liste von labels
    :param word_ids: Liste von word_ids
    :return: angepasste Label-Liste
    """
    #checkpoint = "distilbert-base-multilingual-cased"
    #tokenizer = AutoTokenizer.from_pretrained(checkpoint)
    new_labels = []
    current_word = None
    current_label=-1
    for word_id in word_ids:
        if word_id is not None:
            token = tokenizer.convert_ids_to_tokens(word_id)
            if token == '[CLS]' or token == '[SEP]' or token == '[PAD]':
                #kein Wort
                label = -100
            elif token.startswith('#') is True:
                #gleiches Wort wie vorheriges
                label = labels[current_label]
            else:
                #neues Wort
                current_word = word_id
                current_label+=1
                label = labels[current_label]
            new_labels.append(label)
    return new_labels
```

Abb. 18: Funktion zum Angleichen der Labelsequenz an die Inputsequenz (eigene Grafik)

Damit die Labelsequenzen des ganzen Datensatzes mit der Methode «Dataset.map()» in einem Schritt angeglichen werden können, wurde eine zweite Funktion erstellt. Diese wandelte gleichzeitig die String-Label in Integer-Label um, da dies vom Modell später so erwartet wird. Zudem wurde die Spalte «pos» in «labels» umbenannt, damit sie vom Modell korrekt erkannt wird.

```
def align_all_labels_with_token(element):
    """
    gleicht alle Labelsequenzen eines Datensatzes an und wandelt die Labels in Integers um.
    :param element: Datensatz im Transformers Dataset-format
    :return: Datensatz mit angepassten Labels
    """
    c2l = ClassLabel(num_classes=18, names=['ADJ', 'ADP', 'ADV', 'AUX', 'CCONJ', 'DET', 'INTJ', 'NOUN', 'NUM',
                                            'PART', 'PRON', 'PROPN', 'PUNCT', 'SCONJ', 'SYM', 'VERB', 'X', '_'])

    labels = element['pos']
    element['pos'] = [c2l.str2int(label) for label in labels ]

    label_seq=element['pos']
    word_ids=element['input_ids']
    element['pos'] = align_labels_with_tokens(label_seq, word_ids)
    return element
```

Abb. 19: Funktion zum Angleichen der Labels des ganzen Datensatzes (eigene Grafik)

Nach diesen Vorverarbeitungsschritten sahen die Features der Datensätze folgendermassen aus:

```
{'idx': Value(dtype='int64', id=None),
 'words': Sequence(feature=Value(dtype='string', id=None), length=-1, id=None),
 'labels': ClassLabel(num_classes=18, names=['ADJ', 'ADP', 'ADV', 'AUX', 'CCONJ', 'DET', 'INTJ', 'NOUN', 'NUM', 'PART', 'P
RON', 'PROPN', 'PUNCT', 'SCONJ', 'SYM', 'VERB', 'X', '_'], id=None),
 'input_ids': Sequence(feature=Value(dtype='int32', id=None), length=-1, id=None),
 'attention_mask': Sequence(feature=Value(dtype='int8', id=None), length=-1, id=None)}
```

Abb. 20: Features de vorverarbeiteten UD-Datensätze (eigene Grafik)

Im Anschluss wurden die Spalten «idx» und «words» entfernt, damit ausschliesslich Features vorhanden sind, die vom Modell später erkannt und verarbeitet werden können.

```
{'idx': 0,
 'words': ['Sehr',
  'gute',
  'Beratung',
  'schnelle',
  'Behebung',
  'der',
  'Probleme',
  'so',
  'stelle',
  'ich',
  'mir',
  'Kundenservice',
  'vor'],
 'pos': ['ADV',
  'ADJ',
  'NOUN',
  'ADJ',
  'NOUN',
  'DET',
  'NOUN',
  'ADV',
  'VERB',
  'PRON',
  'PRON',
  'NOUN',
  'ADP']}
```

Abb. 21: Datensatz vor der Vorverarbeitung (eigene Grafik)

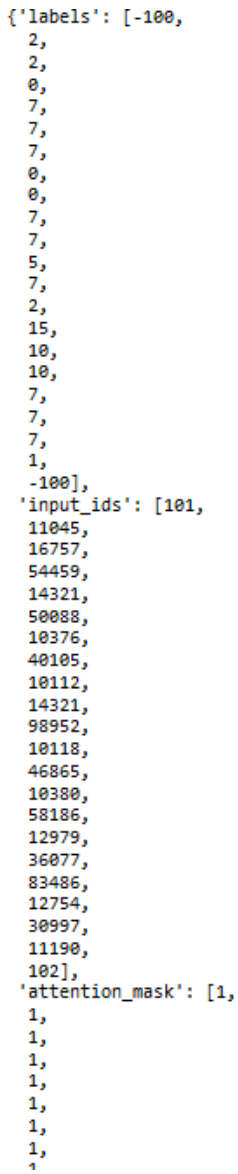

Abb. 22: Datensatz nach der Vorverarbeitung (eigene Grafik)

4.2 Modellierung

Der POS-Tagger wurde basierend auf dem DistilBERT-Modell (Sanh et al., 2020) von Transformers aufgebaut. Bisherige auf Transformers basierende POS-Tagger haben häufig BERT genutzt (Kondratyuk & Straka, 2019; Xue & Zhang, 2021; X. Zhang et al., 2020). DistilBERT baut auf BERT auf. Obwohl DistilBERT weniger Parameter verwendet und schneller ist, erbringt das Modell fast dieselbe Leistung wie BERT (Hugging Face, o. J.-a). Aus diesem Grund schien es sinnvoll, DistilBERT anzuwenden. Genauer wurde das Modell «DistilBERT base multilingual model (cased)» (Hugging Face, o. J.-b) genutzt. Dieses spezifische Modell wurde gewählt, da das multilinguale vortrainierte Modell es erlauben würde, den POS-Tagger in drei Sprachen zu trainieren («multilingual»). Insbesondere im Deutschen hat die Gross-/Kleinschreibung eine wichtige Bedeutung bei der

Zuteilung der Wortart. Aber auch in Englisch und Französisch können so vor allem Eigennamen ausgemacht werden. Aus diesem Grund wurde die «cased»-Version gewählt.

4.2.1 Data Collator

Das Modell verarbeitet die Daten im Datensatz parallel und erwartet dabei stets gleich lange Sequenzen (Hugging Face, o. J.-c). Dies kann beim Tokenisieren mit dem Befehl «padding = True» angepasst werden. Allerdings führt dies zu langen Sequenzen, wenn nur wenige Sätze im Datensatz um einiges länger als alle anderen sind. Aus diesem Grund wurde, wie von Hugging Face empfohlen, der Data Collator zum dynamischen Padding verwendet. Dieser passt automatisch die jeweils miteinander verarbeiteten Einträge (ein sogenanntes «batch») an die Länge des längsten Eintrages an (Hugging Face, o. J.-c). Es werden also nur die Einträge, die mit den längsten Sätzen verarbeitet werden, an die Maximallänge angepasst, was viel effizienter in der Verarbeitung ist (ebd.). Der «DataCollatorForTokenClassification» sorgt dafür, dass die Labels gleich wie die Inputs verlängert werden und ebenfalls den vom Modell ignorierten Wert -100 erhalten (Hugging Face, o. J.-d).

4.2.2 Berechnen der Metriken

Zur Berechnung der Metriken wurde, wie in der Kursdokumentation zur Token Klassifizierung von Hugging Face empfohlen, die seqeval-Bibliothek genutzt. Das PythonFramework ist zur Evaluation von Sequenzlabeling-Aufgaben gedacht (Nakayama, 2018). Diese Funktion, die im Modell bei jeder Epoche die verschiedenen Evaluationsmetriken berechnet, wurde als solche von der Hugging Face-Kursdokumentation zur Token-Klassifizierung (Hugging Face, o. J.-d) übernommen. Es wurde präzisiert, dass die Labels, die -100 als Wert haben, bei der Evaluation nicht berücksichtigt werden, da es sich dabei um künstliche, bedeutungslose Labels handelt.

```python
def compute_metrics(eval_preds):
    """
    Übernommen von https://huggingface.co/course/chapter7/2?fw=pt
    """
    logits, labels = eval_preds
    predictions = np.argmax(logits, axis=-1)

    # Remove ignored index (special tokens) and convert to labels
    true_labels = [[label_names[l] for l in label if l != -100] for label in labels]
    true_predictions = [
        [label_names[p] for (p, l) in zip(prediction, label) if l != -100]
        for prediction, label in zip(predictions, labels)
    ]
    all_metrics = metric.compute(predictions=true_predictions, references=true_labels)
    return {
        "precision": all_metrics["overall_precision"],
        "recall": all_metrics["overall_recall"],
        "f1": all_metrics["overall_f1"],
        "accuracy": all_metrics["overall_accuracy"],
    }
```

Abb. 23: Funktion, die das Modell nach jeder Epoche evaluiert (eigene Grafik)

4.2.3 Modell definieren und feintunen

Nachdem nun die Daten vorbereitet und die Metriken definiert worden waren, konnte das Modell definiert und *feingetuned* («feingestimmt») werden.

Es wurde das in Kapitel 2.4.1.1 beschriebene, vortrainierte DistilBERT-Modell (Sanh et al., 2020) verwendet, das mehrsprachig und mit Berücksichtigung von Gross-/Kleinschreibung vortrainiert wurde (Hugging Face, o. J.-b). Es wurden die empfohlenen Standardeinstellungen verwendet.

```
from transformers import DataCollatorForTokenClassification
from transformers import AutoTokenizer

model_checkpoint = "distilbert-base-multilingual-cased"
tokenizer = AutoTokenizer.from_pretrained(model_checkpoint)

data_collator = DataCollatorForTokenClassification(tokenizer=tokenizer)
```

```
label_names = ['ADJ', 'ADP', 'ADV', 'AUX', 'CCONJ', 'DET', 'INTJ',
               'NOUN', 'NUM', 'PART', 'PRON', 'PROPN', 'PUNCT',
               'SCONJ', 'SYM', 'VERB', 'X', '_']

id2label = {str(i): label for i, label in enumerate(label_names)}
label2id = {v: k for k, v in id2label.items()}
```

```
from transformers import AutoModelForTokenClassification

model = AutoModelForTokenClassification.from_pretrained(
    model_checkpoint,
    id2label=id2label,
    label2id=label2id,
)
```

```
from transformers import TrainingArguments

args = TrainingArguments(
    "distilbert-pos-tagger",
    evaluation_strategy="epoch",
    save_strategy="epoch",
    learning_rate=2e-5,
    num_train_epochs=3,
    weight_decay=0.01,
)
```

```
from transformers import Trainer

trainer = Trainer(
    model=model,
    args=args,
    train_dataset=dataset,
    eval_dataset=load_from_disk('data/val_data_ger.hf'),
    data_collator=data_collator,
    compute_metrics=compute_metrics,
    tokenizer=tokenizer,
)
trainer.train()
```

Abb. 24: Aufbau des Feintuning-Modells (eigene Grafik)

Das Modell wurde mit bis zu sechs Epochen trainiert und nach jeder Epoche mittels den Validierungsdaten über die seqeval-Bibliothek evaluiert. Drei Epochen zu trainieren, dauerten für Deutsch auf dem Laptop mit einer CPU-Leistung von 1.80 GHz knapp 3.5 Stunden. Über Colab, wo auf GPU-Leistung zugegriffen werden konnte, dauerte das

Trainieren des gleichen Modells gut sieben Minuten. Es wurden jeweils acht Einträge («batches») parallel trainiert. Die trainierten Modelle konnten anschliessend lokal abgespeichert und als «Checkpoint» anstelle eines von Transformers zur Verfügung gestellten Modells aufgerufen werden. Das mit drei Epochen trainierte Modell zeigte in allen Sprachen die besten Ergebnisse sowohl auf dem Validierungs-, als auch auf dem Testdatensatz. Aus diesem Grund wurde dieses im Anschluss für den DistilBERT-POS-Tagger gewählt.

Epoch	Training Loss	Validation Loss	Precision	Recall	F1	Accuracy
1	0.110500	0.084791	0.973787	0.975607	0.974696	0.973383
2	0.065600	0.088571	0.975204	0.977742	0.976471	0.974317
3	0.039000	0.095897	0.976546	0.979544	0.978043	0.975447
4	0.025400	0.104332	0.976987	0.978944	0.977964	0.974759
5	0.014300	0.115307	0.977340	0.980145	0.978740	0.975742
6	0.009200	0.120916	0.976967	0.979477	0.978221	0.975718

Abb. 25: Evaluationswerte nach den einzelnen Epochen des trainierten Modells für Französisch (eigene Grafik)

4.3 Evaluation

Nachdem das DistilBERT-Modell für POS-Tagging auf Deutsch, Französisch und Englisch trainiert worden war, mussten die Tagger evaluiert werden. Dies wurde in mehreren Schritten gemacht:

1. Quantitative Evaluation des Taggers auf Wortebene
2. Quantitative Evaluation des Taggers auf Satzebene
3. Qualitative Evaluation des Taggers

Die Ergebnisse dieser Evaluation werden im nächsten Kapitel näher erläutert.

4.3.1 Vorbereiten der Daten

In einem ersten Schritt wurden die Testdaten für die Evaluation vorbereitet. Da die Wahrscheinlichkeit des zugeordneten Tags höher war, wenn die Sätze dem Modell nicht in Wort-Tokens getrennt übergeben werden, wurden die Wortlisten aus den Testdaten zu Sätzen zusammengesetzt. Die Satzzeichen und entsprechenden Tags wurden entfernt, weil das Modell durch die Entfernung dieser in der Vorverarbeitung nicht gelernt hatte, damit umzugehen.

```python
def prepare_data_for_evaluation(data):
    """
    Öffnet ein .conllu-file und wandelt es in die von Transformers geforderte Form um.
    Satzzeichen werden gelöscht und der entsprechende POS-Tag ebenfalls.
    Die einzelnen Wörter werden zu einem Satz zusammengesetzt.
    Elemente, deren Länge zwischen den Wörtern und den Tags nicht übereinstimmt, werden nicht berücksichtigt.
    :param data: .connlu-file der UD
    :return: Transformers-dataset für POS-Tagging mit features "word", "pos", "idx"
    :param data:
    :return:
    """
    data_file = open(data, "r", encoding="utf-8")
    saetze=[]
    for tokenlist in parse_incr(data_file):
        saetze.append(tokenlist)

    data={'idx':[], 'words': [], 'pos': []}
    idx = 0
    for element in saetze:
            woerter=[]
            postagswoerter=[]
            for e in element:
                woerter.append(e['form'])
                postagswoerter.append(e['upos'])
            #Satzzeichen rausnehmen
            pattern = r"[^a-zA-Z0-9ßëäöüïéèîôûâàùçËÄÖÜÏÈÉÀÙÂÎÔÛ]"
            woerter = [re.sub(pattern, '', wort) for wort in woerter]
            #POS-Tag Punktuation entfernen, da wir entsprechende Elemente aus den Wortlisten entfernen werden
            while 'PUNCT' in postagswoerter: postagswoerter.remove('PUNCT')
            while '' in woerter: woerter.remove('')
            #überprüfen, ob Anzahl Elemente mit Anz Pos-Tags übereinstimmt
            if len(woerter) == len(postagswoerter):
                data['idx'].append(idx)
                satz = " ".join(woerter)
                data['words'].append(satz)
                data['pos'].append(postagswoerter)
                idx+=1

    dataset = Dataset.from_dict(data)
    return dataset
```

Abb. 26: Funktion zur Vorbereitung der Daten für die Evaluation (eigene Grafik)

Im Anschluss wurden dem Datensatz vom trainierten Modell POS-Tags zugewiesen. Einige Wörter waren in der Tokenisierung nicht getrennt worden. Diese wurden im Datensatz gesucht und spezifisch als Einzelwörter erneut durch das Modell geschickt.

Für Wörter, die in mehrere Tokens unterteilt worden waren, wurde evaluiert, welches Token-POS-Tag mit dem höchsten Score zugeteilt worden war. Dieser Tag wurde dann für das gesamte Wort angenommen. Im Beispiel wurde das Wort «Allem» in die Tokens «Alle» und «m» unterteilt. Die Wortart «Pronomen» wurde mit einem höheren Score zugeteilt als die Wortart «Nomen». Für dieses Wort wird in der Evaluation daher ein Pronomen als vorhergesagter POS-Tag angenommen.

```
{'entity_group': 'PRON',
 'score': 0.5210863,
 'word': 'Alle',
 'start': 9,
 'end': 13},
{'entity_group': 'NOUN',
 'score': 0.41776463,
 'word': '##m',
 'start': 13,
 'end': 14},
```

Abb. 27: Wort, das beim Ausführen des POS-Taggers in 2 Tokens unterteilt wurde (eigene Grafik)

Mit diesen Schritten konnte eine Liste von vorhergesagten POS-Tags erstellt werden. Einige wenige Sätze stimmten trotz der Anpassungen in der Länge noch nicht zwischen den tatsächlichen und vorhergesagten Labels überein. Es handelte sich allerdings nur um

sehr wenige (zwischen zwei und vier Sätzen pro Sprache). Daher wurden sie aus dem Testdatensatz entfernt.

Da nun die Länge der vorhergesagten POS-Tags mit der Länge der tatsächlichen POS-Tags übereinstimmte, konnte mit der eigentlichen Evaluation begonnen werden.

4.3.2 Quantitative Evaluation auf Wort- und Satzebene

Um die Genauigkeit des DistilBERT-POS-Taggers auf Wortebene evaluieren zu können, wurde je eine Liste mit den vorhergesagten und den tatsächlichen POS-Tags erstellt. Die Satzanfänge und -enden wurden nicht gekennzeichnet, da ausschliesslich die Gesamt-Accuracy berechnet werden sollte.

Mittels den Metriken von Sklearn wurde damit ein Klassifikationsreport berechnet, der die Gesamt-Accuracy sowie die Precision, den Recall und den F1-Wert pro Wortart aufzeigte. Zudem wurde aufgeführt, wie viele Einträge pro Wortart im Testdatensatz existieren.

Um die Statistiken etwas anschaulicher zu machen, wurde mit Pandas eine Konfusionsmatrix erstellt und diese mit Hilfe von Seaborn grafisch dargestellt. Eine Heatmap auf den Matrizen ermöglichte auf einen Blick die Wortarten zu erkennen, die häufig miteinander vertauscht wurden.

Auf Satzebene sollte herausgefunden werden, wie viele Sätze vollkommen fehlerfrei sind, da bereits ein einziger Fehler schwerwiegende Folgen bei den darauf aufbauenden NLP-Aufgaben haben kann. Die vorhergesagten und tatsächlichen POS-Tags wurden daher pro Satz in einen String zusammengeführt.

```
def join_labels(liste):
    #die einzelnen Labels pro Satz werden zusammengeklebt, damit eine Genauigkeit auf Satzebene festgestellt werden
    neue_liste = []
    labels_joined = [" ".join(satz) for satz in liste]
    neue_liste.append(labels_joined)
    neue_liste=neue_liste[0]
    return neue_liste
```

Abb. 28: Funktion, die die Wortlisten in einen String pro Satz zusammenführt (eigene Grafik)

Im Anschluss wurde die durchschnittliche Anzahl Fehler pro fehlerhaftem Satz eruiert. Hierfür wurde zuerst mit Hilfe einer Funktion bei allen falschen Sätzen die Anzahl Fehler gezählt und anschliessend mit der Numpy-Bibliothek der Durchschnitt davon berechnet.

Da in allen Sprachen Nomen und Eigennamen am häufigsten verwechselt wurden, wurde die Evaluation ein zweites Mal durchgeführt, ohne einen Unterschied zwischen diesen beiden Wortarten zu machen. Dieses Vorgehen wurde gewählt, da das Erkennen von Eigennamen die Aufgabe der Named-Entity-Recognition ist und einen eigenen

Algorithmus erfordert. Zudem ist für auf das POS-Tagging aufbauende Aufgaben eine Unterscheidung der beiden Nomen-Arten nicht zwingend entscheidend.

4.3.3 Quantitative Evaluation der Vergleichstagger

Die Vergleichstagger taggten dieselben Testdaten, wie zuvor DistilBERT. Die Daten wurden mit wenigen Ausnahmen genau gleich vorverarbeitet.

Da der Tag «X – andere» nur sehr restriktiv genutzt wird und es hier eine grosse Interpretationsspannweite in der Annotation der Trainingsdaten geben kann, wurden für den Vergleich keine Wörter berücksichtigt, die im Original-Testdatensatz mit «X» getaggt worden waren. Das gleiche gilt für den Tag «_», der Verschmelzungen zweier Wörter (im, vom, …) bezeichnet. Je nach Tokenisierungsverfahren (z.B. SpaCy) werden diese dabei getrennt, was zu unberechtigten Fehlern in der Evaluation führt. Der Vollständigkeit halber und um einen korrekten Vergleich zu ermöglichen, wurde auch der DistilBERT-Datensatz ein zweites Mal evaluiert, ohne die genannten Tags zu berücksichtigen. Das Ergebnis veränderte sich dadurch nur minim.

Für den quantitativen Vergleich wurde eine Konfusionsmatrix gewählt, die die zugeordneten und tatsächlichen Tags prozentual statt in absoluten Zahlen anzeigt. Dies, um einen direkten Vergleich zwischen den Taggern zu ermöglichen und die Performance der Wortarten auf einen Blick erkennen zu können.

4.3.3.1 Stanford

Der Stanford-Tagger funktioniert nach der Methode der maximalen Entropie mit einem zyklischem Abhängigkeitsnetz (Toutanova et al., 2003), wie in Kapitel 3.1.3 erläutert. Da er in Java geschrieben ist, wurde dieser über einen Umweg in Python implementiert. Dabei wurde die Anleitung von Bartsch (2019) genutzt. Mit dieser konnte der POS-Tagger über Python aufgerufen werden, während er im Hintergrund auf die Java-Installation auf dem lokalen Rechner zugreifen konnte.

Um die Evaluation mit dem Stanford-Tagger vorzunehmen, wurden die Sätze dem Modell in Wort-Tokens übergeben. Da auf Englisch mit dem Penn Treebank POS-Tagset (University of Pennsylvania. Departement of Linguistics, 2003) getaggt wird, mussten die vergebenen Tags in die UPOS-Tags übersetzt werden. Hierfür wurde ein Dictionary erstellt (siehe Anhang 1). Die Zuordnung konnte jedoch nicht immer eindeutig vorgenommen werden, was sich in den Ergebnissen widerspiegelte (siehe Kapitel 5.2.1).

4.3.3.2 SpaCy

Der SpaCy-Tagger basiert auf einem neuronalen Netzwerk und dem Aufmerksamkeitsmechanismus (Honnibal & Montani, 2017, zitiert nach Partalidou et al., 2019). Er existiert in den drei getesteten Sprachen. Dafür muss ein entsprechender Sprachkorpus geladen werden. Beim Tokenisieren trennt SpaCy verbundene Wörter (beispielsweise *vom* → [von, dem]). Die Universal Dependencies-Datensätze enthalten entsprechende Wörter einmal zusammen (mit dem Tag «_») und einmal getrennt (mit den entsprechenden Tags, beispielsweise [ADP, DET]). Damit die von SpaCy getaggten Sätze nicht länger als das Original waren, wurden alle Kontraktionen und ihre Tags aus dem Testdatensatz entfernt, bevor sie dem SpaCy-Tagger übergeben wurden. SpaCy trennt auch in einem Wort zusammengesetzte Metriken (beispielsweise *200m*→ [200, m]). Davon gab es jedoch nur eine sehr kleine Anzahl. Aus diesem Grund wurden entsprechende Sätze gleichzeitig mit den anderen Sätzen, die aus einem Grund nach dem Taggen eine ungleiche Länge zwischen dem Original- und Testdatensatz aufwiesen, entfernt.

Im Gegensatz zu den anderen vortrainierten englischen POS-Tagger wurde der SpaCy-Tagger auf UPOS-Tags trainiert. Somit war keine Übersetzung der Tags notwendig.

4.3.3.3 NTLK

Der Standard-POS-Tagger von NLTK basiert auf der Averaged Perceptron-Methodik, einer Sonderform des Hidden Markov Modells (Honnibal, 2013; NLTK Project, 2022a), das in Kapitel 3.1.2 beschrieben wurde. NLTK-Tagger sind derzeit nur auf Englisch trainiert verfügbar. Aus diesem Grund musste er für Französisch und Deutsch erst trainiert werden. Dafür wurden die gleichen Daten aus den Universal Dependencies verwendet, die bereits für den DistilBERT-POS-Tagger zum Einsatz gekommen waren. Um einen korrekten Vergleich zu gewährleisten, wurde auch der englische Tagger mit diesen Daten neu trainiert. Wäre der vortrainierte POS-Tagger verwendet worden, hätten die POS-Tags für das Englische erneut aus dem Penn-Treebank-Datensatz in die Universal POS-Tags übersetzt werden müssen.

```
[('Sehr', 'ADV'),
 ('gute', 'ADJ'),
 ('Beratung', 'NOUN'),
 ('schnelle', 'ADJ'),
 ('Behebung', 'NOUN'),
 ('der', 'DET'),
 ('Probleme', 'NOUN'),
 ('so', 'ADV'),
 ('stelle', 'VERB'),
 ('ich', 'PRON'),
 ('mir', 'PRON'),
 ('Kundenservice', 'NOUN'),
 ('vor', 'ADP')]
```

Abb. 29: Von NLTK erwartetes Datenformat

Für das Training mussten die UD-Datensätze in die von NLTK erwartete Form gebracht werden. Diese besteht aus einer Liste mit Listen pro Satz, wobei jede Satz-Liste Tupel mit der Wort-POS-Tag-Kombination enthält (siehe Beispiel rechts).

```
def prepare_data_for_nltk(data):
    """
    Öffnet ein .conllu-file und wandelt es in die von NLKT geforderte Form um.
    Satzzeichen werden gelöscht und der entsprechende POS-Tag ebenfalls.
    :param data: .connlu-file der UD
    :return: Liste von Listen je Satz mit Tupeln pro Wort/POS-Tag
    """
    data_file = open(data, "r", encoding="utf-8")
    saetze=[]
    for tokenlist in parse_incr(data_file):
        saetze.append(tokenlist)

    dataset=[]
    for element in saetze:
        satz=[]
        for e in element:
            #Satzzeichen rausnehmen
            if e['upos'] != 'PUNCT':
                satz.append((e['form'], e['upos']))
        dataset.append(satz)
    return dataset
```

Abb. 30: Funktion zur Vorbereitung der Trainingsdaten für NLTK (eigene Grafik)

Anschliessend wurde der untrainierte Perceptron-Tagger aus NLTK mit dem Befehl «PerceptronTagger(load=False)» aufgerufen und mit den Trainingsdaten der Universal Dependencies trainiert.

Erst in einem zweiten Schritt konnten die Testdaten mit dem neu trainierten NLTK-Tagger getaggt und evaluiert werden.

4.3.4 Qualitative Evaluation

Für die qualitative Evaluation wurde einerseits eine Liste mit Homographen-Sätzen für jede Sprache erstellt, die anschliessend von den verschiedenen Taggern getaggt wurden. Diese Sätze sind in den Anhängen 2-4 zu finden.

Andererseits wurden alle Sätze aus dem Testdatensatz gefiltert, die nur vom DistilBERT-Tagger korrekt getaggt wurden. Dafür wurden die POS-Tag-Satzlisten aller Tagger miteinander verglichen. Da für SpaCy zusammengezogene Wörter mit dem Tag «_» entfernt worden waren, musste dieser in den Originaltestdaten vor dem Vergleich ebenfalls entfernt werden.

Zusätzlich wurden alle Sätze gefiltert, die von allen Taggern korrekt, vom DistilBERT-Tagger jedoch fehlerhaft getaggt wurden. In beiden Fällen wurden die Fehler genauer analysiert. Zuletzt wurde der Satz mit den meisten Fällen eruiert und angezeigt.

In einem letzten Schritt wurde anhand der Klassifikationsreports pro Wortart Sätze aus dem Testdatensatz gesucht, die von DistilBERT im Gegensatz zum nächstbesseren Tagger richtig getaggt wurden. Das gleiche wurde durchgeführt bei Wortarten, bei denen ein anderer Tagger die höchsten Werte erzielte. Hier wurden Sätze mit entsprechenden Wörtern gesucht, die vom besten Tagger korrekt und von DistilBERT falsch getaggt wurden.

5 Ergebnisse

Das folgende Kapitel zeigt die Ergebnisse der Evaluation des DistilBERT-POS-Taggers auf und vergleicht diese mit den Ergebnissen von drei bestehenden Taggern, die auf unterschiedlichen Methoden basieren. Nach einer quantitativen Analyse der Ergebnisse werden einige konkreten Beispiele näher angeschaut, die vom DistilBERT-Tagger besser oder schlechter als von den anderen Taggern getaggt wurden. Alle Ergebnisse basieren auf den ausgewählten Testdatensätzen der Universal Dependencies.

5.1 Quantitative Evaluation

Die quantitative Evaluation des DistilBERT POS-Taggers zeigte für alle Sprachen hohe Genauigkeitswerte. Am höchsten waren diese für Französisch, am niedrigsten für Deutsch. Die Unterschiede beliefen sich auf Wortebene auf wenige Prozentpunkte, auf Satzebene waren die Unterschiede bereits etwas grösser. Der deutsche DistilBERT-POS-Tagger war der einzige Tagger, der keine Accuracy auf Satzebene über 50% erreichte. Auch mit der Gleichbehandlung aller Nomenklassen (Nomen und Eigennamen) blieb dieser Wert knapp darunter. Dafür war auf Deutsch die durchschnittliche Anzahl Fehler pro falschem Satz am niedrigsten. Dies spricht dafür, dass der Tagger im Deutschen trotz einer fehlerhaften Klassierung die anderen Wörter eines Satzes besser richtig zuordnen konnte, wohingegen in den anderen Sprachen das Risiko weiterer Fehler höher war.

Wenn Nomen und Eigennamen als gleiche Klasse gelten, konnte die Genauigkeit insbesondere auf Satzebene erhöht werden. Die durchschnittliche Anzahl Fehler pro falschem Satz reduzierte sich insbesondere im Deutschen sichtbar.

	Deutsch	**Französisch**	**Englisch**
Accuracy auf Wortebene	93%	96%	94%
Accuracy auf Satzebene	42%	56%	53%
Durchschnittliche Anzahl Fehler pro falschem Satz	1.76	1.8	2.1
Keine Unterscheidung zwischen Nomen (NOUN) und Eigennamen (PROPN)			
Accuracy auf Wortebene	95%	97%	95%
Accuracy auf Satzebene	49%	62%	60%

Durchschnittliche Anzahl Fehler pro falschem Satz	1.54	1.7	1.81

Tabelle 4: Quantitativer Vergleich des DistilBERT POS-Taggers zwischen den evaluierten Sprachen (eigene Tabelle)

Insgesamt erreichte der DistilBERT-Tagger in allen Sprachen Accuracy-Werte von über 90% auf Wortebene und auf Satzebene in immerhin zwei Sprachen von über 50%.

5.1.1 Deutsch

Der DistilBERT-POS-Tagger konnte eine Accuracy von 93% auf Wortebene für den Testdatensatz erreichen, was einer Fehlerrate von 7% entspricht. Es kann beobachtet werden, dass insbesondere Adjektive und Eigennamen vom Modell schlecht erkannt wurden. Interjektionen waren in den Daten kaum vorhanden und der tiefe Wert ist somit nicht repräsentativ.

	precision	recall	f1-score	support
ADJ	0.78	0.93	0.85	1029
ADP	0.99	0.99	0.99	1587
ADV	0.92	0.81	0.86	1255
AUX	0.94	0.96	0.95	678
CCONJ	0.96	0.94	0.95	453
DET	0.97	0.98	0.98	2007
INTJ	0.00	0.00	0.00	3
NOUN	0.94	0.95	0.95	3086
NUM	0.91	0.97	0.94	236
PART	0.97	0.95	0.96	205
PRON	0.93	0.90	0.92	896
PROPN	0.86	0.83	0.85	1009
SCONJ	0.95	0.87	0.90	159
VERB	0.96	0.94	0.95	1307
X	0.18	0.33	0.24	21
_	1.00	0.99	1.00	273
accuracy			0.93	14204
macro avg	0.83	0.83	0.83	14204
weighted avg	0.93	0.93	0.93	14204

Abb. 31: Klassifikations-Report deutscher Testdatensatz (eigene Grafik)

Wird ein Blick auf die Konfusionsmatrix geworfen, kann erkannt werden, dass Nomen häufig mit Eigennamen und umgekehrt verwechselt wurden. Dies ist nicht besonders erstaunlich, da es sich beides Mal um eine Art von Nomen handelt und auch Menschen, insbesondere Fremdsprachige, diese beiden nicht immer unterscheiden können. Eine ähnliche Begründung kann bei der Verwechslung zwischen Verben und Hilfsverben aufgestellt werden; es handelt sich beide Male um Verbformen. Hilfsverben können zudem auch als normale Verben vorkommen (Ich bin/VERB im Zoo vs. Ich bin/AUX in den Zoo gegangen).

Adverbien wurden zudem häufig mit Adjektiven verwechselt. Da diese beiden Wortarten im Deutschen nur durch ihre grammatikalische Zugehörigkeit und nicht das Wort selbst unterschieden werden können, können auch diese Fehler nachvollzogen werden. Pronomen (PRON) und Artikel (DET) wurden relativ häufig gegenseitig verwechselt, was vermutlich an der teilweise ähnlichen Position im Satz liegt (alle/PRON Hunde vs. die/DET Hunde).

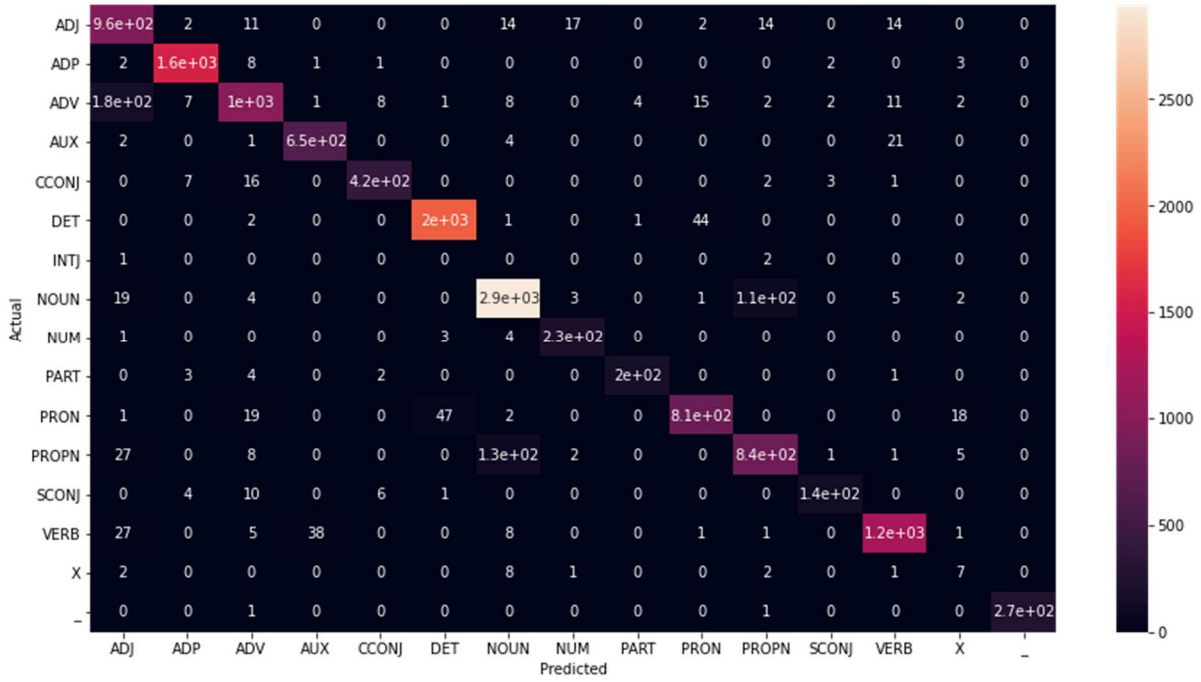

Abb. 32: Konfusionsmatrix des deutschen Testdatensatzes (eigene Grafik)

Auf Satzebene konnte nur eine Accuracy von 42% erreicht werden. Mit einer Fehlerrate von 58% enthielten über die Hälfte der Sätze des Testdatensatzes mindestens einen Fehler. Durchschnittlich waren dies 1.76 Fehler pro Satz.

Wurden Nomen und Eigennamen als gleichwertige Nomen behandelt, konnte die Fehlerrate auf Wortebene um zwei Prozentpunkte auf 5% und die durchschnittliche Anzahl Fehler pro falschem Satz auf 1.54 verringert werden. Die Accuracy vollständig korrekt getaggter Sätze stieg auf fast 50%.

5.1.2 Französisch

Auf Französisch erreichte der DistilBERT-POS-Tagger einen Accuracy-Wert von 96% auf Wortebene und von 56% auf Satzebene, was verglichen mit den anderen Sprachen ein Höchstwert ist. Die durchschnittliche Anzahl Fehler pro falschem Satz war mit 1.8 Fehlern jedoch etwas höher als im Deutschen. Dies könnte daran liegen, dass französische Sätze oftmals sehr lang sind.

Adpositionen, deren bekannteste Vertreter die Präpositionen (z.B. «sur») sind (Universal Dependencies, 2021r), nebenordnende Konjunktionen (z.B. «et») und Determinative (z.B. «le») wurden vollständig korrekt zugeordnet. Auch in diesem Datensatz waren kaum Interjektionen vorhanden, was die tiefen Werte nicht repräsentativ macht. Mit einem F1-Score von 0.92, bzw. 0.93 wurden Pronomen und Eigennamen am schlechtesten zugeordnet.

	precision	recall	f1-score	support
ADJ	0.95	0.94	0.94	2108
ADP	1.00	1.00	1.00	5486
ADV	0.98	0.98	0.98	1209
AUX	0.99	0.90	0.94	1079
CCONJ	0.99	1.00	1.00	886
DET	1.00	1.00	1.00	5279
INTJ	0.14	0.33	0.20	6
NOUN	0.95	0.98	0.96	6458
NUM	0.99	0.99	0.99	870
PRON	0.98	0.87	0.92	1489
PROPN	0.94	0.92	0.93	2507
SCONJ	0.99	0.97	0.98	241
SYM	0.14	0.27	0.18	11
VERB	0.97	0.94	0.96	2674
X	0.14	0.31	0.20	156
_	1.00	1.00	1.00	1007
accuracy			0.96	31466
macro avg	0.82	0.84	0.82	31466
weighted avg	0.97	0.96	0.97	31466

Abb. 33: Klassifikations-Report französischer Testdatensatz (eigene Grafik)

Ein Blick auf die Konfusionsmatrix verrät, dass Hilfsverben (AUX) erstaunlicherweise nicht am häufigsten mit Verben (VERB), sondern mit Nomen verwechselt wurden. In einigen Fällen wurden sie auch als «X - andere» klassiert. Auch Adjektive und Verben wurden in einigen Fällen als Nomen zugeteilt. Der Verdacht liegt nahe, dass dem im Deutschen durch die Gross-/Kleinschreibung vorgebeugt werden konnte. Aus dem gleichen Grund wurden aber vermutlich Eigennamen besser zugeordnet. Mit einigen wenigen Ausnahmen werden im Französischen nur diese ausserhalb des Satzanfanges grossgeschrieben.

Weiter fällt auf, dass auffallend viele Pronomen nicht zugeteilt werden konnten und das Label «X – andere» erhielten.

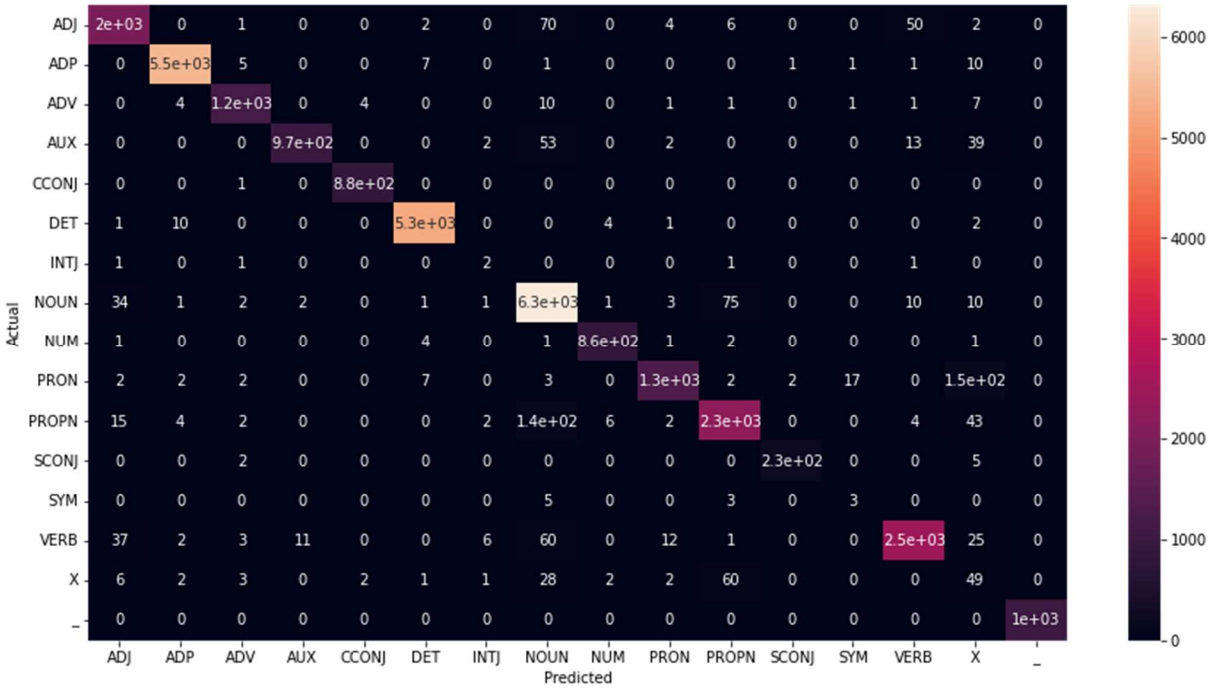

Abb. 34: Konfusionsmatrix des französischen Testdatensatzes (eigene Grafik)

Obwohl Nomen und Eigennamen vom französischen DistilBERT-POS-Tagger generell seltener vertauscht wurden als von seinem deutschen Pendant, war dies neben den mit X getaggten Pronomen die zweitgrösste Verwechslungsgruppe. Aus diesem Grund wurde auch hier die Evaluation ein zweites Mal durchgeführt, ohne zwischen Nomen und Eigennamen einen Unterschied zu machen. Auf Satzebene konnte so die Error-Rate auf knapp 38% reduziert werden. Allerdings veränderten sich die Anzahl Fehler pro falschem Satz kaum. Dies ist nicht verwunderlich, wenn bedacht wird, dass Verwechslungen zwischen Nomen und Eigennamen, bzw. Eigennamen und Nomen im Französischen nicht die Hauptfehlerquelle war.

Auf Wortebene konnte die Accuracy mit der Gleichbehandlung von Nomen und Eigennamen immerhin um einen Prozentpunkt auf 97% gesteigert werden.

5.1.3 Englisch

Mit einer Accuracy von 94% auf den Testdaten, positioniert sich der englische DistilBERT-POS-Tagger genau zwischen den Werten für Deutsch und Französisch. Obwohl die Interjektionen in diesem Datensatz etwas stärker vertreten waren, erreichten sie keine besonders hohen Werte. Adverbien und Eigennamen wiesen den niedrigsten F1-Score auf. Im Gegenzug erreichten nebenordnende Konjunktionen (CCONJ), Artikel (DET) und Pronomen (PRON) beinahe Höchstwerte.

	precision	recall	f1-score	support
ADJ	0.90	0.94	0.92	1265
ADP	0.98	0.96	0.97	1919
ADV	0.88	0.87	0.88	847
AUX	0.98	0.87	0.92	894
CCONJ	0.99	1.00	0.99	664
DET	0.99	0.98	0.99	1647
INTJ	0.42	0.80	0.56	145
NOUN	0.91	0.96	0.93	3372
NUM	0.95	0.98	0.97	332
PART	0.98	0.99	0.99	413
PRON	0.99	0.99	0.99	1380
PROPN	0.96	0.74	0.83	1272
SCONJ	0.92	0.92	0.92	296
SYM	0.00	0.00	0.00	2
VERB	0.93	0.97	0.95	2035
X	0.37	0.27	0.31	26
_	1.00	0.99	1.00	256
accuracy			0.94	16765
macro avg	0.83	0.84	0.83	16765
weighted avg	0.94	0.94	0.94	16765

Abb. 35: Klassifikations-Report englischer Testdatensatz (eigene Grafik)

Aus der Konfusionsmatrix kann herausgelesen werden, dass Eigennamen häufig mit Nomen verwechselt wurden, die Verwechslung umgekehrt jedoch weitaus seltener war. Interessanterweise wurden Eigennamen neben Nomen auch mit Adjektiven und Interjektionen vertauscht, und Nomen hauptsächlich mit Verben verwechselt. Adverbien und Hilfsverben wurden vom Modell bei je etwa 50 Wörtern fälschlicherweise als Interjektionen getaggt.

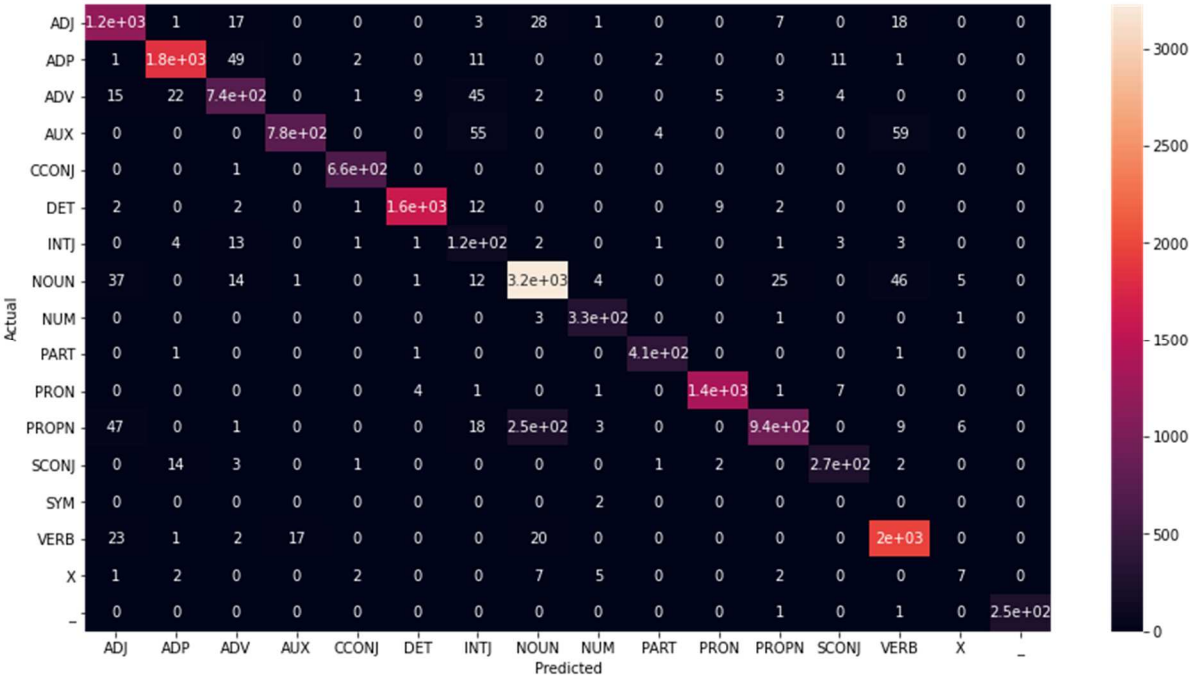

Abb. 36: Konfusionsmatrix des englischen Testdatensatzes (eigene Grafik)

Auf Satzebene erreichte der DistilBERT-Tagger eine Accuracy von 53%. Mit 2.1 falsch getaggten Wörtern war die Fehlerrate pro falschem Satz jedoch am höchsten. Es scheint, als würde die POS-Tag-Folge einen noch wichtigeren Einfluss als in den anderen

Sprachen haben. Ist ein Wort falsch getaggt, scheint die Wahrscheinlichkeit hoch zu sein, dass mindestens noch ein zweites Wort vom Modell ebenfalls falsch zugeordnet wird.

Obwohl im Englischen nur in Ausnahmefällen andere Wörter als Eigennamen ausserhalb von Satzanfängen mit einem Grossbuchstaben beginnen, verwechselte der Tagger diese oftmals mit Nomen. Aus diesem Grund wurde erneut eine Evaluation vorgenommen, nachdem alle PROPN- und NOUN-Tags als NOUN vereinheitlicht wurden. Die Error-Rate verringerte sich dabei um einen Prozentpunkt und betrug noch 5%. Die grösste Verbesserung konnte jedoch auf Satzebene erreicht werden, auf der sich die Accuracy von 53% auf 60% erhöhte. Die durchschnittliche Anzahl Fehler pro falschem Satz konnte auf 1.81 Fehler reduziert werden.

5.2 Quantitativer Vergleich mit anderen Taggern

Ein Vergleich auf dem Testdatensatz über alle Sprachen zeigt, dass der auf Transformers basierende DistilBERT-Tagger insgesamt die beste Accuracy erreichte. Die Unterschiede zu den anderen Taggern waren jedoch nicht immens und betrugen häufig nur wenige Prozentpunkte.

SpaCy, ein Tagger, der ebenfalls auf Deep Learning basiert, stach im Vergleich mit den nicht auf Deep Learning basierenden Taggern nicht hervor. Auf Französisch erreichte der SpaCy-Tagger sogar mit Abstand die schlechtesten Accuracy-Werte. Auf Deutsch erreichte der SpaCy-Tagger nach dem DistilBERT-Tagger die zweitbesten Werte, die Unterschiede betrugen zum nächsthöheren und -tieferen Wert jedoch nur einen einzigen Prozentpunkt. Auffallend tiefe Werte zeigte der Stanford-Tagger auf Englisch. Wie in Kapitel 5.2.1 beschrieben wird, liegt dies aber vermutlich daran, dass die Penn Treebank-POS-Tags für den Vergleich, in die Universal POS-Tags übersetzt werden mussten. Für Französisch erreichte der Stanford-Tagger die zweitbesten Accuracy-Werte.

Der NLTK-Perceptron-Tagger lag im Mittelfeld. Auf Englisch zeigte er den zweithöchsten Accuracy-Wert, auf Deutsch lag er mit dem Stanford-Tagger gleichauf.

	DistilBERT	Stanford	NLTK	SpaCy
Deutsch	**93%**	91%	91%	92%
Französisch	**97%**	95%	94%	87%
Englisch	**94%**	77%	93%	92%

Tabelle 5: Vergleich der Accuracies der verschiedenen Tagger auf Wortebene (eigene Tabelle)

Etwas differenzierter zeigt sich das Bild, wenn die Genauigkeit der verschiedenen Tagger auf Satzebene verglichen wird. Hier erreichte der DistilBERT-Tagger mit Abstand die höchsten Werte in allen drei Sprachen.

Für Deutsch erreichte der SpaCy-Tagger mit sechs Prozentpunkten Unterschied zum DistilBERT-Tagger den zweithöchsten Wert. Stanford, der zweitbeste französische Tagger, erreichte um zehn Prozentpunkte niedrigere Accuracy-Werte als der Distil-BERT-Tagger. Neun Prozentpunkte betrug schlussendlich der Unterschied von der Accuracy auf Satzebene auf Englisch zwischen NLTK und DistilBERT.

Am schlechtesten schnitt für den deutschen Testdatensatz der NLTK-Tagger ab sowie der SpaCy-Tagger für Französisch. Nachdem sich der Stanford-Tagger für Englisch nicht direkt vergleichen lässt, kommen die niedrigsten Werte auch für diese Sprache von SpaCy.

	DistilBERT	Stanford	NLTK	SpaCy
Deutsch	**43%**	34%	32%	37%
Französisch	**57%**	41%	39%	13%
Englisch	**53%**	14%	44%	42%

Tabelle 6: Vergleich der Accuracies der verschiedenen Tagger auf Satzebene (eigene Tabelle)

Wird die durchschnittliche Anzahl Fehler pro falschem Satz betrachtet, erreichte nicht mehr zwingend der DistilBERT-Tagger die besten Werte. Dies kann damit erklärt werden, dass mit der Transformers-Methodik die Position im Satz und die Zusammenhänge zwischen den Tags stärker berücksichtigt wird. Somit hat ein Fehler einen grösseren Einfluss auf die anderen Tags und die Wahrscheinlichkeit, dass noch ein zweites oder drittes Wort dieses Satzes falsch klassiert wird, ist ungleich grösser, als wenn das einzelne Wort stärker isoliert betrachtet wird.

Der Stanford-Tagger erreichte, in allen Sprachen gemeinsam betrachtet, die niedrigste Fehlerquote in falschen Sätzen.

	DistilBERT	Stanford	NLTK	SpaCy
Deutsch	2.1	**1.7**	2	1.8
Französisch	1.8	**1.4**	2	3.3
Englisch	2.1	4.4	2.3	**2**

Tabelle 7: Durchschnittliche Anzahl Fehler in fehlerhaften Sätzen (eigene Tabelle)

Diese Ergebnisse zeigen, dass schon wenige Prozentpunkte Unterschied in der Accuracy auf Wortebene, die Anzahl vollständig korrekt getaggter Sätze massiv beeinflusst.

Die Transformers-Technologie führt insgesamt zu besseren Tagging-Ergebnissen als bisherige Methoden.

5.2.1 Stanford-Tagger

Der Stanford-Tagger erreichte auf dem deutschen Testdatensatz eine leicht schlechtere Accuracy als der DistilBERT-Tagger. Auf Wortebene lag der Unterschied nur bei zwei Prozentpunkten, auf Satzebene war das Ergebnis des Stanford-Taggers gar um ganze neun Prozentpunkte tiefer.

Ähnlich wie im DistilBERT-POS-Tagger wurden vom Stanford-Tagger auf Deutsch Nomen häufig mit Eigennamen und Adverbien mit Adjektiven verwechselt. Eigennamen wurden vom Stanford-Tagger jedoch seltener mit Nomen vertauscht als vom DistilBERT-Tagger. Dafür schien der Stanford-Tagger mehr Schwierigkeiten zu haben, Verben nicht mit Hilfsverben zu vertauschen. Auch Determinative taggte er leicht öfter falsch als Pronomen als der DistilBERT-Tagger. Auffallend ist, dass der Stanford-Tagger seltener mit «X – andere» taggte. Der Verdacht liegt nahe, dass dies an den Daten liegt, mit denen er trainiert wurde.

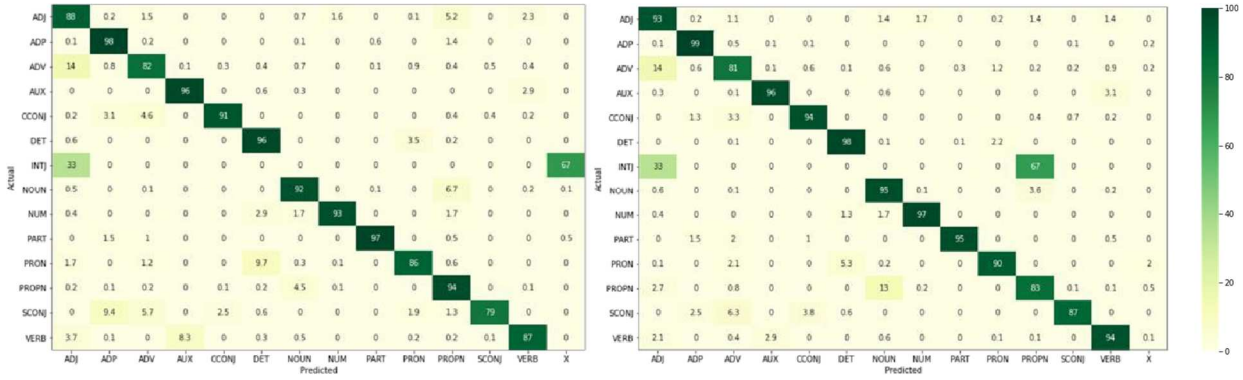

Abb. 37: Vergleich Stanford - DistilBERT Deutsch (eigene Grafik)

Auf Französisch schnitt der Stanford-Tagger auf Wortebene nur mit zwei Prozentpunkten weniger in der Accuracy als der DistilBERT-POS-Tagger ab. Auf Satzebene betrug die Differenz bereits 21 Prozentpunkte (41% Stanford vs. 62% DistilBERT).

Die besseren Ergebnisse erreichte der Stanford-Tagger mit den Hilfsverben (AUX), die er in fast allen Fällen richtig zuordnete und wenn, dann mit Verben verwechselte. Mehr Schwierigkeiten hatte er im Gegensatz zum DistilBERT-Tagger mit Adverbien. Er verwechselte diese am häufigsten mit Pronomen oder konnte sie nicht zuordnen und taggte sie mit «X – andere». Mehr Falschzuweisungen als der DistilBERT-Tagger machte der Stanford-Tagger bei Nomen. Neben der häufigsten Verwechslung mit Eigennamen, taggte er diese fälschlicherweise unter anderem auch als Adjektiv oder Verb.

Eigennamen hingegen wurden von beiden Taggern etwa gleich häufig mit gewöhnlichen Nomen verwechselt.

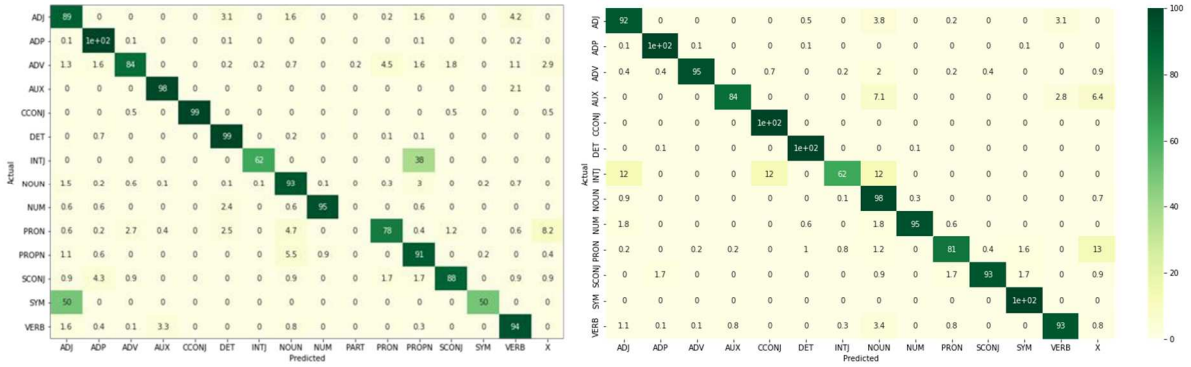

Abb. 38: Vergleich Stanford - DistilBERT Französisch (eigene Grafik)

Der Stanford-Tagger, der als Benchmark-POS-Tagger gilt, auf Englisch nur eine Accuracy von 77% auf Wortebene. Somit ist es nicht verwunderlich, dass auf Satzebene eine Error Rate von 86% folgte. Dies kann grösstenteils damit erklärt werden, dass in dieser Arbeit mit den Universal POS-Tags gearbeitet wurde, der englische Stanford-Tagger jedoch die Penn Treebank POS-Tags nutzt. Somit mussten die Tags «übersetzt» werden, was nicht immer einwandfrei ausgeführt werden konnte. Beispielsweise enthält die Penn Treebank-Klasse «IN» sowohl Präpositionen als auch untergeordnete Konjunktionen (Universal Dependencies, 2021p).

Adpositionen erreichten einen F1-Score von nur 0.06. Vermutlich liegt dies am im Englischen häufig verwendeten «to». In den Penn Treebank-Tags hat dieses Wort eine eigene Klasse, egal ob es als Konjunktion oder Präposition genutzt wird (Universal Dependencies, 2021c). In dieser Arbeit wurde die Penn Treebank-Klasse «TO» auf «ADP» getaggt, wodurch alle «to», die keine Präpositionen sind, fälschlicherweise als falsch gerechnet wurden.

Weiter gab es eine grosse Diskrepanz zwischen Hilfsverben (AUX) und Verben (VERB). Dies liegt daran, dass in der Penn Treebank nur Modalverben separat ausgewiesen werden. Alle anderen Hilfsverben gelten als «gewöhnliche» Verben (Universal Dependencies, 2021f). Die gleiche Erklärung gilt auch für die scheinbar häufige Verwechslung von Partikeln (PART) mit anderen Klassen. In der Penn Treebank sind diese über verschiedene Klassen verteilt und können somit nur für die Genitiv-Partikel («'s»), die eine eigene Penn-Treebank-Klasse bilden (Universal Dependencies, 2021m), eindeutig zugeteilt werden.

Somit kann der quantitative Vergleich zwischen dem englischen DistilBERT-POS-Tagger und dem englischen Stanford-Tagger nicht als signifikant betrachtet werden. Wie der

spätere qualitative Vergleich zeigen wird, ist der Stanford-Tagger dem DistilBERT-Tagger jedoch durchaus konkurrenzfähig.

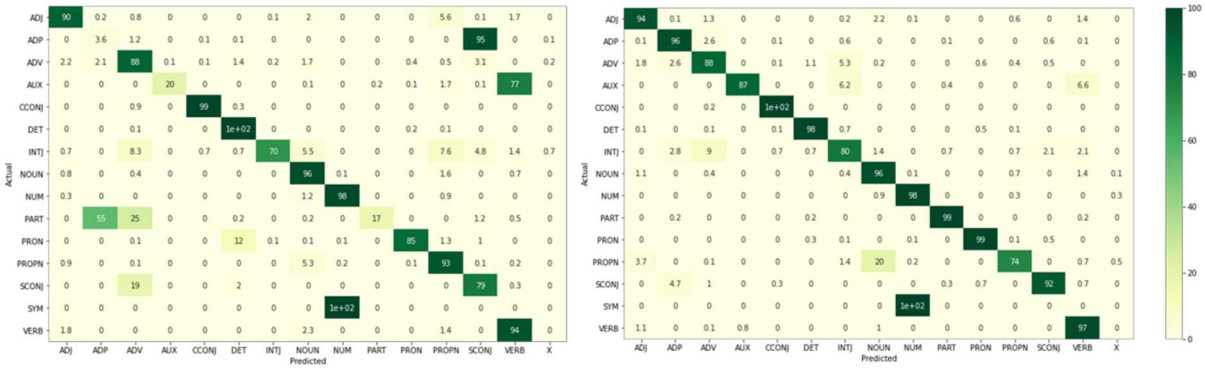

Abb. 39: Vergleich Stanford - DistilBERT Englisch (eigene Grafik)

Insgesamt erreichte der Stanford-Tagger sehr gute Ergebnisse und war in einer Minderheit von Wortarten sogar besser als der DistilBERT-Tagger. Doch einige Wortarten, die ihm Mühe bereiteten, führten dazu, dass das Gesamtergebnis trotzdem schlechter als dasjenige des DistilBERT-Taggers war.

5.2.2 NLTK

Der auf Deutsch trainierte Perceptron-NTLK-Tagger erreichte auf dem Testdatensatz eine Wort-Accuracy von 91%, was nur zwei Prozentpunkte unter dem Wert des DistilBERT-Taggers liegt. Auf Satzebene sank dieser Wert um elf Prozentpunkte, auf 32%.

Grosse Schwierigkeiten bereiteten dem NLTK-Tagger insbesondere Nomen und Eigennamen. Beide Klassen wurden von ihm noch häufiger als vom DistilBERT-Tagger vertauscht. Adjektive wurden häufiger als Eigennamen oder Verben getaggt und Hilfsverben öfters mit anderen Verben vertauscht. Hingegen taggte NLTK Eigennamen seltener als Adjektive, als dies der DistilBERT-Tagger tat. Im Grossen und Ganzen wies der NTLK-Tagger jedoch dieselben Schwächen wie der DistilBERT-Tagger auf, wenngleich er die gleichen Verwechslungen in erhöhter Zahl machte.

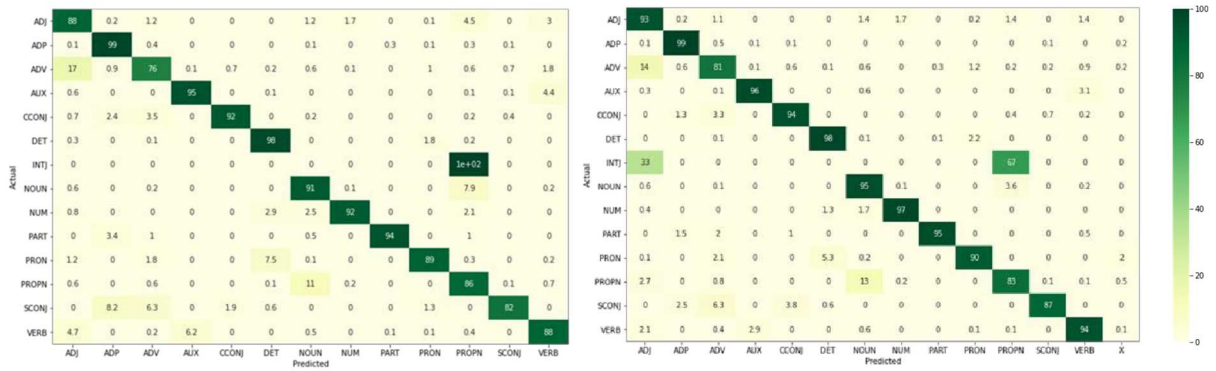

Abb. 40: Vergleich NLTK - DistilBERT Deutsch (eigene Grafik)

Auf Französisch auf Wortebene war der NLTK-Tagger um drei Prozentpunkte schlechter als der DistilBERT-POS-Tagger und erreichte eine Accuracy von 94%. Auf Satzebene erreichte das Perceptron-Verfahren von NLTK einen Wert von 39%, was um 23 Prozentpunkte niedriger als der Wert des DistilBERT-Taggers ist. Auffallend ist, dass im Gegensatz zum DistilBERT-Tagger, vom NLTK-Tagger weniger Eigennamen mit Nomen und mehr Nomen mit Eigennamen verwechselt wurden. Viele Wortarten wurden fälschlicherweise als Verben getaggt. Hilfsverben hingegen wurden vom NLTK-Tagger besser zugewiesen. Einige wurden zwar mit Verben verwechselt, doch dies kann als eine weniger gravierende Verwechslung als diejenige mit Nomen, die dem DistilBERT-Tagger unterliefen, betrachtet werden.

Adjektive und Adverbien wurden zudem einige Male mit Nomen und Verben verwechselt, der DistilBERT-Tagger tat dies nur bei Adjektiven. Verben wurden von beiden Taggern ab und zu mit Nomen und Eigennamen verwechselt. Neben der Unterscheidung von Nomen und Eigennamen, schien der NTLK-Tagger vermehrt insbesondere falsche Verben-Tags zu vergeben.

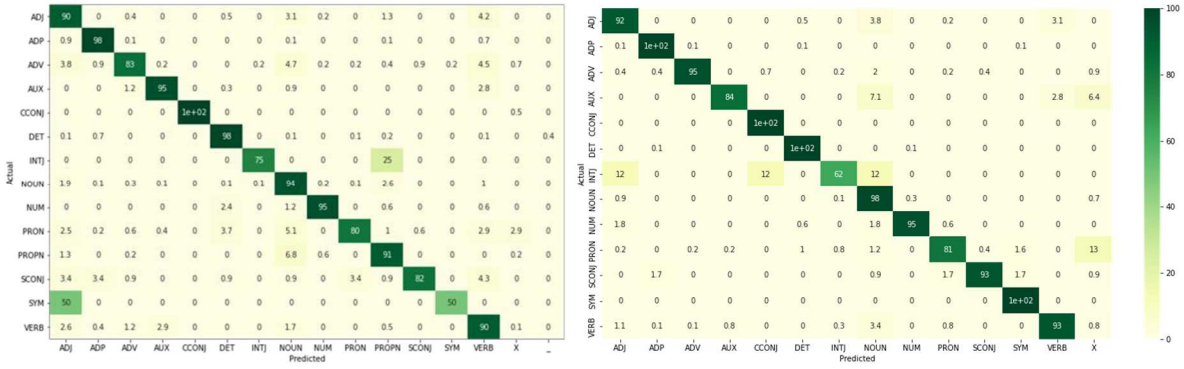

Abb. 41: Vergleich NLTK - DistilBERT Französisch (eigene Grafik)

Auf Englisch erreichte der mit den Universal Dependencies-Daten neu trainierte NLTK-Perceptron-POS-Tagger eine Wortebene-Accuracy von 93%, was nur um einen Prozentpunkt tiefer als der DistilBERT-Tagger ist. Auch auf Satzebene war der Unterschied zum

DistilBERT-Tagger nicht auffallend. Wenngleich der NLTK-Tagger mit 44% eine um 9 Prozentpunkte niedrigere Accuracy aufwies, so überragte sein Ergebnis doch das der anderen beiden getesteten Sprachen in NLTK.

Der NTLK-Tagger schien für Englisch Eigennamen besser als der DistilBERT-Tagger zuordnen zu können. Auch das korrekte Taggen von Hilfsverben gelang ihm besser. Wenngleich er Fehler machte, so handelte es sich mehrheitlich um Verwechslungen mit anderen Verben. Zudem ordnete er seltener fälschlicherweise Wörter der Klasse der Interjektionen zu, als dies der DistilBERT-Tagger tat.

Bei Partikeln hingegen war der DistilBERT-Tagger klar besser. Der NTLK-Tagger teilte diese häufiger falsch zu und markierte sie einige Male auch mit «X – andere». Auch Verben konnten vom DistilBERT-Modell etwas besser zugeteilt werden, während diese durch NLTK häufig in die falsche Klasse «NOUN» fielen.

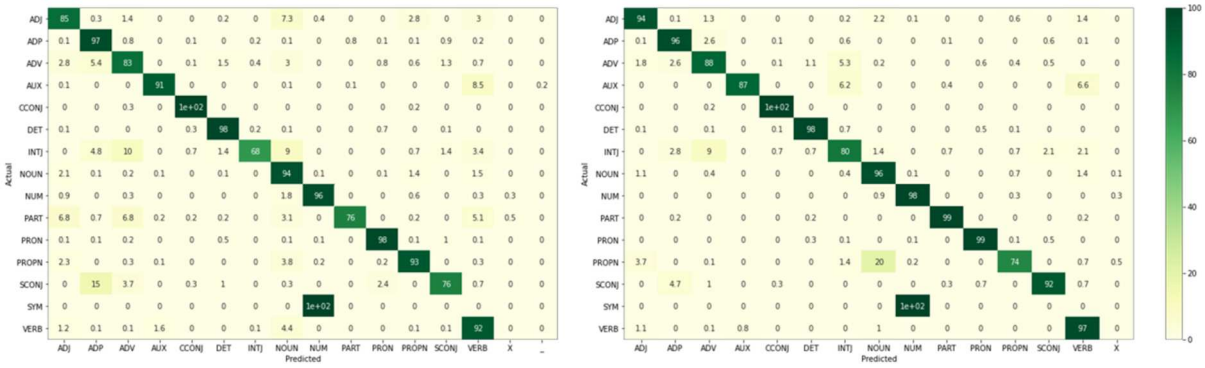

Abb. 42: Vergleich NLTK - DistilBERT Englisch (eigene Grafik)

Zusammengefasst kann gesagt werden, dass der NLTK-Tagger nur wenig schlechter als der DistilBERT-Tagger ist, dies sich aber auf die Genauigkeit auf Satzebene auswirkt, die sich dadurch massiv verschlechtert. Eigennamen scheint der NTLK-Tagger jedoch besser erkennen zu können als der DistilBERT-Tagger. Allenfalls könnte dies daran liegen, dass die Gross-/Kleinschreibung stärker berücksichtigt wird als die Position im Satz.

5.2.3 SpaCy

Auf Deutsch erreichte der SpaCy-Tagger auf dem Testdatensatz die zweithöchsten Accuracy-Werte und war auf Wortebene nur um einen Prozentpunkt niedriger in der Accuracy als der DistilBERT-Tagger. Auch auf Satzebene erreichte der SpaCy-Tagger den zweithöchsten Accuracy-Wert (37%) nach dem DistilBERT-Tagger (42%).

Eigennamen, Pronomen und Adjektive konnte der SpaCy-Tagger im Gegensatz zum DistilBERT-Tagger schlechter erkennen. Noch etwas häufiger als der DistilBERT-Tagger vergab er Eigennamen den Tag «NOUN». Umgekehrt klassierte der DistilBERT-Tagger

häufiger Nomen als Eigennamen. Auch bei den Klassen der Adjektive und Adverbien machten die beiden Tagger die falsche Zuordnung gegenteilig. DistilBERT klassierte Adverbien häufig fälschlicherweise als Adjektiv, SpaCy Adjektive als Adverbien.

Obschon dieser Fehler auch dem DistilBERT-Tagger einige Male unterlief, so ordnete der SpaCy-Tagger Pronomen häufiger der Klasse der Artikel zu. Auffallend ist zudem, dass SpaCy kaum Wörter als «X – andere» taggte. Dies lag aber vermutlich an den Daten, mit denen dieser Tagger trainiert worden war.

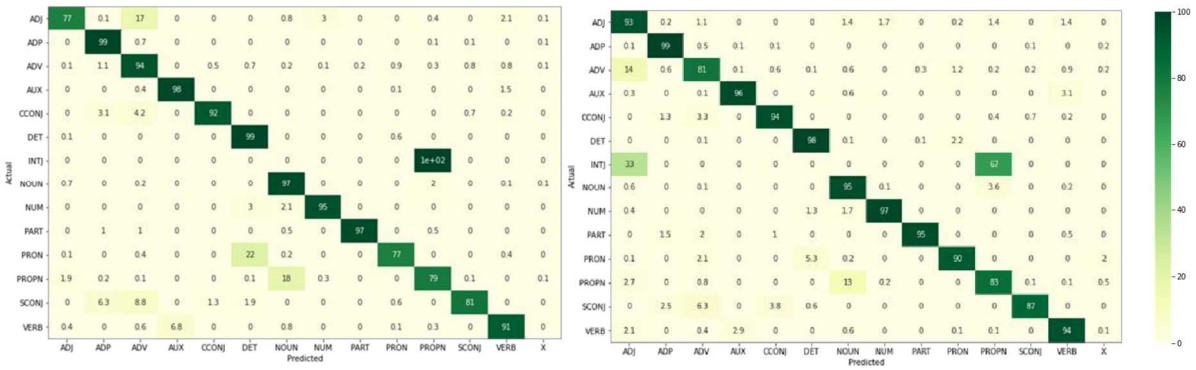

Abb. 43: Vergleich SpaCy - DistilBERT Deutsch (eigene Grafik)

Im Gegensatz zum Deutschen landete der französische SpaCy-Tagger auf der letzten Position, was seine Accuracy-Werte angeht. Auf Wortebene waren diese fast 10 Prozentpunkte niedriger (86% vs. 95%) und auf Satzebene gar 32 Prozentpunkte (14% vs. 46%).

Während viele Wörter fälschlicherweise als Nomen getaggt wurden, wurden die Nomen selbst oftmals mit anderen Klassen verwechselt, allen voran mit Adjektiven und Verben. Hilfsverben hingegen konnte der SpaCy-Tagger besser zuordnen als Distil-BERT. Gab es eine Verwechslung, so handelte es sich meist um die Klasse «VERB», ein weniger gravierender Fehler als bei DistilBERT, der diese Wortart am häufigsten fälschlicherweise der Klasse «NOUN» zuordnete. Die Verwechslung von Adjektiven und Verben mit Nomen passierte auch dem DistilBERT-Tagger, allerdings seltener als beim SpaCy-Tagger und die korrekte Zuordnung war höher.

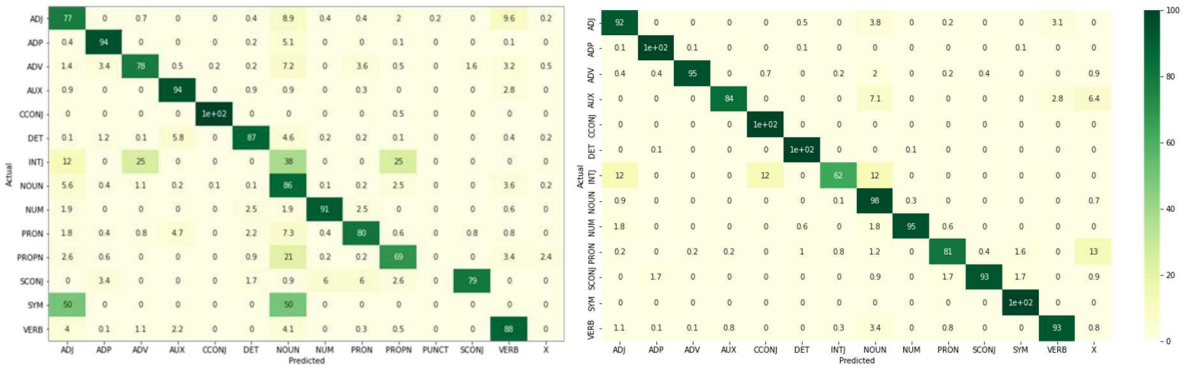

Abb. 44: Vergleich SpaCy - DistilBERT Französisch (eigene Grafik)

Auf dem englischen Testdatensatz erreichte der SpaCy-Tagger auf Wortebene eine Accuracy von 92% und war somit um zwei Prozentpunkte niedriger als der DistilBERT-Tagger. Auf Satzebene erhöhte sich die Error Rate des SpaCy-Taggers um elf Prozentpunkte auf 58%.

Während der DistilBERT-Tagger Schwierigkeiten hatte, Eigennamen nicht als Nomen zu kennzeichnen, schien dies für den SpaCy-Tagger eine viel kleinere Hürde darzustellen. Hilfsverben wurden zwar von SpaCy häufiger fälschlicherweise der Klasse «VERB» zugeordnet, jedoch fast gar nicht anderen Klassen.

Eigennamen konnten insgesamt besser zugeordnet werden, insbesondere gab es kaum Fehler, die nicht auf Verwechslungen mit der allgemeinen Nomenklasse beruhten. Auch Artikel wurden in fast allen Fällen korrekt getaggt, während dies beim Distil-BERT-Tagger immerhin bei 2% der im Datensatz vorhandenen Artikel nicht der Fall war.

Während nebenordnende Konjunktionen vom DistilBERT POS-Tagger relativ problemlos zugewiesen werden konnten, schaffte der SpaCy-Tagger dies nur in der Hälfte aller entsprechenden Wörter. Das gleiche kann bei Pronomen beobachtet werden, SpaCy klassierte diese oft als Determinative.

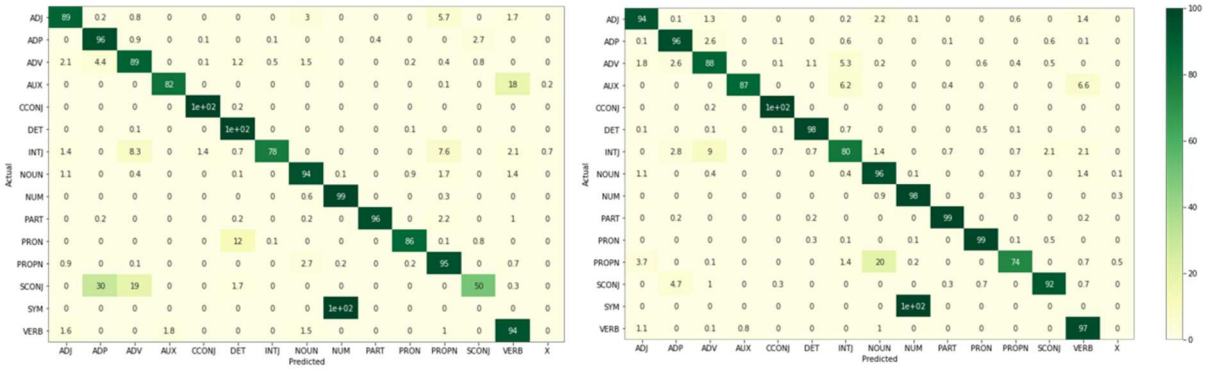

Abb. 45: Vergleich SpaCy - DistilBERT Englisch (eigene Grafik)

Zusammenfassend konnte der SpaCy-Tagger sich gegenüber dem DistilBERT-Tagger in einigen Wortarten zwar durchsetzen, die Gesamt-Accuracy blieb über alle Sprachen jedoch niedriger, insbesondere auf Satzebene.

5.3 Qualitative Evaluation

In der quantitativen Evaluation wurden die F1-Werte der einzelnen POS-Tags zwischen dem DistilBERT-Tagger und den Vergleichstaggern verglichen. Dabei zeigte sich, dass

der DistilBERT-Tagger über alle Sprachen hinweg bei der Mehrheit der Tags bessere Ergebnisse erzielen konnte.

Die Klassifikations-Reports mit den F1-Werten pro Wortart und Sprache, auf denen sich die Auswertung basiert, sind in den Anhängen 5-7 zu finden.

Der Lesbarkeit halber wurden in den nachfolgenden Unterkapiteln den Sätzen aus dem Testdatensatz Satzzeichen hinzugefügt. Da diese in der Vorverarbeitung entfernt wurden, hat das Modell diese jedoch nicht gelernt und auch der Testdatensatz wurde vom Algorithmus entsprechend ohne Satzzeichen getaggt. Rechtschreibfehler wurden nicht korrigiert und entsprechen den originalen Daten.

5.3.1 Deutsch

Die Tabelle zeigt einen Vergleich der verschiedenen F1-Werte über alle Wortarten der verschiedenen Tagger. Der F1-Score des DistilBERT-Taggers wurde dabei mit dem Wert des Taggers verglichen, der den höchsten Score hatte. Dieser Benchmark-Tagger ist in Klammer angegeben. Das fette Wort der entsprechenden Wortart wurde vom besseren Tagger korrekt, vom schlechteren falsch getaggt. Auffallend ist, dass der DistilBERT-Tagger in fast jeder Wortart bessere Werte erreichte, die Differenz zum nächstbesseren Tagger jedoch sehr klein ist. Nur Zahlwörter und Partikel wurden von einem anderen Tagger besser zugeordnet.

Wortart	F1-Score DistilBERT im Vergleich	Beispielsatz
ADJ	+/- 0.00 (SpaCy)	Ansonsten gemütliche Bar mit *innenstadtüblichen* Preisen und *spätabendlicher* Küche.
ADP	+ 0.01 (NLTK, SpaCy)	Aber **über** die Freundlichkeit, Zuverlässigkeit und Komptenz des gesamten Team kann man nur eines Sagen: PERFEKTION.
ADV	+ 0.02 (SpaCy)	Ausserdem gehört Herr Lansky zu den Anwälten, die gerade **herraus** sagen, wie die Chancen stehen, den Fall zu gewinnen oder nicht.
AUX	+ 0.02 (SpaCy)	Dennoch **ist** es immer sehr ordentlich; man bekommt alles, was man möchte zu einer sehr guten Qualität.
CCONJ	+/- 0.00 (NLTK, SpaCy)	Das ist leider keine Kundenorientierung *und* deshalb gibt es einen Kaktus für die Angestellten des Restaurants.
DET	+ 0.01 (NLTK)	**Den** in diesem Land vorhandenen Unternehmergeist lobte der Minister als eine milliardenschwere Infrastruktur.
NOUN	+/- 0.00 (SpaCy, Stanford)	Alles in *Allem* ein super Erlebnis.
NUM	- 0.01 (Stanford)	Ein umfassender Ausbau der Verkehrswege in der Bundesrepublik bis zum Jahre 2005 würde weit mehr als eine **Billion** Mark verschlingen.
PART	- 0.02 (SpaCy)	**Nein**, man muss das Problem stärker thematisieren.
PRON	+ 0.01 (NLTK)	Welcher Kollege hat **Ihnen** denn 99 gesagt?
PROPN	+ 0.01 (SpaCy, Stanford)	Parteitag der **Christdemokraten**, in Dillingen wählte am Samstag den 40jährigen Fraktionsvorsitzenden Peter Müller zum neuen Landeschef.
SCONJ	+0.04 (NLTK)	Schade, **das** solche Daten nicht geprüft werden.
VERB	+ 0.02 (SpaCy)	Aussage des Verkäufers, in 15 minuten **freigeschalten** sein.

Tabelle 8: Vergleich F1-Score pro Wortart Deutsch (eigene Tabelle)

Der Distilbert-Tagger verwechselte auf Deutsch neben Eigennamen mit Nomen häufig Determinative mit Pronomen. Nachfolgend sind einige Beispiele aus dem Testdatensatz aufgeführt:

- US-Präsident William Jefferson Clinton, **der** wie so viele die tiefe rauhe Stimme Rabins vermißte, schien gut informiert über die Lage in Israel.
- Der Franzose, **der** über seine Firma «Bernard Tapie Finance» Mehrheitsgesellschafter des nordbayerischen Unternehmens ist, ging auf die angeblich mit einer Milliarde Mark dotierte Offerte nicht ein.
- Ist wirklich zu empfehlen und ich wundere mich immer wider, dass nach sovielen Jahren **der** laden so wenig Bewertungen hat, deshalb habe ich mir jetzt ein Konto eingerichtet, um auch meine Meinung schreiben zu können.

Auch Adverbien und Adjektive wurden häufig verwechselt, wie die folgenden Beispiele zeigen:

- Ist wirklich zu empfehlen und ich wundere mich immer **wider**, dass nach sovielen Jahren der laden so wenig Bewertungen hat, deshalb habe ich mir jetzt ein Konto eingerichtet, um auch meine Meinung schreiben zu können.
- Bei Kontaktanfragen wird man bei Problemen **wochenlang** ignoriert.

Der Satz mit den meisten Fehlern (Total acht), war der folgende:

- **Diese** (PRON/**DET**) Mischung aus schlechten und etwas besseren Motiven zu einer europäischen Position **abzüglich** (ADJ/**ADP**) der **offenkundig** (ADJ/**ADV**) **nachbarschaftlich** (ADJ/**ADV**) befangenen Skandinavier **zu** (ADP/**PART**) **adeln** (NOUN/**VERB**) hieß **nun** (PART/**ADV**) **freilich** (ADJ/**ADV**) die Rechnung auf stoffelige Weise ohne den Wirt zu machen.

  (der erste POS-Tag ist der von DistilBERT vergebene, der fett gedruckte, der tatsächliche Tag)

Insgesamt 13 Sätze aus dem Testdatensatz wurden von allen POS-Taggern vollständig korrekt, vom DistilBERT jedoch falsch getaggt. Es handelt sich dabei in drei Fällen um ein Wort, das der DistilBERT-Tagger nicht zuordnen konnte und mit «X – andere» getaggt hat. Bei fünf Wörtern hat er Nomen und Eigennamen gegenseitig verwechselt. Zwei Wörtern wurde zudem die Klasse «NOUN» zugeordnet, während es sich in Wirklichkeit um Hilfsverben gehandelt hätte.

	Satz	Korrekte Wortarten	DistilBERT
0	Aber mal abwarten was sich in näherer Zukunft abspielt	- CCONJ - ADV - VERB - PRON - PRON - ADP - ADJ - NOUN - VERB	- CCONJ - ADV - VERB - X - PRON - ADP - ADJ - NOUN - VERB
1	Anders kann ich es nicht ausdrücken	- ADV - AUX - PRON - PRON - PART - VERB	- ADV - AUX - PRON - X - PART - VERB
2	Das ist seit nunmehr 20 Stunden noch nicht passiert	- PRON - AUX - ADP - ADV - NUM - NOUN - ADV - PART - VERB	- PRON - AUX - ADP - ADV - NUM - NOUN - ADV - PART - ADJ
3	Es fing mit einer Wartezeit von ca 20 Minuten an zu welcher mir jedoch ein Kaffee angeboten wurde	- PRON - VERB - ADP - DET - NOUN - ADP - ADV - NUM - NOUN - ADP - ADP - PRON - PRON - ADV - DET - NOUN - VERB - AUX	- PRON - VERB - ADP - DET - NOUN - ADP - ADV - NUM - NOUN - X - ADP - PRON - PRON - ADV - DET - NOUN - VERB - AUX
4	Ich kann Frau Lewandowski nur wärmstens empfehlen	- PRON - AUX - NOUN - PROPN - ADV - ADV - VERB	- PRON - AUX - NOUN - PROPN - ADV - PROPN - VERB
5	Die Bundespolizei erhielt Informationen denen zufolge das Trio dabeigewesen sein könnte eine Bombe zu bauen	- DET - NOUN - VERB - NOUN - PRON - ADP - DET - NOUN - VERB - AUX - AUX - DET - NOUN - PART - VERB	- DET - NOUN - VERB - NOUN - PRON - ADP - DET - NOUN - ADJ - NOUN - NOUN - DET - NOUN - PART - VERB
6	The Independent hatte berichtet Irving habe die Tagebücher den russischen Archiven für eine sechsstellige Summe abgekauft	- PROPN - PROPN - AUX - VERB - PROPN - AUX - DET - NOUN - DET - ADJ - NOUN - ADP - DET - ADJ - NOUN - VERB	- PROPN - ADJ - AUX - VERB - PROPN - AUX - DET - NOUN - DET - ADJ - NOUN - ADP - DET - ADJ - NOUN - VERB
7	Rita Neubauer Havanna fürchtet noch Schlimmeres	- PROPN - PROPN - PROPN - VERB - ADV - NOUN	- PROPN - NOUN - NOUN - VERB - ADV - PROPN
8	Wir sind die grünen Lotsen auf einem Containerschiff	- PRON - AUX - DET - ADJ - NOUN - ADP - DET - NOUN	- PRON - VERB - DET - ADJ - NOUN - ADP - DET - NOUN
9	Nazli Top erinnert sich	- PROPN - PROPN - VERB - PRON	- PROPN - ADJ - VERB - PRON
10	Auch die Regierung in Estland wurde wegen einer Ausgrenzung der starken russischen Minderheit kritisiert	- ADV - DET - NOUN - ADP - PROPN - AUX - ADP - DET - NOUN - DET - ADJ - ADJ - NOUN - VERB	- ADV - DET - NOUN - ADP - PROPN - AUX - ADP - DET - NOUN - DET - ADJ - NOUN - NOUN - VERB
11	Sie habe die Tür schnell wieder zugeschlagen sagte Chrétien	- PRON - AUX - DET - NOUN - ADV - ADV - VERB - VERB - PROPN	- PRON - AUX - DET - NOUN - ADJ - ADV - VERB - VERB - PROPN
12	Von Ulrike Füssel	- ADP - PROPN - PROPN	- ADP - PROPN - NOUN

Abb. 46: Deutsch - Sätze, die von allen Taggern, ausser dem DistilBERT-Tagger, korrekt zugeordnet wurden (eigene Grafik)

Bei 86 Sätzen des Testdatensatzes konnte der DistilBERT POS-Tagger als einziger Tagger den ganzen Satz korrekt taggen. Häufig handelte es sich dabei um Verwechslungen der anderen Tagger zwischen Verben und Hilfsverben. Aber auch Verwechslungen zwischen Nomen und Eigennamen waren dabei sowie zwischen Adjektiven und Adverbien.

	Satz	Korrekte Wortarten	Stanford	NLTK	SpaCy
0	Welcher Kollege hat Ihnen denn 99 gesagt	- PRON - NOUN - AUX - PRON - ADV - NUM - VERB	- PRON - NOUN - VERB - PRON - ADV - NUM - VERB	- PRON - NOUN - VERB - NOUN - CCONJ - NUM - VERB	- DET - NOUN - AUX - PRON - ADV - NUM - VERB
1	Ansonsten war alles bestens	- ADV - AUX - PRON - ADJ	- ADV - AUX - PRON - ADV	- ADV - AUX - PRON - ADV	- ADV - AUX - PRON - ADV
2	Aussage des Verkäufers in 15 minuten freigeschalten sein	- NOUN - DET - NOUN - ADP - NUM - NOUN - VERB - AUX	- NOUN - DET - NOUN - ADP - NUM - NOUN - VERB - DET	- NOUN - DET - NOUN - ADP - NUM - ADJ - ADJ - AUX	- NOUN - DET - NOUN - ADP - NUM - NOUN - NOUN - AUX
3	Bester Kaffee im in dem Veedel sowieso	- ADJ - NOUN - _ - ADP - DET - PROPN - ADV	- PROPN - NOUN - PROPN - ADP - DET - NOUN - ADV	- ADJ - NOUN - _ - ADP - DET - NOUN - ADV	- NOUN - NOUN - ADP - DET - NOUN - ADV
4	Bis ich bei Uwe war seit dem gehe ich nur noch zu ihm	- SCONJ - PRON - ADP - PROPN - VERB - ADP - DET - VERB - PRON - ADV - ADV - ADP - PRON	- ADP - PRON - ADP - PROPN - AUX - ADP - DET - VERB - PRON - ADV - ADV - ADP - PRON	- ADP - PRON - ADP - PROPN - AUX - ADP - DET - VERB - PRON - ADV - ADV - ADP - PRON	- SCONJ - PRON - ADP - PROPN - AUX - ADP - PRON - VERB - PRON - ADV - ADV - ADP - PRON
5	Daraufhin hat er zu anderen Gästen sich beklagt das wir kein verständnis haben und uns zur zu der Zahlung warten lassen	- ADV - AUX - PRON - ADP - ADJ - NOUN - PRON - VERB - SCONJ - PRON - PRON - NOUN - VERB - CCONJ - PRON - _ - ADP - DET - NOUN - VERB - VERB	- ADV - AUX - PRON - ADP - ADJ - NOUN - PRON - VERB - PRON - PRON - PRON - NOUN - VERB - CCONJ - PRON - ADV - ADP - DET - NOUN - VERB - VERB	- ADV - AUX - PRON - ADP - ADJ - NOUN - PRON - VERB - DET - PRON - PRON - VERB - AUX - CCONJ - PRON - _ - ADP - DET - NOUN - VERB - VERB	- ADV - AUX - PRON - ADP - ADJ - NOUN - PRON - VERB - DET - PRON - DET - NOUN - VERB - CCONJ - PRON - ADP - DET - NOUN - VERB - VERB
6	Darüberhinaus ist das Personal sehr freundlich und zuvorkommend und jeder Gast wird persönlich vom von dem Chef begrüßt	- ADV - AUX - DET - NOUN - ADV - ADJ - CCONJ - ADJ - CCONJ - PRON - NOUN - AUX - ADV - _ - ADP - DET - NOUN - VERB	- ADV - AUX - DET - NOUN - ADV - ADJ - CCONJ - ADJ - CCONJ - PRON - NOUN - AUX - ADV - ADV - ADP - DET - NOUN - VERB	- ADV - AUX - DET - NOUN - ADV - ADJ - CCONJ - ADJ - CCONJ - PRON - NOUN - AUX - ADJ - _ - ADP - DET - NOUN - VERB	- ADV - AUX - DET - NOUN - ADV - ADV - CCONJ - ADV - CCONJ - DET - NOUN - AUX - ADV - ADP - DET - NOUN - VERB
7	Das Lokal ist sauber hat einen gemütlichen Raucherraum und wird gut besucht	- DET - NOUN - AUX - ADJ - VERB - DET - ADJ - NOUN - CCONJ - AUX - ADV - VERB	- DET - NOUN - AUX - ADJ - AUX - DET - ADJ - NOUN - CCONJ - AUX - ADV - VERB	- DET - NOUN - AUX - ADJ - VERB - DET - ADJ - NOUN - CCONJ - AUX - ADJ - VERB	- DET - NOUN - AUX - ADV - VERB - DET - ADJ - NOUN - CCONJ - AUX - ADV - VERB
8	Der Kellner war sehr nett und die Atmosphäre ruhig und entspannt	- DET - NOUN - AUX - ADV - ADJ - CCONJ - DET - NOUN - ADJ - CCONJ - ADJ	- DET - NOUN - AUX - ADV - ADJ - CCONJ - DET - NOUN - ADJ - CCONJ - VERB	- DET - NOUN - AUX - ADV - ADJ - CCONJ - DET - NOUN - ADJ - CCONJ - VERB	- DET - NOUN - AUX - ADV - ADV - CCONJ - DET - NOUN - ADV - CCONJ - VERB
9	Die Küche ist gut wenn nicht der Koch oder die Köche manchmal nicht immer auf dem Posten sind	- DET - NOUN - AUX - ADJ - SCONJ - PART - DET - NOUN - CCONJ - DET - NOUN - ADV - PART - ADV - ADP - DET - NOUN - VERB	- DET - NOUN - AUX - ADJ - SCONJ - PART - DET - NOUN - CCONJ - DET - NOUN - ADV - PART - ADV - ADP - DET - NOUN - AUX	- DET - NOUN - AUX - ADJ - SCONJ - PART - DET - NOUN - CCONJ - DET - NOUN - ADV - PART - ADV - ADP - DET - NOUN - AUX	- DET - NOUN - AUX - ADV - SCONJ - PART - DET - PROPN - CCONJ - DET - NOUN - ADV - PART - ADV - ADP - DET - NOUN - AUX

Abb. 47: Deutsch - Ausschnitt der Sätze, die nur vom DistilBERT-Tagger vollständig korrekt getaggt wurden (eigene Grafik)

Die Analyse von Sätzen mit Homographen unterschiedlicher Wortarten (siehe Anhang 2) zeigte, dass alle Tagger gut damit umgehen konnten und die Position im Satz stärker berücksichtigen als das Wort selbst. Ein Vergleich mit ausschliesslich Kleinschreibung zeigte zudem, dass alle Modelle im Deutschen Nomen mit Grossschreibung assoziieren. Dies kann in Texten mit Rechtschreibefehlern einen negativen Einfluss haben.

Zusammenfassend kann gesagt werden, dass der DistilBERT-Tagger auf Deutsch in den einzelnen Wortarten vergleichbare Werte wie bestehenden Taggern erzielte. Die Verwechslungen waren in der Mehrheit zwei sich sehr ähnliche Wortarten, die auch von Menschen nicht immer eindeutig zugeordnet werden können. Auf Satzebene konnte DistilBERT die Genauigkeit bisheriger Taggern verbessern.

5.3.2 Französisch

Im Vergleich des F1-Scores über die verschiedenen Wortarten kann gesehen werden, dass der DistilBERT-Tagger auf Französisch in jeder Kategorie ausser den Hilfsverben bessere Werte erzielte. Besonders gross war die Differenz bei den Adverbien und den unterordnenden Konjunktionen.

Wortart	F1-Score DistilBERT im Vergleich	Beispielsatz
ADJ	+ 0.03 (Stanford)	Vraiment très bon resto avec un accueil aussi chaleureux qu'**enthousiaste**, Lesplats sont copieux et cuisinés avec de bons produits.
ADP	+ 0.01 (Stanford)	Il fut dédié en 493 **av**. JC par Spurius Cassius.
ADV	+ 0.09 (Stanford, NLTK)	Vous n'avez **qu'à** apporter vos valises, tout est pensé pour un séjour des plus agréable.
AUX	- 0.02 (Stanford)	En 1872, quand les Georgiens ont repris les commandes du gouvernement, Barnett, qui **avait été** réélu par ce point, a rapporté le sceau.
CCONJ	+ 0.01 (SpaCy, NLTK)	Selon certaines sources c'est igaud qui a rédigé les ouvrages de ce dernier, soit en latin **soit** en français.
DET	+ 0.02 (NLTK, Stanford)	Les places d'accueil s'adressent en priorité aux parents qui travaillent et n'ont pas depossibilité **de** garde pour leur enfant.
NOUN	+ 0.01 (NLTK, Stanford)	Pas de sèche-cheveux ni de prise **rasoir** dans la salle de bains.
NUM	+ 0.01 (NLTK, Stanford)	Par la suite, le château fut démantelé durant la Guerre de **Trente** Ans.
PRON	+ 0.03 (NLTK)	Il se compose de deux ensembles de lamelles articulés et fixées les **unes** aux autres par des rivets.
PROPN	+/- 0.00 (NLTK)	La jeune *Marian*, contraction de *Mary Anne*, se révèle d'un caractère obstiné et ombrageux.
SCONJ	+0.10 (NLTK)	C'est inadmissible de voler les gens ainsi, ce n'est même pas **comme s**'ils avaient utilisé beaucoup de matériel ; un robinet et quelques fils d'étanchéité.
VERB	+ 0.02 (Stanford)	Le gaz développe ses applications domestiques à la cuisine pour **cuire** et **chauffer** l'eau ou comme mode de chauffage central

Tabelle 9: Vergleich F1-Score pro Wortart Französisch (eigene Tabelle)

Im Französischen hatte der DistilBERT-Tagger Schwierigkeiten, Pronomen nicht als «X – andere» zu klassieren. Folgende drei Beispiele illustrieren dies:

- **J'y** ai déjeuné avec un collègue le vendredi 11 septembre 2009 et nous avons été déçus
- Lorsqu'elle **m*'**a examiné elle a appuyé comme une malade à l'endroit de l'opération et en me voyant crier elle **en** a finalement déduit qu'il fallait que je retourne à l'hopital
- **dieses Pronomen wurde nicht als «X – andere», sondern als Nomen getaggt*
- Située à proximité de la route menant au parc national de Zion, **il s**'agit d'une des villes fantômes les plus connues du pays

Des Weiteren wurden Adjektive öfters fälschlicherweise als Nomen getaggt. Folgend sind zwei Beispiele:

- Créé en 1964 par Pierre Billon Sifca est alors le premier groupe **privé** Ivoirien.
- Asterotrygon avait une forme typique de raie avec un disque **plat**, arrondi formé de la tête et des nageoires pectorales et une longue queue étroite avec un puissant dard.

Die meisten Fehler (13 insgesamt) hatte der folgende Satz:

- L'**Oscar** (NOUN/**PROPN**) du de le meilleur mixage de **son** (DET/**NOUN**) (**Academy** (PROPN/**X**) **Award** (PROPN/**X**) for **Best** (PROPN/**X**) **Sound** (NOUN/**X**) **Mixing** (NOUN/**X**)) est une récompense cinématographique américaine décernée chaque année, depuis 1930 (le cinéma parlant n'étant pas encore assez répandu avant **cette** (VERB/**ADJ**) date) par l'**Academy** (PROPN/**X**) of **Motion** (NOUN/**X**) **Picture** (PROPN/**X**) **Arts** (PROPN/**X**) **and** (CCONJ/**X**) **Sciences** (AMPAS) laquelle décerne également tous les autres Oscars.

  (der erste POS-Tag ist der von DistilBERT vergebene, der fett gedruckte, der tatsächliche Tag)

Bei diesem Satz muss allerdings angemerkt werden, dass Fremdwörter zur Zeit in den französischen Universal Dependencies noch als «X – andere» getaggt sind (Universal Dependencies, 2021t). Von diesem Gesichtspunkt her hat DistilBERT diese Wörter korrekt als Eigennamen getaggt. Er konnte zudem sogar das englische «and» als nebenordnende Konjunktion erkennen.

Die Wörter von insgesamt zehn Sätzen im französischen Testdatensatz konnte der DistilBERT-Tagger als einziger nicht alle richtig klassieren. Am häufigsten ordnete er in diesen Fällen andere Wortarten der Nomen-Klasse zu (sechs Fälle). Zwei Mal konnte er ein Wort nicht zuordnen und taggte es als «X – andere». Bei zwei Wörtern verwechselte er Verben mit Adjektiven und bei zweien Hilfsverben, bzw. Verben mit Pronomen. Ein Verb klassierte er als Hilfsverb.

	Satz	Korrekte Wortarten	DistilBERT
0	Le gardien algérien a lui aussi laissé filer le ballon dans ses buts	- DET - NOUN - ADJ - AUX - PRON - ADV - VERB - VERB - DET - NOUN - ADP - DET - NOUN	- DET - NOUN - ADJ - AUX - PRON - ADV - VERB - NOUN - DET - NOUN - ADP - DET - NOUN
1	Léaniс Olivier est un coureur cycliste français né le 17 novembre 1977 à Meulan	- PROPN - PROPN - AUX - DET - NOUN - ADJ - ADJ - VERB - DET - NUM - NOUN - NUM - ADP - PROPN	- PROPN - PROPN - AUX - DET - NOUN - NOUN - ADJ - VERB - DET - NUM - NOUN - NUM - ADP - PROPN
2	Un contrat de trois ans et un salaire de 45 millions d euros par an auraient été évoqués	- DET - NOUN - ADP - NUM - NOUN - CCONJ - DET - NOUN - ADP - NUM - NOUN - ADP - NOUN - ADP - NOUN - AUX - AUX - VERB	- DET - NOUN - ADP - NUM - NOUN - CCONJ - DET - NOUN - ADP - NUM - NOUN - ADP - NOUN - ADP - NOUN - PRON - NOUN - VERB
3	Mohamed Zran sait de quoi il parle	- PROPN - PROPN - VERB - ADP - PRON - PRON - VERB	- PROPN - PROPN - VERB - ADP - PRON - X - VERB
4	Il y a donc le monde de la réalité comprenant les nobles et les artisans et le monde de la magie	- PRON - PRON - VERB - ADV - DET - NOUN - ADP - DET - NOUN - VERB - DET - NOUN - CCONJ - DET - NOUN - CCONJ - DET - NOUN - ADP - DET - NOUN	- PRON - X - VERB - ADV - DET - NOUN - ADP - DET - NOUN - VERB - DET - NOUN - CCONJ - DET - NOUN - CCONJ - DET - NOUN - ADP - DET - NOUN
5	La tumeur peut devenir symptomatique rapidement mais il arrive qu elle reste asymptomatique malgré une taille énorme	- DET - NOUN - VERB - VERB - ADJ - ADV - CCONJ - PRON - VERB - SCONJ - PRON - VERB - ADJ - ADP - DET - NOUN - ADJ	- DET - NOUN - PRON - NOUN - ADJ - ADV - CCONJ - PRON - VERB - SCONJ - PRON - VERB - ADJ - ADP - DET - NOUN - ADJ
6	Contrairement à celui de Wonder Woman celui de Cassie est une arme pouvant projeter la lumière de Zeus	- ADV - ADP - PRON - ADP - PROPN - PROPN - PRON - ADP - PROPN - AUX - DET - NOUN - VERB - VERB - DET - NOUN - ADP - PROPN	- ADV - ADP - PRON - ADP - PROPN - PROPN - PRON - ADP - PROPN - AUX - DET - NOUN - NOUN - VERB - DET - NOUN - ADP - PROPN
7	Un nouvel album devrait voir le jour en 2008	- DET - ADJ - NOUN - VERB - VERB - DET - NOUN - ADP - NUM	- DET - ADJ - NOUN - ADJ - VERB - DET - NOUN - ADP - NUM
8	Environ 80000 vétérants bénéficient de ces mesures créant une masse de paysans et de propriétaires aigris par cette spoliation	- ADV - NUM - NOUN - VERB - ADP - DET - NOUN - VERB - DET - NOUN - ADP - NOUN - CCONJ - ADP - NOUN - VERB - ADP - DET - NOUN	- ADV - NUM - NOUN - VERB - ADP - DET - NOUN - VERB - DET - NOUN - ADP - NOUN - CCONJ - ADP - NOUN - ADJ - ADP - DET - NOUN
9	Le renforcement de la présence policière y serait pour beaucoup	- DET - NOUN - ADP - DET - NOUN - ADJ - PRON - VERB - ADP - ADV	- DET - NOUN - ADP - DET - NOUN - ADJ - PRON - AUX - ADP - ADV

Abb. 48: Französisch - Sätze, die von allen Taggern, ausser dem DistilBERT-Tagger, korrekt zugeordnet wurden (eigene Grafik)

304 Sätze des Testdatensatzes konnten nur vom DistilBERT vollständig korrekt getaggt werden. Die anderen Tagger hatten alle mindestens einen Fehler. Mehrere Male wurde von den anderen Taggern ein zusammengezogenes Wort nicht erkannt und mit einer Wortart anstelle «_» getaggt. Immer wieder wurden Wörter zudem – meist von NLTK und Stanford – fälschlicherweise als «X – andere» klassiert. Zudem wurden mehrere Male falsche Adjektive und Verben zugeordnet.

	Satz	Korrekte Wortarten	Stanford	NLTK	SpaCy
0	Les études durent six ans mais leur contenu diffère donc selon les Facultés	- DET - NOUN - VERB - NUM - NOUN - CCONJ - DET - NOUN - VERB - ADV - ADP - DET - NOUN	- DET - NOUN - VERB - NUM - NOUN - CCONJ - DET - NOUN - VERB - ADV - ADP - DET - PROPN	- DET - NOUN - VERB - NUM - NOUN - CCONJ - DET - NOUN - VERB - ADV - ADP - DET - PROPN	- DET - NOUN - VERB - NUM - NOUN - CCONJ - DET - NOUN - ADJ - ADV - ADP - DET - NOUN
1	L oasis de vie dans un milieu o règne l obscurité totale et une pression hydrostatique importante est riche et varié les chercheurs y découvrent de nouvelles espèces de bivalves de poissons de crustacés de poulpes dans des zones pensées jusqu alors désertiques	- DET - NOUN - ADP - NOUN - ADP - DET - NOUN - ADV - VERB - DET - NOUN - ADJ - CCONJ - DET - NOUN - ADJ - ADJ - AUX - ADJ - CCONJ - ADJ - DET - NOUN - PRON - VERB - DET - ADJ - NOUN - ADP - NOUN - ADP - NOUN - ADP - NOUN - ADP - DET - NOUN - VERB - ADP - ADV - ADJ	- DET - NOUN - ADP - NOUN - ADP - DET - NOUN - ADP - NOUN - DET - NOUN - ADJ - CCONJ - DET - NOUN - ADJ - ADJ - AUX - ADJ - CCONJ - ADJ - DET - NOUN - ADV - VERB - DET - ADJ - NOUN - ADP - NOUN - ADP - NOUN - ADP - NOUN - ADP - DET - NOUN - VERB - VERB - ADV - ADJ	- DET - NOUN - ADP - NOUN - ADP - DET - NOUN - ADJ - VERB - DET - NOUN - ADJ - CCONJ - DET - NOUN - ADJ - ADJ - AUX - ADJ - CCONJ - VERB - DET - NOUN - PRON - VERB - ADP - ADJ - NOUN - ADP - NOUN - ADP - NOUN - ADP - NOUN - ADP - DET - NOUN - ADJ - VERB - ADV - ADJ	- NOUN - NOUN - ADP - NOUN - ADP - DET - NOUN - AUX - AUX - AUX - NOUN - ADJ - CCONJ - DET - NOUN - ADJ - ADJ - AUX - ADJ - CCONJ - VERB - DET - NOUN - PRON - VERB - DET - ADJ - NOUN - ADP - ADJ - ADP - NOUN - ADP - NOUN - ADP - NOUN - ADP - DET - NOUN - VERB - ADJ - ADV - ADJ
2	On retrouve ici une touche très melvillienne et qui au à le lieu de souligner la thématique du de le livre y ajoute du sens	- PRON - VERB - ADV - DET - NOUN - ADV - ADJ - CCONJ - PRON - _ - ADP - DET - NOUN - ADP - VERB - DET - NOUN - _ - ADP - DET - NOUN - PRON - VERB - DET - NOUN	- PRON - VERB - ADV - DET - NOUN - ADV - ADJ - CCONJ - PRON - VERB - ADP - DET - NOUN - ADP - VERB - DET - ADJ - PROPN - ADP - DET - NOUN - ADV - VERB - DET - NOUN	- PRON - VERB - ADV - DET - NOUN - ADV - ADJ - CCONJ - PRON - _ - ADP - DET - NOUN - ADP - VERB - DET - NOUN - _ - ADP - DET - NOUN - PRON - VERB - _ - NOUN	- PRON - VERB - ADV - DET - NOUN - ADV - NOUN - CCONJ - PRON - ADP - DET - NOUN - ADP - VERB - DET - NOUN - ADP - DET - NOUN - PRON - VERB - ADP - NOUN
3	Comprenant six sommets dont un point culminant à 2001 mètres et une arrivée en altitude c est une étape typique de montagne	- VERB - NUM - NOUN - PRON - DET - NOUN - VERB - ADP - NUM - NOUN - CCONJ - DET - NOUN - ADP - NOUN - PRON - AUX - DET - NOUN - ADJ - ADP - NOUN	- VERB - NUM - NOUN - PRON - DET - NOUN - ADJ - ADP - NUM - NOUN - CCONJ - DET - NOUN - ADP - NOUN - X - AUX - DET - NOUN - ADJ - ADP - NOUN	- VERB - NUM - NOUN - PRON - DET - NOUN - ADJ - ADP - NUM - NOUN - CCONJ - DET - NOUN - ADP - NOUN - X - AUX - DET - NOUN - ADJ - ADP - NOUN	- VERB - NUM - NOUN - PRON - DET - NOUN - VERB - ADP - NUM - NOUN - CCONJ - DET - NOUN - ADP - NOUN - NOUN - AUX - DET - NOUN - ADJ - ADP - ADJ
4	Johnson s appuyait sur une unique forme de rhétorique et sa réfutation de l immatérialisme de George Berkeley est restée célèbre Berkeley affirmait que la matière n existait pas mais semblait seulement exister	- PROPN - PRON - VERB - ADP - DET - ADJ - NOUN - ADP - NOUN - CCONJ - DET - NOUN - ADP - DET - NOUN - ADP - PROPN - PROPN - AUX - VERB - ADJ - PROPN - VERB - SCONJ - DET - NOUN - ADV - VERB - ADV - CCONJ - VERB - ADV - VERB	- PROPN - X - VERB - ADP - DET - ADJ - NOUN - ADP - NOUN - CCONJ - DET - NOUN - ADP - DET - NOUN - ADP - PROPN - PROPN - AUX - VERB - ADJ - PROPN - VERB - SCONJ - DET - NOUN - ADJ - VERB - ADV - CCONJ - VERB - ADV - VERB	- PROPN - X - VERB - ADP - DET - ADJ - NOUN - ADP - NOUN - CCONJ - DET - NOUN - ADP - DET - NOUN - ADP - PROPN - PROPN - AUX - VERB - ADJ - PROPN - VERB - SCONJ - DET - NOUN - ADJ - VERB - ADV - CCONJ - VERB - ADV - VERB	- PROPN - AUX - VERB - ADP - DET - ADJ - NOUN - ADP - NOUN - CCONJ - DET - NOUN - ADP - NUM - NOUN - ADP - PROPN - PROPN - AUX - VERB - ADJ - PROPN - VERB - SCONJ - DET - NOUN - NOUN - VERB - ADV - CCONJ - VERB - ADV - VERB
5	L album de photos de police exposé au à le musée d Histoire politique de la Russie de SaintPétersbourg révèle le visage de Raspoutine défoncé par des coups et son corps avec quatre points d impacts de balles qui ont traversé le cur le cou et le cerveau	- DET - NOUN - ADP - NOUN - ADP - NOUN - VERB - _ - ADP - DET - NOUN - ADP - NOUN - ADJ - ADP - DET - PROPN - ADP - PROPN - VERB - DET - NOUN - ADP - PROPN - VERB - ADP - DET - NOUN - CCONJ - DET - NOUN - ADP - NUM - NOUN - ADP - NOUN - ADP - NOUN - PRON - AUX - VERB - DET - NOUN - DET - NOUN - CCONJ - DET - NOUN	- DET - NOUN - ADP - NOUN - ADP - NOUN - ADJ - X - ADP - DET - NOUN - ADP - NOUN - ADJ - ADP - DET - PROPN - ADP - PROPN - VERB - DET - NOUN - ADP - PROPN - VERB - ADP - DET - NOUN - CCONJ - DET - NOUN - ADP - NUM - NOUN - ADP - NOUN - ADP - NOUN - PRON - AUX - VERB - DET - NOUN - DET - NOUN - CCONJ - DET - NOUN	- DET - NOUN - ADP - NOUN - ADP - NOUN - VERB - _ - ADP - DET - NOUN - ADJ - NOUN - ADJ - ADP - DET - PROPN - ADP - PROPN - VERB - DET - NOUN - ADP - PROPN - VERB - ADP - DET - NOUN - CCONJ - DET - NOUN - ADP - NUM - NOUN - ADP - NOUN - ADP - NOUN - PRON - AUX - VERB - DET - NOUN - DET - NOUN - CCONJ - DET - NOUN	- NOUN - ADV - ADP - NOUN - ADP - NOUN - ADJ - ADP - DET - PROPN - NOUN - NOUN - ADJ - ADP - DET - PROPN - ADP - PROPN - VERB - DET - NOUN - ADP - NOUN - VERB - ADP - DET - NOUN - CCONJ - DET - NOUN - ADP - NUM - NOUN - NOUN - ADJ - ADP - NOUN - PRON - AUX - VERB - DET - NOUN - DET - NOUN - CCONJ - DET - NOUN

Abb. 49: Französisch - Ausschnitt der Sätze, die nur vom DistilBERT-Tagger vollständig korrekt getaggt wurden (eigene Grafik)

Wie die Analyse der Homographensätze zeigte, schienen auch auf Französisch alle Tagger weniger das einzelne Wort zu gewichten als seine Positionen und die vorherigen/nachfolgenden Wörter. Der DistilBERT-Tagger konnte zwar die meisten Wörter korrekt zuordnen (14 von 16 Homographen), doch die anderen Tagger waren mit 11/16 (Stanford), 12/16 (SpaCy), 12/16 (NLTK) sehr nahe an diesem Ergebnis. Somit scheinen auch für Französisch die Wörter an sich nicht die höchste Bedeutung zu haben. Bei einem Satz machten alle Tagger den gleichen Fehler: «Ces cuisiniers **excellent** (ADJ/**VERB**) à composer cet excellent plat.» Alle Tagger haben das konjugierte Verb «excellent» (sich auszeichnen/brillieren) fälschlicherweise als Adjektiv getaggt. Doch mit nur der Angleichung an die Mehrzahl (excellent**s**), bzw. der Einzahlsetzung des Nomens (Ce cuisinier) wäre das Wort tatsächlich ein Verb, auch wenn dann der Rest des Satzes

grammatikalisch nicht mehr korrekt wäre. Für Fremdsprachige wäre die Wortart dieses Wortes darum vermutlich ebenfalls nicht eindeutig.

Es zeigte sich insgesamt, dass der eigene Tagger auf Französisch über alle Wortarten bessere Ergebnisse als bestehende Tagger erzielte und so zu einer Erhöhung der Genauigkeit auf Satzebene führte. Insbesondere ordnete er fremdsprachige Wörter besser als die manuelle Annotation der Universal Dependencies zu.

5.3.3 Englisch

Im Vergleich mit den anderen Taggern klassierte der englische DistilBERT-POS-Tagger insbesondere unterordnende Konjunktionen besser. Interjektionen und Eigennamen hingegen konnte er weniger gut zuordnen.

Wortart	**F1-Score DistilBERT im Vergleich**	**Beispielsatz**
ADJ	+ 0.01 (Stanford)	Late in his life he published "**Analytic** Syntax" (1937) in which he presents his views on syntactic structure using an idiosyncraticshorthand notation.
ADP	+ 0.02 (NLTK, SpaCy)	She's swept past the whalelike oval of the public pool on the 202nd, **past** the sloping mandala of the Google of offices on the 164th
ADV	+/- 0.00 (SpaCy)	*Maybe* I should go with Miles.
AUX	- 0.01 (NLTK)	Thus, all estimates reported here **can be** considered nationally representative of the United States.
CCONJ	- 0.01 (SpaCy)	"This government is so consumed with control of information and secrecy, **yet** they don't seem to be able to get the fundamentals right", ewar said.
DET	- 0.01 (NLTK)	Neiafu offers **all** the usual amenities including banks, schools, tour companies, restaurants, cafes and bars, supermarkets, a market, and a hospital.
INTJ	- 0.29 (SpaCy)	But, but I remember **like** I went there with this person, it's kind of funny.
NOUN	- 0.03 (Stanford)	A barn **owl** is the best.

NUM	- 0.01 (SpaCy, Stanford)	Whoever wins election on November **5**, 2013, would replace outgoing governor Bob McDonnell.
PART	+ 0.02 (SpaCy)	Is your cigarette out, everybody'**s**?
PRON	+ 0.01 (NLTK)	Alright, so let's start off with the bed and the couch area.
PROPN	- 0.10 (NLTK)	They are located north of the Casamance **River** in the Jola Bluf area.
SCONJ	+0.13 (NLTK)	But ad says they're a sort of collection and too expensive **for** me to play with.
VERB	+ 0.04 (NLTK, SpaCy)	Nowadays most Eegimaa speakers from Mofvvi **live** outside their homeland, generally in urban areas like Ziguinchor and Dakar.

Tabelle 10: Vergleich F1-Score pro Wortart Englisch (eigene Tabelle)

Adverbien und Hilfsverben wurden vom englischen DistilBERT-POS-Tagger häufig mit Interjektionen verwechselt. Entsprechende Fehler mit Adverbien betrafen oftmals den Ausdruck «as well». Bei Falschzuordnungen von Hilfsverben handelte es sich fast ausschliesslich um die Phrase «can be» und ihre verschiedenen grammatikalischen Formen, die der DistilBERT-Tagger fälschlicherweise als Interjektion interpretierte. Folgend sind je zwei Beispiel aus dem Testdatensatz.

INTJ statt korrekt ADV:

- While many mousers are great pets **as well**, not all are
- These restrictions have played a key role in keeping COVID-19 out of New Zealand for **so** long and have given us time to better understand the disease and ramp up our preparations.

INTJ statt korrekt AUX:

- Thus, all estimates reported here **can be** considered nationally representative of the United States.
- Hopefully they'**ll be** here in about a week.

Gewisse Adverbien wurden zudem fälschlicherweise als Adpositionen getaggt, wie das folgende Beispiel zeigt. Teilweise handelt es sich dabei um Wörter, die in einem anderen Zusammenhang tatsächlich als Präpositionen genutzt werden (z.B. «down»).

- You're leaning on the railing waiting to ask Dery **about** a job, watching the glittering stream of mites that arc over half the sky, flying up to rewind their nano-springs in the stratospheric sunlight flying down to make Frankfurt run.

Die meisten Fehler (Total 19) hatte der folgende Satz:

- NASA held a ceremony commemorating the date outside the hangar, known as **Orbiter** (NOUN/**PROPN**) **Processing** (NOUN/**PROPN**) **Facility** (NOUN/**PROPN**) 1, for **Space** (NOUN/**PROPN**) **Shuttle** (NOUN/**PROPN**) Atlantis, which **is** (VERB/**AUX**) **being** (VERB/**AUX**) prepped for its final mission which will **be** (AUX/**INTJ**) STS 135, which will **be** (AUX/**INTJ**) the last **Space** (NOUN/**PROPN**) **Shuttle** (NOUN/**PROPN**) mission.

  (der erste POS-Tag ist der von DistilBERT vergebene, der fett gedruckte, der tatsächliche Tag)

Die meisten Fehler betrafen hier die Verwechslungen zwischen Nomen und Eigennamen, was nicht als gravierende Falschklassierung einzuordnen ist. Auch Hilfsverben, die fälschlicherweise als Verben klassiert wurden, sind vertreten. Zudem gibt es Beispiele der oben genannten Verwechslungen mit Interjektionen.

Im englischen Testdatensatz gab es fünfzig Sätze, deren Wortarten nur der DistilBERT-Tagger nicht vollständig korrekt zuordnen konnte. Der Stanford-Tagger wurde für diese Evaluation weggelassen, da ein direkter Vergleich wie erklärt, schwierig ist.

Es handelte sich bei diesen Fehlern um Wörter, die falsch als Adjektive oder Interjektionen getaggt wurden und tatsächlich beispielsweise Verben oder Nomen gewesen wären. Die meisten Fehler jedoch betrafen Eigennamen, die vom DistilBERT-POS-Tagger als «gewöhnliche» Nomen klassiert wurden.

	Satz	Korrekte Wortarten	DistilBERT
0	The following categories from the original questionnaire were collapsed into one category for the analysis race ancestry or national origin and shade of skin color	- DET - VERB - NOUN - ADP - DET - ADJ - NOUN - AUX - VERB - ADP - NUM - NOUN - ADP - DET - NOUN - NOUN - NOUN - CCONJ - ADJ - NOUN - CCONJ - NOUN - ADP - NOUN - NOUN	- DET - ADJ - NOUN - ADP - DET - ADJ - NOUN - AUX - VERB - ADP - NUM - NOUN - ADP - DET - NOUN - NOUN - NOUN - CCONJ - ADJ - NOUN - CCONJ - NOUN - ADP - NOUN - NOUN
1	Because a race variable is not available from the Wave 4 interviews we use the racial category reported by the respondent during the Wave 1 interview	- SCONJ - DET - NOUN - NOUN - AUX - PART - ADJ - ADP - DET - PROPN - NUM - NOUN - PRON - VERB - DET - ADJ - NOUN - VERB - ADP - DET - NOUN - ADP - DET - PROPN - NUM - NOUN	- SCONJ - DET - NOUN - ADJ - AUX - PART - ADJ - ADP - DET - PROPN - NUM - NOUN - PRON - VERB - DET - ADJ - NOUN - VERB - ADP - DET - NOUN - ADP - DET - PROPN - NUM - NOUN
2	Language names are written following the recommendation for the transcription of national languages of Senegal Decree 2005981	- NOUN - NOUN - AUX - VERB - VERB - DET - NOUN - ADP - DET - NOUN - ADP - ADJ - NOUN - ADP - PROPN - PROPN - NUM	- NOUN - NOUN - AUX - VERB - VERB - DET - NOUN - ADP - DET - NOUN - ADP - ADJ - NOUN - ADP - PROPN - NOUN - NUM
3	They are located north of the Casamance River in the Jola Bluf area	- PRON - AUX - VERB - ADV - ADP - DET - PROPN - PROPN - ADP - DET - PROPN - PROPN - NOUN	- PRON - AUX - VERB - ADV - ADP - DET - PROPN - NOUN - ADP - DET - PROPN - NOUN - NOUN
4	Antonn Dvok	- PROPN - PROPN	- PROPN - NOUN
6	In 1891 Dvok was appointed as a professor at the Prague Conservatory	- ADP - NUM - PROPN - AUX - VERB - ADP - DET - NOUN - ADP - DET - PROPN - PROPN	- ADP - NUM - PROPN - AUX - VERB - ADP - DET - NOUN - ADP - DET - PROPN - NOUN
7	He supported himself during his studies through part time work as a schoolteacher and as a shorthand reporter in the Danish parliament	- PRON - VERB - PRON - ADP - PRON - NOUN - ADP - NOUN - NOUN - NOUN - ADP - DET - NOUN - CCONJ - ADP - DET - NOUN - NOUN - ADP - DET - ADJ - NOUN	- PRON - VERB - PRON - ADP - PRON - NOUN - ADP - NOUN - NOUN - NOUN - ADP - DET - NOUN - CCONJ - ADP - DET - ADV - NOUN - ADP - DET - ADJ - NOUN

Abb. 50: Englisch - Sätze, die von allen Taggern, ausser dem DistilBERT-Tagger, korrekt zugeordnet wurden (eigene Grafik)

Insgesamt wurden 103 Sätze von NLTK und SpaCy falsch, aber vom DistilBERT-POS-Tagger korrekt getaggt. Auch für diese Filterung wurde der Stanford-Tagger nicht berücksichtigt. Im untenstehenden Tabellenausschnitt wurden die Resultate des Stanford-Taggers ausgegeben, allerdings mit den originalen PENN Treebank-Tags. Fehler wurden nur dann markiert, wenn es in den Universal POS-Tags und in den PENN Treebank-POS-Tags eine Unterscheidung der Klassen gibt (Einzahl und Mehrzahl von Nomen, die nur in den PENN Treebank unterschieden werden, wurden beispielsweise als gleichwertige Nomenklasse berücksichtigt, bei wh-Adverbien wurde sowohl der WRB-Tag als auch alle Formen von RB akzeptiert, da UPOS nur eine Adverbien-Klasse kennt).

Bei der Analyse fällt auf, dass die anderen Tagger in der Nomen-Klasse viele Falschpositive haben (Wörter, die als Nomen getaggt wurden, aber einer anderen Klasse angehören).

	Satz	Korrekte Wortarten	Stanford	NLTK	SpaCy
0	What also should not be lost in discussion of discrimination is the growing push to implement social policy aimed at reducing the occurrence of discriminatory practices	- PRON - ADV - AUX - PART - AUX - VERB - ADP - NOUN - ADP - NOUN - AUX - DET - VERB - NOUN - PART - VERB - ADJ - NOUN - VERB - SCONJ - VERB - DET - NOUN - ADP - ADJ - NOUN	- WP - RB - MD - RB - VB - VBN - IN - NN - IN - NN - VBZ - DT - VBG - NN - TO - VB - JJ - NN - VBN - IN - VBG - DT - NN - IN - JJ - NNS	- PRON - ADV - AUX - PART - AUX - VERB - ADP - NOUN - ADP - NOUN - AUX - DET - VERB - NOUN - PART - VERB - ADJ - NOUN - VERB - ADP - VERB - DET - NOUN - ADP - NOUN	- PRON - ADV - AUX - PART - AUX - VERB - ADP - NOUN - ADP - NOUN - AUX - DET - VERB - NOUN - PART - VERB - ADJ - NOUN - VERB - ADP - VERB - DET - NOUN - ADP - ADJ - NOUN
1	Indeed the implementation of certain policies is rooted in the assumption that discrimination and biases are at least to some appreciable amount present in modern society	- ADV - DET - NOUN - ADP - ADJ - NOUN - AUX - VERB - ADP - DET - NOUN - SCONJ - NOUN - CCONJ - NOUN - AUX - ADP - ADJ - ADP - DET - ADJ - NOUN - ADJ - ADP - ADJ - NOUN	- RB - DT - NN - IN - JJ - NNS - VBZ - VBN - IN - DT - NN - IN - NN - CC - NNS - VBP - RB - RBS - IN - DT - JJ - NN - JJ - IN - JJ - NN	- ADV - DET - NOUN - ADP - ADJ - NOUN - AUX - VERB - ADP - DET - NOUN - PRON - NOUN - CCONJ - NOUN - AUX - ADV - ADJ - ADP - DET - ADJ - NOUN - NOUN - ADP - ADJ - NOUN	- ADV - DET - NOUN - ADP - ADJ - NOUN - AUX - VERB - ADP - DET - NOUN - SCONJ - NOUN - CCONJ - NOUN - VERB - ADP - ADJ - ADP - DET - ADJ - NOUN - ADJ - ADP - ADJ - NOUN
2	Additionally the analysis address the perceived reasons for reported discrimination experiences	- ADV - DET - NOUN - VERB - DET - VERB - NOUN - ADP - VERB - NOUN - NOUN	- RB - DT - NN - NN - DT - VBN - NNS - IN - VBN - NN - NNS	- ADV - DET - NOUN - VERB - DET - ADJ - NOUN - ADP - VERB - NOUN - NOUN	- ADV - DET - NOUN - NOUN - DET - VERB - NOUN - ADP - VERB - NOUN - NOUN
3	To provide information on the analytical sample as a whole two additional demographic variables are included	- PART - VERB - NOUN - ADP - DET - ADJ - NOUN - ADP - DET - NOUN - NUM - ADJ - ADJ - NOUN - AUX - VERB	- TO - VB - NN - IN - DT - JJ - NN - IN - DT - JJ - CD - JJ - JJ - NNS - VBP - VBN	- PART - VERB - NOUN - ADP - DET - ADJ - NOUN - ADP - DET - ADJ - NUM - ADJ - ADJ - NOUN - AUX - VERB	- PART - VERB - NOUN - ADP - DET - ADJ - NOUN - ADP - DET - ADJ - NUM - ADJ - ADJ - NOUN - AUX - VERB
4	First summary statistics of the study variables and racial categories were produced	- ADV - NOUN - NOUN - ADP - DET - NOUN - NOUN - CCONJ - ADJ - NOUN - AUX - VERB	- JJ - NN - NNS - IN - DT - NN - NNS - CC - JJ - NNS - VBD - VBN	- ADV - ADJ - NOUN - ADP - DET - NOUN - NOUN - CCONJ - ADJ - NOUN - AUX - VERB	- ADJ - NOUN - NOUN - ADP - DET - NOUN - NOUN - CCONJ - ADJ - NOUN - AUX - VERB
5	For example the name Kusillay 1 the language of Essil is the name given by speakers of Kujireray to the Eegimaa language because Essil of which Bajjat was a district is the village which has a border with their village	- ADP - NOUN - DET - NOUN - PROPN - NUM - DET - NOUN - ADP - PROPN - AUX - DET - NOUN - VERB - ADP - NOUN - ADP - PROPN - ADP - DET - PROPN - NOUN - SCONJ - PROPN - ADP - PRON - PROPN - AUX - DET - NOUN - AUX - DET - NOUN - PRON - VERB - DET - NOUN - ADP - PRON - NOUN	- IN - NN - DT - NN - NNP - CD - DT - NN - IN - NNP - VBZ - DT - NN - VBN - IN - NNS - IN - NNP - IN - DT - NNP - NN - IN - NNP - IN - WDT - NNP - VBD - DT - NN - VBZ - DT - NN - WDT - VBZ - DT - NN - IN - PRP$ - NN	- ADP - NOUN - DET - NOUN - PROPN - NUM - DET - NOUN - ADP - PROPN - AUX - DET - NOUN - VERB - ADP - NOUN - ADP - PROPN - ADP - DET - PROPN - NOUN - SCONJ - NOUN - ADP - PRON - PROPN - AUX - DET - NOUN - AUX - DET - NOUN - PRON - VERB - DET - NOUN - ADP - PRON - NOUN	- ADP - NOUN - DET - NOUN - PROPN - NUM - DET - NOUN - ADP - PROPN - AUX - DET - NOUN - VERB - ADP - NOUN - ADP - PROPN - ADP - DET - PROPN - NOUN - SCONJ - PROPN - ADP - DET - PROPN - VERB - DET - NOUN - VERB - DET - NOUN - DET - VERB - DET - NOUN - ADP - PRON - NOUN

Abb. 51: Englisch - Ausschnitt der Sätze, die nur vom DistilBERT-Tagger vollständig korrekt getaggt wurden (eigene Grafik)

6 Ausblick und Diskussion

Die vorliegende Entwicklung, Implementierung und Evaluation eines auf Transformers basierenden POS-Taggers hat die Vorteile dieser Technologie klar vor Augen geführt.

Zu beachten ist jedoch, dass die Modelle auf ausgewählten Datensätzen der Universal Dependencies feingetuned und evaluiert wurden, so dass andere Datensätze zu abweichenden Ergebnissen führen könnten. Zudem wurden die empfohlenen Standardeinstellungen von Hugging Face für den Aufbau des Modells verwendet und nur mit der Anzahl Epochen experimentiert. Auch hier könnte allenfalls noch vertiefte Forschung zu einer Veränderung der Resultate führen.

6.1 Verbesserungsmöglichkeiten

Obwohl der DistilBERT-Tagger in allen drei Sprachen bereits hohe Werte erzielt, hat er noch Verbesserungspotenzial. Insbesondere sollte das Modell trainiert werden, ohne die Satzzeichen zu entfernen. Französische Pronomen, die durch zwei aufeinanderfolgende Vokale verkürzt werden (z.B. Ils **t'**ont abondonné – Sie haben dich verlassen) könnten so vermutlich besser erkannt werden. Zudem würde es dem Transformers-Algorithmus helfen, die Satzstruktur besser zu verstehen. Beispielsweise das Deutsche «dass», das in den meisten Fällen auf ein Komma folgt. Nicht zuletzt könnte so der Tagger Satzzeichen und Symbole im gleichen Zug mit den anderen Wortarten zuordnen. Jedoch muss dafür eine Lösung gefunden werden, um nach der Tokenisierung Satzzeichen zu erkennen, die gemeinsam ein Symbol bilden (z.B. ;-)).

In den französischen und deutschen Testdaten gab es insgesamt sehr wenig Interjektionen. Entsprechend konnte das Modell diese nicht gut lernen. Ein erweiterter Testdatensatz mit einem Akzent auf zusätzlichen Interjektionen könnte hier Abhilfe schaffen. Ebenfalls könnten mehr Sätze hinzugefügt werden, die Wortarten enthalten, die von DistilBERT schlechter als von den anderen Taggern zugeordnet werden konnten. Für Englisch sollte auch spezifisch auf mehr Sätze mit «can be» und seinen verschiedenen Formen sowie «as well» geachtet werden.

Um die Unterscheidung zwischen Nomen und Pronomen zu verbessern, könnte der DistilBERT-POS-Tagger in einem zweiten Schritt mit Named Entity Recognition (NER) kombiniert werden. Somit würden Eigennamen besser erkannt und zusätzlich spezifiziert werden können.

Zuletzt könnte es interessant sein, die Daten der verschiedenen Sprachen zu verbinden und ein Modell für alle drei Sprachen gemeinsam zu trainieren. So könnten nicht nur

mehrsprachige Texte getaggt werden, aber auch Fremdwörter aus den jeweils anderen Sprachen mit einem höheren Score erkannt werden (z.B. «Stempel», ein deutsches Wort, das im Freiburger Französisch statt des französischen Worts «tampon» verwendet wird, beispielsweise).

7 Schlusswort

In dieser Arbeit wurde untersucht, welche Ergebnisse ein auf Transformers basierender Part-of-Speech-Tagger im Gegensatz zu bestehenden Taggern erreicht. Insbesondere sollte herausgefunden werden, ob die Genauigkeit auf Satzebene verbessert werden kann, d.h. ob mehr Sätze vollständig fehlerfrei getaggt werden können. Dafür wurde ein POS-Tagger mit DistilBERT entwickelt und mit Daten der Universal Dependencies in drei Sprachen trainiert und getestet. Mit den gleichen Testdaten wurden auch drei Vergleichstagger evaluiert. Die Ergebnisse wurden anschliessend quantitativ und qualitativ miteinander verglichen.

Insgesamt erreichte der DistilBERT-Tagger über alle getesteten Sprachen hinweg bessere Ergebnisse als bestehende Tagger. Die Performance-Unterschiede zeigten sich insbesondere auf Satzebene, wo der DistilBERT-Tagger eine starke Verbesserung gegenüber den bestehenden Taggern aufwies. Das grösste Problem des DistilBERT-Taggers indessen, war die Unterscheidung von Nomen und Eigennamen.

Der Vergleich zwischen den Sprachen zeigte, dass die Transformers-Technologie sich insbesondere für Französisch eignet, da der DistilBERT-Tagger dort die höchsten Accuracy-Werte erzielte. Im Vergleich der einzelnen Wortarten war es die einzige Sprache, in der praktisch jede Klasse durch den DistilBERT-Tagger Höchstwerte erreicht wurden. Doch auch für Deutsch und Englisch konnte mit DistilBERT hohe Werte erzielt werden.

Der Tagger könnte noch weiter verbessert werden, indem er mit zusätzlichen Sätzen trainiert würde, die Wortarten enthalten, die oft vertauscht oder falsch klassiert wurden. Ebenfalls könnte eine Kombination mit der Named Entity Recognition zu einer Verbesserung der Accuracy-Werte führen.

Zusammengefasst konnte in dieser Arbeit gezeigt werden, dass durch die Deep Learning Transformers-Methode mit dem Aufmerksamkeitsmechanismus, die bestehenden Ergebnisse verbessert werden können und die Genauigkeit auf Satzebene der automatischen Wortartenbestimmung erhöht werden kann.

8 Bibliografie

Ahmadian, S., & Khanteymoori, A. (2015, Mai 26). Training back propagation neural networks using asexual reproduction optimization. *2015 7th Conference on Information and Knowledge Technology (IKT)*. IKT2015 7th International Conference on Information and Knowledge Technology. https://doi.org/10.1109/IKT.2015.7288738

Akbik, A., Blythe, D., & Vollgraf, R. (2018). Contextual String Embeddings for Sequence Labeling. *Proceedings of the 27th International Conference on Computational Linguistics*, 1638–1649. https://aclanthology.org/C18-1139

Ali, W., Kumar, R., Dai, Y., Kumar, J., & Tumrani, S. (2021). Neural Joint Model for Part-of-Speech Tagging and Entity Extraction. *2021 13th International Conference on Machine Learning and Computing*, 239–245. https://doi.org/10.1145/3457682.3457718

Alrajhi, K., & A ELAffendi, M. (2019). Automatic Arabic Part-of-Speech Tagging: Deep Learning Neural LSTM Versus Word2Vec. *International Journal of Computing and Digital System*, *8*(3), 307–315. https://doi.org/10.12785/ijcds/080310

Anastasyev, D., Gusev, I., & Indenbom, E. (2018). Improving Part-of-Speech Tagging via Multi-Task Learning and Character-level Word Representations. *Computational Linguistics and Intellectual Technologies: Proceedings of the International Conference "Dialogue 2018"*. http://arxiv.org/abs/1807.00818

Ankita, & Abdul Nazeer, K. A. (2018). Part-of-speech Tagging and Named Entity Recognition Using Improved Hidden Markov Model and Bloom Filter. *2018 International Conference on Computing, Power and Communication Technologies (GUCON)*, 1072–1077. https://doi.org/10.1109/GUCON.2018.8674901

Arakelyan, G., Hambardzumyan, K., & Khachatrian, H. (2018). Towards JointUD: Part-of-speech Tagging and Lemmatization using Recurrent Neural Networks. *arXiv:1809.03211 [cs]*. http://arxiv.org/abs/1809.03211

Awad, M., & Khanna, R. (2015). Support Vector Machines for Classification. In *Efficient Learning Machines: Theories, Concepts, and Applications for Engineers and System Designers* (S. 39–66). Springer Nature. https://library.oapen.org/handle/20.500.12657/28170

Azeraf, E., Monfrini, E., Vignon, E., & Pieczynski, W. (2020). Hidden Markov Chains, Entropic Forward-Backward, and Part-Of-Speech Tagging. *arXiv:2005.10629 [cs, stat]*. http://arxiv.org/abs/2005.10629

Baishya, D., & Baruah, R. (2021). Highly Efficient Parts of Speech Tagging in Low Resource Languages with Improved Hidden Markov Model and Deep Learning. *International Journal of Advanced Computer Science & Applications*, *12*(10). https://doi.org/10.14569/IJACSA.2021.0121011

Banga, R., & Mehndiratta, P. (2017). Tagging Efficiency Analysis on Part of Speech Taggers. *2017 International Conference on Information Technology (ICIT)*, 264–267. https://doi.org/10.1109/ICIT.2017.57

Bărbulescu, A., & Morariu, . I. (2020). Part of Speech Tagging Using Hidden Markov Models. *International Journal of Advanced Statistics and IT&C for Economics and Life Sciences*, *10*(1), 31–42. https://doi.org/10.2478/ijasitels-2020-0005

Bartsch, S. (2019, Januar). *Stanford PoS Tagger: Tagging from Python*. linguisticsweb.org. https://www.linguisticsweb.org/doku.php?id=linguisticsweb:tutorials:linguistics_tutorials:automaticannotation:stanford_pos_tagger_python

Bibliographisches Institut GmbH. (2022). Homograf. In *Duden*. https://www.duden.de/rechtschreibung/Homograf

Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2017). Enriching Word Vectors with Subword Information. *arXiv:1607.04606v2*. https://doi.org/10.48550/ARXIV.1607.04606

Bölücü, N., & Can, B. (2021). A Cascaded Unsupervised Model for PoS Tagging. *ACM Transactions on Asian and Low-Resource Language Information Processing*, *20*(1), 17:1-17:23. https://doi.org/10.1145/3447759

Brants, T. (2000). TnT - A Statistical Part-of-Speech Tagger. *arXiv:cs/0003055*. http://arxiv.org/abs/cs/0003055

Brill, E. (1992). A simple rule-based part of speech tagger. *Proceedings of the third conference on Applied natural language processing*, 152–155. https://doi.org/10.3115/974499.974526

Carrigan, M., Debut, L., Gugger, S., Noyan, M., Saulnier, L., Tunstall, L., & Werra, L. von. (o. J.). *Hugging Face Course*. huggingface.co. Abgerufen 13. Mai 2022, von https://huggingface.co/course/chapter1/1

Chaudhary, A., Anastasopoulos, A., Sheikh, Z., & Neubig, G. (2021). Reducing Confusion in Active Learning for Part-Of-Speech Tagging. *Transactions of the Association for Computational Linguistics*, *9*, 1–16. https://doi.org/10.1162/tacl_a_00350

Chollet, F. (2021). *Deep Learning with Python* (Second edition). Manning.

CodeEmporium. (13. Januar 2020). *Transformer Neural Networks—EXPLAINED! (Attention is all you need)*. https://www.youtube.com/watch?v=TQQlZhbC5ps

Collins, M. (2002). Discriminative Training Methods for Hidden Markov Models: Theory and Experiments with Perceptron Algorithms. *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing*, *10*, 1–8.

Cotta, C., Mathieson, L., & Moscato, P. (2017). Memetic Algorithms. In R. Martí, P. Panos, & M. G. C. Resende (Hrsg.), *Handbook of Heuristics* (S. 1–32). Springer. https://doi.org/10.1007/978-3-319-07153-4_29-1

Culurciello, E. (10. Januar 2019). The fall of RNN / LSTM. *Medium*. https://towards-datascience.com/the-fall-of-rnn-lstm-2d1594c74ce0

Cutting, D., Kupiec, J., Pedersen, J., & Sibun, P. (1992). A practical part-of-speech tagger. *Proceedings of the Third Conference on Applied Natural Language Processing*, 133–140. https://doi.org/10.3115/974499.974523

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv:1810.04805 [cs]*. http://arxiv.org/abs/1810.04805

Dhumal Deshmukh, R., & Kiwelekar, A. (2020). Deep Learning Techniques for Part of Speech Tagging by Natural Language Processing. *2020 2nd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA)*, 76–81. https://doi.org/10.1109/ICIMIA48430.2020.9074941

Dirac, L. (3. Dezember 2019). *LSTM is dead. Long Live Transformers!*

https://www.youtube.com/watch?v=S27pHKBEp30 Facebook Inc. (2022). *FastText*. https://fasttext.cc/index.html

Fanoon, A. R. F. S., & Uwanthika, G. A. I. (2019). Part of speech tagging for Twitter conversations using Conditional Random Fields model. *2019 International Research Conference on Smart Computing and Systems Engineering (SCSE)*, 108–112. https://doi.org/10.23919/SCSE.2019.8842669

Farrah, S., El Manssouri, H., Ziyati, E. H., & Ouzzif, M. (2018). An hybrid approach to improve part of speech tagging system. *2018 International Conference on Intelligent Systems and Computer Vision (ISCV)*, 1–6. https://doi.org/10.1109/ISACV.2018.8354032

Finn, C., Abbeel, P., & Levine, S. (2017). Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. *arXiv:1703.03400 [cs]*. https://doi.org/10.48550/arXiv.1703.03400

Gers, F. A. (2001). *Long Short-Term Memory in Recurrent Neural Networks* [Ecole polytechnique fédérale]. http://www.felixgers.de/papers/phd.pdf

Gimenez, J., & Màrquez, L. (2004). Fast and Accurate Part-of-Speech Tagging: The SVM Approach Revisited. In *Recent Advances in Natural Language Processing III: Selected papers from RANLP* (Bd. 260, S. 153-). John Benjamins Publishing Company. https://doi.org/10.1075/cilt.260.17gim

Goodfellow, I., Bengio, Y., & Courville, A. (2018). *Deep Learning: Das umfassende Handbuch*. mitp.

Google. (o. J.). *Google Colaboratory*. Abgerufen 31. Juli 2022, von https://colab.research.google.com/

Google. (2013). *Word2vec*. https://code.google.com/archive/p/word2vec/

Gopalakrishnan, A., Soman, K. P., & Premjith, B. (2019). Part-of-Speech Tagger for Biomedical Domain Using Deep Neural Network Architecture. *2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, 1–5. https://doi.org/10.1109/ICCCNT45670.2019.8944559

Graves, A., & Schmidhuber, J. (2005). Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Networks*, *18*(5), 602– 610. https://doi.org/10.1016/j.neunet.2005.06.042

Griebenouw, A., Drevin, G., & Snyman, D. (2019). A Combination Part of Speech Tagger using Selected Voting Methods. *2019 International Multidisciplinary Information Technology and Engineering Conference (IMITEC)*, 1–6. https://doi.org/10.1109/IMITEC45504.2019.9015872

Heid, S., Wever, M., & Hüllermeier, E. (2021). Reliable Part-of-Speech Tagging of Historical Corpora through Set-Valued Prediction. *arXiv:2008.01377 [cs, stat]*. http://arxiv.org/abs/2008.01377

Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, *9*(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

Honnibal, M. (18. September 2013). A Good Part-of-Speech Tagger in about 200 Lines of Python. *Explosion*. https://explosion.ai/blog/part-of-speech-pos-tagger-in-python

Huang, Z., Xu, W., & Yu, K. (2015). Bidirectional LSTM-CRF Models for Sequence Tagging. *arXiv:1508.01991 [cs]*. http://arxiv.org/abs/1508.01991

Hugging Face. (o. J.-a). *Bert-base-multilingual-cased*. Abgerufen 11. August 2022, von https://huggingface.co/bert-base-multilingual-cased

Hugging Face. (o. J.-a). *DistilBERT*. huggingface.co. Abgerufen 12. März 2022, von https://huggingface.co/docs/transformers/model_doc/distilbert

Hugging Face. (o. J.-b). *Distilbert-base-multilingual-cased*. huggingface.co. Abgerufen 14. Mai 2022, von https://huggingface.co/distilbert-base-multilingual-cased

Hugging Face. (o. J.-c). *Processing the data—Hugging Face Course*. hugging-face.co. Abgerufen 16. Juli 2022, von https://huggingface.co/course/chapter3/2

Hugging Face. (o. J.-d). *Token classification—Hugging Face Course*. hugging-face.co. Abgerufen 15. Juli 2022, von https://huggingface.co/course/chapter7/2

Hugging Face. (2021b). https://huggingface.co/

JetBrains. (2022). *PyCharm: The Python IDE for Professional Developers by Jet-Brains*. Jetbrains.Com. https://www.jetbrains.com/pycharm/

Jurafsky, D., & Martin, J. H. (2022). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition* (Third Edition draft). https://web.stanford.edu/~jurafsky/slp3/

Kalchbrenner, N., Espeholt, L., Simonyan, K., Oord, A. van den, Graves, A., & Kavukcuoglu, K. (2017). Neural Machine Translation in Linear Time. *arXiv:1610.10099 [cs]*. http://arxiv.org/abs/1610.10099

Kemos, A., Adel, H., & Schütze, H. (2020). Neural Semi-Markov Conditional Random Fields for Robust Character-Based Part-of-Speech Tagging. *arXiv:1808.04208 [cs]*. http://arxiv.org/abs/1808.04208

Khan, W., Daud, A., Khan, K., Nasir, J. A., Basheri, M., Aljohani, N., & Alotaibi, F. S. (2019). Part of Speech Tagging in Urdu: Comparison of Machine and Deep Learning Approaches. *IEEE Access*, *7*, 38918–38936. https://doi.org/10.1109/ACCESS.2019.2897327

Kim, Y., Jernite, Y., Sontag, D., & Rush, A. M. (2015). Character-Aware Neural Language Models. *arXiv:1508.06615 [cs, stat]*. http://arxiv.org/abs/1508.06615

Kolesau, A., Sesok, D., & Rybokas, M. (2018). A Character-Based Part-of-Speech Tagger with Feedforward Neural Networks. *Romanian Journal of Information Science and Technology*, *21*(4), 446–459.

Kondratyuk, D., & Straka, M. (2019). 75 Languages, 1 Model: Parsing Universal Dependencies Universally. *arXiv:1904.02099 [cs]*. http://arxiv.org/abs/1904.02099

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, 1097–1105.

Kumar, S., Kumar, M. A., & Soman, K. P. (2019). Deep Learning Based Part-of-Speech Tagging for Malayalam Twitter Data (Special Issue: Deep Learning Techniques for Natural Language Processing). *Journal of Intelligent Systems*, *28*(3), 423–435. https://doi.org/10.1515/jisys-2017-0520

Lafferty, J. D., McCallum, A., & Pereira, F. C. N. (2001). Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. *Proceedings of the Eighteenth International Conference on Machine Learning*, 282–289.

Lecun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), 2278–2324. https://doi.org/10.1109/5.726791

Lee, C., Kim, Y.-B., Lee, D., & Lim, H. (2018). Character-Level Feature Extraction with Densely Connected Networks. *arXiv:1806.09089 [cs]*. http://arxiv.org/abs/1806.09089

Lhoest, Q., del Moral, A. V., Jernite, Y., Thakur, A., von Platen, P., Patil, S., Chaumond, J., Drame, M., Plu, J., Tunstall, L., Davison, J., Šaško, M., Chhablani, G., Malik, B., Brandeis, S., Scao, T. L., Sanh, V., Xu, C., Patry, N., … Wolf, T. (2021). *Datasets: A Community Library for Natural Language Processing* (arXiv:2109.02846). arXiv. https://doi.org/10.48550/arXiv.2109.02846

C. (2021). Datasets: A Community Library for Natural Language Processing. *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 175–184. https://aclanthology.org/2021.emnlp- demo.21

Li, H., Mao, H., & Wang, J. (2021). Part-of-Speech Tagging with Rule-Based Data Preprocessing and Transformer. *Electronics (Basel)*, *11*(1), 56-. https://doi.org/10.3390/electronics11010056

Liu, Y. (2017). Investigation of Viterbi Algorithm Performance on Part-of-Speech Tagger of Natural Language Processing. *2017 International Conference on Computer Systems, Electronics and Control (ICCSEC)*, 1430–1433. https://doi.org/10.1109/ICCSEC.2017.8446837

Ma, X., & Hovy, E. (2016). End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF. *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1064–1074. https://doi.org/10.18653/v1/P16-1101

Maksutov, A. A., Zamyatovskiy, V. I., Morozov, V. O., & Dmitriev, S. O. (2021). The Transformer Neural Network Architecture for Part-of-Speech Tagging. *2021 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (ElConRus)*, 536–540. https://doi.org/10.1109/ElConRus51938.2021.9396231

Manning, C. (10. Mai 2021). *Lecture 84—Some Methods and Results on Sequence Models for POS Tagging*. https://www.youtube.com/watch?v=QMZsurbHVwQ

Manning, C. D. (2011). Part-of-Speech Tagging from 97% to 100%: Is It Time for Some Linguistics? In A. F. Gelbukh (Hrsg.), *Computational Linguistics and Intelligent Text Processing* (Bd. 6608, S. 171–189). Springer. https://doi.org/10.1007/978-3- 642-19400-9 14

McCoy, N. (27. Oktober 2016). Evaluating NLTK Taggers Tutorial. *Natemccoy & The Linguistic Multiverse*. https://natemccoy.github.io/2016/10/27/evaluatingnltktaggerstutorial.html

McDonald, R., Nivre, J., Quirmbach-Brundage, Y., Goldberg, Y., Das, D., Ganchev, K., Hall, K., Petrov, S., Zhang, H., Täckström, O., Bedini, C., Bertomeu Castelló, N., & Lee, J. (2013). Universal Dependency Annotation for Multilingual Parsing. *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 92–97. https://aclanthology.org/P13-2017

Meftah, S., Tamaazousti, Y., Semmar, N., Essafi, H., & Sadat, F. (2019). Joint Learning of Pre-Trained and Random Units for Domain Adaptation in Part-of-Speech Tagging. *arXiv:1904.03595 [cs, stat]*. http://arxiv.org/abs/1904.03595

Memari, I. (12. Januar 2021). Precision, Accuracy and F1 Score for Multi-Label Classification. *Synthesio Engineering*. https://medium.com/synthesio-engineering/precision-accuracy-and-f1-score-for-multi-label-classification-34ac6bdfb404

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). *Efficient Estimation of Word Representations in Vector Space* (arXiv:1301.3781). arXiv. https://doi.org/10.48550/arXiv.1301.3781

Mundotiya, R. K., Mehta, A., Baruah, R., & Singh, A. K. (2021). Integration of morphological features and contextual weightage using monotonic chunk attention for part of speech tagging. *Journal of King Saud University - Computer and Information Sciences*. https://doi.org/10.1016/j.jksuci.2021.08.023

Muñoz-Valero, D., Rodriguez-Benitez, L., Jimenez-Linares, L., & Moreno-Garcia, J. (2020). Using Recurrent Neural Networks for Part-of-Speech Tagging and Subject and

Predicate Classification in a Sentence. *International Journal of Computational Intelligence Systems*, *13*(1), 706–716. https://doi.org/10.2991/ijcis.d.200527.005

Nakayama, H. (2018). *seqeval: A Python framework for sequence labeling evaluation*

(1.2.2) [Python]. https://github.com/chakki-works/seqeval

NLTK Project. (2022a). *Source code for nltk.tag.perceptron*. https://www.nltk.org/_modules/nltk/tag/perceptron.html

NLTK Project. (2022b, März 25). *NLTK: Natural Language Toolkit*. https://www.nltk.org/

NumPy Developers. (2022). *NumPy* (1.22.2) [C, Python; MacOS, Microsoft: Windows, POSIX, Unix]. https://www.numpy.org

Olah, C. (27. August 2015). Understanding LSTM Networks. *Colah's Blog*. https://colah.github.io/posts/2015-08-Understanding-LSTMs/

Otman, M., Rachid, E. A., & Mohamed, B. (2021). Amazigh Part Of Speech Tagging using Gated recurrent units (GRU). *2021 7th International Conference on Optimization and Applications (ICOA)*, 1–6. https://doi.org/10.1109/ICOA51614.2021.9442662

Partalidou, E., Spyromitros-Xioufis, E., Doropoulos, S., Vologiannidis, S., & Diamantaras, K. I. (2019). Design and implementation of an open source Greek POS Tagger and Entity Recognizer using spaCy. *arXiv:1912.10162*. https://doi.org/10.48550/arXiv.1912.10162

Patterson, J., & Gibson, A. (2017). *Deep Learning: A practitioner's approach*. O' eilly.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, É. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, *12*(85), 2825– 2830.

Pennington, J., Socher, R., & Manning, C. (2014a). Glove: Global Vectors for Word Representation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1532–1543. https://doi.org/10.3115/v1/D14-1162

Pennington, J., Socher, R., & Manning, C. D. (2014b). *GloVe: Global Vectors for Word Representation*. nlp.stanford.edu. https://nlp.stanford.edu/projects/glove/

Plank, B., & Agić, Ž. (2018). istant Supervision from isparate Sources for Low-Resource Part-of-Speech Tagging. *arXiv:1808.09733 [cs]*. http://arxiv.org/abs/1808.09733

Plank, B., Klerke, S., & Agic, Z. (2018). The Best of Both Worlds: Lexical Resources To Improve Low-Resource Part-of-Speech Tagging. *Natural Language Engineering*, *1*, 1–29.

POS Tagging (State of the art). (2019). In *Wiki of the Association for Computational Linguistics*. https://aclweb.org/aclwiki/POS_Tagging_(State_of_the_art)

Premjith, B., Soman, K. P., & Prabaharan, P. (2018). A deep learning based Part-of-Speech (POS) tagger for Sanskrit language by embedding character level features. *Proceedings of the 10th annual meeting of the Forum for Information Retrieval Evaluation*, 56–60. https://doi.org/10.1145/3293339.3293352

Project Jupyter. (2022). https://jupyter.org

Rabiner, L. R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, *77*(2), 257–286. https://doi.org/10.1109/5.18626

Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018). Improving Language Understanding by Generative Pre-Training. *Open AI*, 12.

Ratnaparkhi, A. (1996). A Maximum Entropy Model for Part-Of-Speech Tagging. *Conference on Empirical Methods in Natural Language Processing*, 133–142. https://aclanthology.org/W96-0213

Rothman, D. (2021). *Transformers for natural language processing: Build innovative deep neural network architectures for NLP with Python, Pytorch, TensorFlow, BERT, RoBERTa, and more*. Packt Publishing.

Sadredini, E., Guo, D., Bo, C., Rahimi, R., Skadron, K., & Wang, H. (2018). A Scalable Solution for Rule-Based Part-of-Speech Tagging on Novel Hardware

Accelerators. Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 665–674. https://doi.org/10.1145/3219819.3219889

Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2020). DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter. arXiv:1910.01108 [cs]. http://arxiv.org/abs/1910.01108

Sarawagi, S., & Cohen, W. W. (2004). Semi-Markov Conditional Random Fields for Information Extraction. Advances in Neural Information Processing Systems, 1185– 1192.

Sarkar, D. (2019). Text Analytics with Python: A Practitioner's Guide to Natural Language Processing (2nd ed.). Apress. https://doi.org/10.1007/978-1-4842-4354-1

Schmid, H. (1994). Probabilistic Part-of-Speech Tagging using Decision Trees. Conference on New Methods in Language Processing. Conference on New Methods in Language Processing, Manchester, UK.

https://citeseerx.ist.psu.edu/viewdoc/down-load;jses-sionid=2541E8F4EA99B5A4D0E5572852852F13?doi=10.1.1.28.1139&rep=rep1&type=pdf

Shen, Y., Mai, Y., Shen, X., Ding, W., & Guo, M. (2020). Jointly Part-of-Speech Tagging and Semantic Role Labeling Using Auxiliary Deep Neural Network Model. Computers, Materials, & Continua, 65(1), 529–541. http://dx.doi.org/10.32604/cmc.2020.011139

Sierra Martínez, L. M., Cobos, C. A., & Corrales, J. C. (2017). Memetic Algorithm Based on Global-Best Harmony Search and Hill Climbing for Part of Speech Tagging. In A. Ghosh, R. Pal, & R. Prasath (Hrsg.), Mining Intelligence and Knowledge Exploration (S. 198–211). Springer International Publishing. https://doi.org/10.1007/978-3- 319-71928-3_20

Solano Jiménez, M. A., Tobar Cifuentes, J. J., Sierra Martínez, L. M., & Cobos Lozada, C. A. (2020). Adaptation, Comparison, and Improvement of Metaheuristic Algorithms to the Part-of-Speech Tagging Problem. Revista FI-UPTC, 29(54), e11762– e11762. https://doi.org/10.19053/01211129.v29.n54.2020.11762

Srivastava, P., Chauhan, K., Aggarwal, D., Shukla, A., Dhar, J., & Jain, V. P. (2018). Deep Learning Based Unsupervised POS Tagging for Sanskrit. Proceedings of the 2018 International Conference on Algorithms, Computing and Artificial Intelligence, 1–6. https://doi.org/10.1145/3302425.3302487

Stenström, E. (2022). CoNLL-U Parser (4.5.2) [Python]. https://github.com/EmilStens-trom/conllu

The pandas development team. (2022). Pandas-dev/pandas (1.4.1) [Python]. Zenodo. https://doi.org/10.5281/zenodo.6053272

The Stanford Natural Language Processing Group. (2020). Stanford Log-linear Part-Of-Speech Tagger. https://nlp.stanford.edu/software/tagger.shtml

Togatorop, P. R., Siagian, R., Nainggolan, Y., & Simanungkalit, K. (2020). Implementation of ontology-based on Word2Vec and DBSCAN for part-of-speech. Proceedings *of the 5th International Conference on Sustainable Information Engineering and Technology*, 51–56. https://doi.org/10.1145/3427423.3427431

Toutanova, K., Klein, D., Manning, C. D., & Singer, Y. (2003). Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network. *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, 252–259. https://aclanthology.org/N03-1033

Toutanova, K., & Manning, C. D. (2000). Enriching the knowledge sources used in a maximum entropy part-of-speech tagger. *Proceedings of the 2000 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora Held in Conjunction with the 38th Annual Meeting of the Association for Computational Linguistics* -, *13*, 63–70. https://doi.org/10.3115/1117794.1117802

Tunstall, L., Werra, L. von, & Wolf, T. (2022). *Natural Language Processing with Transformers.* O' eilly Media, Incorporated.

Tutorials Point. (2022). *Part of Speech (PoS) Tagging*. tutorialspoint.com. https://www.tutorialspoint.com/natural_language_processing/natural_language_processing_part_of_speech_tagging.htm

Universal Dependencies. (2021a). *ADJ: adjective*. universaldependencies.org. https://universaldependencies.org/u/pos/ADJ.html

Universal Dependencies. (2021b). *ADP: adposition*. universaldependencies.org. https://universaldependencies.org/u/pos/all.html#al-u-pos/ADP

Universal Dependencies. (2021c). *ADP: adposition [in English]*. universaldependencies.org. https://universaldependencies.org/en/pos/ADP.html

Universal Dependencies. (2021d). *ADV: adverb*. universaldependencies.org. https://universaldependencies.org/u/pos/ADV.html

Universal Dependencies. (2021e). *AUX: auxiliary verb*. universaldependencies.org. https://universaldependencies.org/u/pos/AUX_.html

Universal Dependencies. (2021f). *AUX: auxiliary verb [in English]*. universaldependencies.org. https://universaldependencies.org/en/pos/AUX_.html

Universal Dependencies. (2021g). *CCONJ: coordinating conjunction*. universaldependencies.org. https://universaldependencies.org/u/pos/CCONJ.html

Universal Dependencies. (2021h). *DET: determiner*. universaldependencies.org. https://universaldependencies.org/u/pos/DET.html

Universal Dependencies. (2021i). *INTJ: interjection*. universaldependencies.org. https://universaldependencies.org/u/pos/INTJ.html

Universal Dependencies. (2021j). *NOUN: noun*. universaldependencies.org. https://universaldependencies.org/u/pos/NOUN.html

Universal Dependencies. (2021k). *NUM: numeral*. universaldependencies.org. https://universaldependencies.org/u/pos/NUM.html

Universal Dependencies. (2021l). *PART: particle*. universaldependencies.org. https://universaldependencies.org/u/pos/PART.html

Universal Dependencies. (2021m). *PART: particle [in English]*. universaldependencies.org. https://universaldependencies.org/en/pos/PART.html

Universal Dependencies. (2021n). *PRON: pronoun*. universaldependencies.org. https://universaldependencies.org/u/pos/PRON.html

Universal Dependencies. (2021o). *PROPN: proper noun*. universaldependencies.org. https://universaldependencies.org/u/pos/PROPN.html

Universal Dependencies. (2021p). *SCONJ: subordinating conjunction [in English]*. universaldependencies.org. https://universaldependencies.org/en/pos/SCONJ.html

Universal Dependencies. (2021q). https://universaldependencies.org/

Universal Dependencies. (2021r). *Universal POS tags*. universaldependencies.org. https://universaldependencies.org/u/pos/

Universal Dependencies. (2021s). *VERB: verb*. universaldependencies.org. https://universaldependencies.org/u/pos/VERB.html

Universal Dependencies. (2021t). *X: other*. universaldependencies.org. https://universaldependencies.org/u/pos/X.html

Universal Dependencies. (2022a). *UD_French-GSD*. universaldependencies.org. https://universaldependencies.org/treebanks/fr_gsd/index.html

Universal Dependencies. (2022b). *UD_German-GSD*. universaldependencies.org. https://universaldependencies.org/treebanks/de_gsd/index.html

University of Pennsylvania. Departement of Linguistics. (2003). *Penn Treebank P.O.S. Tags*. ling.upenn.edu.

https://www.ling.up-enn.edu/courses/Fall_2003/ling001/penn_treebank_pos.html

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017, Dezember 5). *Attention Is All You Need*. 31st Conference on Neural Information Processing Systems, Long Beach, CA, USA. http://arxiv.org/abs/1706.03762

Wallach, H. M. (2004). *Conditional Random Fields: An Introduction*. 9.

Wang, C., Wang, Y., Mo, J., & Wang, S. (2020). End-to-end relation extraction based on part of speech syntax tree. *2020 2nd International Conference on Machine Learning,*

Big Data and Business Intelligence (MLBDBI), 5–9. https://doi.org/10.1109/MLBDBI51377.2020.00008

Warjri, S., Pakray, P., Lyngdoh, S. A., & Maji, A. K. (2021). Part-of-Speech (POS) Tagging Using Deep Learning-Based Approaches on the Designed Khasi POS Corpus. *ACM Transactions on Asian and Low-Resource Language Information Processing*, *21*(3), 63:1-63:24. https://doi.org/10.1145/3488381

Waskom, M. L. (2021). seaborn: Statistical data visualization. *Journal of Open Source Software*, *6*(60), 3021. https://doi.org/10.21105/joss.03021

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Le Scao, T., Gugger, S., … Rush, A. (2020). Transformers: State- of-the-Art Natural Language Processing. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 38–45. https://doi.org/10.18653/v1/2020.emnlp-demos.6

Xue, X., & Zhang, J. (2021). Part-of-speech tagging of building codes empowered by deep learning and transformational rules. *Advanced Engineering Informatics*, *47*(101235). https://doi.org/10.1016/j.aei.2020.101235

Yang, L., Zhang, M., Liu, Y., Sun, M., Yu, N., & Fu, G. (2018). Joint POS Tagging and Dependence Parsing With Transition-Based Neural Networks. *IEEE/ACM Transactions on Audio, Speech and Language Processing*, *26*(8), 1352–1358. https://doi.org/10.1109/TASLP.2017.2788181

Ye, A. (13. September 2020). Long Short-Term Memory etworks Are ying: What's Replacing It? *Medium*. https://medium.com/mlearning-ai/long-short-term-memory-networks-are-dying-whats-replacing-it-5ff3a99399fe

Yu, Y., Si, X., Hu, C., & Zhang, J. (2019). A Review of Recurrent Neural Networks: LSTM Cells and Network Architectures. *Neural Computation*, *31*(7), 1235–1270. https://doi.org/10.1162/neco_a_01199

Zeldes, A. (2017). The GUM Corpus: Creating Multilayer Resources in the Classroom. *Language Resources and Evaluation*, *51*(3), 581–612. http://dx.doi.org/10.1007/s10579-016-9343-x

Zhang, X., Li, Y., Zhang, P., & Yan, Y. (2020). Lingual-Agnostic Meta-Learning for Low-Resource Part-of-Speech Tagging. *2020 The 8th International Conference on Information Technology: IoT and Smart City*, 35–39. https://doi.org/10.1145/3446999.3447006

Zhang, Y., Chen, H., Zhao, Y., Liu, Q., & Yin, D. (2018). Learning Tag Dependencies for Sequence Tagging. *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*, 4581–4587. https://doi.org/10.24963/ijcai.2018/637

Zhou, D., Zhang, Z., Zhang, M.-L., & He, Y. (2018). Weakly Supervised POS Tagging without Disambiguation. *ACM Transactions on Asian and Low-Resource Language Information Processing*, *17*(4), 35:1-35:19. https://doi.org/10.1145/3214707

9 Anhang

Anhang 1: Übersetzung Penn-Treebank-Tagset in UPOS-Tags

UPENN	UPOS
CC	CCONJ
CD	NUM
DT	DET
EX	PRON
FW	X
IN	SCONJ
JJ	ADJ
JJR	ADJ
JJS	ADJ
LS	_
MD	AUX
NN	NOUN
NNS	NOUN
NNP	PROPN
NNPS	PROPN
PDT	DET
POS	PART
PRP	PRON
PRP$	PRON
RB	ADV
RBR	ADV
RBS	ADV
RP	ADP
TO	ADP
UH	INTJ
VB	VERB
VBD	BERB
VBG	VERB

VBN	VERB
VBP	VERB
VBZ	VERB
WDT	DET
WP	PRON
WP$	PRON
WRB	ADV
$	SYM
‘’	PUNCT
(	PUNT
)	PUNCT
,	PUNCT
--	PUNCT
.	PUNCT
:	PUNCT

Anhang 2: Homograph-Sätze auf Deutsch

Gross-/Kleinschreibung beachtet

Wort	Satz	True POS	DistilBERT	Stanford	SpaCy	NLTK
Modern	Heutzutage ist es modern, wenn die Äpfel in der Küche etwas modern.	ADJ, VERB	ADJ, ADJ	ADJ, ADJ	ADV, ADV	ADV, ADJ
Steuern	Wir steuern auf das Gemeindehaus zu, um unsere Steuern zu begleichen.	VERB, NOUN	VERB, NOUN	VERB, NOUN	VERB, NOUN	VERB, NOUN
Beige	Das beige Handtuch kommt auf die grosse Beige.	ADJ, NOUN	ADJ, NOUN	ADJ, NOUN	ADJ, NOUN	ADJ, NOUN
Weg	Solange die Kuhherde nicht weg ist, gehe ich nicht zurück auf den Weg!	ADV, NOUN	ADV, NOUN	ADV, NOUN	ADV, NOUN	ADV, NOUN
Flucht	Die Chefin flucht, bis sie alle ihre Mitarbeiter in die Flucht geschlagen hat.	VERB, NOUN	VERB, NOUN	VERB, NOUN	VERB, NOUN	VERB, NOUN
Sucht	Aufgrund seiner Sucht nach Adrenalin, sucht er ständig einen noch höheren Gipfel.	NOUN, VERB	NOUN, VERB	NOUN, VERB	NOUN, VERB	NOUN, VERB
Liebe	Ich liebe mein Haustier; es ist meine grosse Liebe.	VERB, NOUN	VERB, NOUN	VERB, NOUN	VERB, NOUN	VERB, NOUN
Lache	Ich lache noch heute, wenn ich daran denke, wie sich plötzliche eine rote Lache im ganzen Büro ausbreitete.	VERB, NOUN	VERB, NOUN	VERB, NOUN	VERB, NOUN	VERB, NOUN

Gross-/Kleinschreibung nicht beachtet

Wort	Satz	True POS	DistilBERT	Stanford	SpaCy	NLTK
Modern	Heutzutage ist es modern, wenn die Äpfel in der Küche etwas modern.	ADJ, VERB	ADJ, ADJ	ADJ, ADJ	ADV, ADV	ADV, ADJ
Steuern	Wir steuern auf das Gemeindehaus zu, um unsere Steuern zu begleichen.	VERB, NOUN	VERB, NOUN	VERB, VERB	VERB, NOUN	VERB, NOUN
Beige	Das beige Handtuch kommt auf die grosse Beige.	ADJ, NOUN	NOUN, NOUN	ADJ, ADJ	ADJ, ADJ	ADJ, ADJ
Weg	Solange die Kuhherde nicht weg ist, gehe ich nicht zurück auf den Weg!	ADV, NOUN	ADV, ADV	ADV, ADV	ADV, NOUN	ADV, ADV
Flucht	Die Chefin flucht, bis sie alle ihre Mitarbeiter in die Flucht geschlagen hat.	VERB, NOUN	VERB, NOUN	VERB, NOUN	NOUN, NOUN	ADJ, ADJ
Sucht	Aufgrund seiner Sucht nach Adrenalin, sucht er ständig einen noch höheren Gipfel.	NOUN, VERB	NOUN, VERB	VERB, VERB	NOUN, VERB	ADJ, VERB
Liebe	Ich liebe mein Haustier; es ist meine grosse Liebe.	VERB, NOUN	VERB, NOUN	VERB, NOUN	VERB, NOUN	VERB, ADJ
Lache	Ich lache noch heute, wenn ich daran denke, wie sich plötzliche eine rote Lache im ganzen Büro ausbreitete.	VERB, NOUN	VERB, NOUN	VERB, ADJ	VERB, NOUN	VERB, ADJ

Anhang 3: Homograph-Sätze auf Französisch

Sätze: https://fr.m.wikipedia.org/wiki/Homographe

Wort	Satz	Übersetzung	True POS	DistilBERT	Stanford	SpaCy	NLTK
Est	Il est né à l'est	Er ist im Osten geboren.	AUX, NOUN	AUX, NOUN	AUX, AUX	AUX, AUX	AUX, NOUN
Son	Le son de son piano fait rêver.	Der Klang seines Klaviers lässt träumen.	NOUN, PRON	NOUN, DET	NOUN, DET	NOUN, DET	DET, DET
Fier	Cet homme est fier ; peut-on s'y fier ?	Dieser Mann ist stolz; kann man ihm trauen?	ADJ, VERB	ADJ, VERB	ADJ, VERB	ADJ, NOUN	ADJ, VERB
Résident	Ils résident à Paris chez le résident d'une ambassade étrangère.	Sie wohnen in Paris beim Bewohner einer ausländischen Botschaft.	VERB, NOUN	VERB, NOUN	NOUN, NOUN	VERB, NOUN	VERB, NOUN
Faille	Je ne pense pas qu'il faille relever la faille de son raisonnement.	Ich glaube nicht, dass wir den Fehler in seiner Argumentation hervorheben sollten.	VERB, NOUN	VERB, NOUN	NOUN, NOUN	VERB, NOUN	VERB, NOUN
Rose	Elle arrose la rose rose.	Sie giesst die rosa Rose.	NOUN, ADJ	NOUN, ADJ	NOUN, ADJ	NOUN, ADJ	NOUN, ADJ
Excellent	Ces cuisiniers excellent à composer cet excellent plat.	Diese Köche zeichnen sich durch die Zusammenstellung dieses ausgezeichneten Gerichts aus	VERB, ADJ	ADJ, ADJ	ADJ, ADJ	ADJ, ADJ	ADJ, NOUN
Parent	Ces dames se parent de fleurs pour leur parent.	Diese Damen schmücken sich mit Blumen für ihren Verwandten.	VERB, NOUN	VERB, NOUN	VERB, NOUN	VERB, NOUN	VERB, NOUN

Anhang 4: Homograph-Sätze auf Englisch

Wort	Satz	Übersetzung	True POS	DistilBERT	Stanford	SpaCy	NLTK
Can	Can you open this can for me please?	Kannst du diese Dose bitte für mich öffnen?	AUX, NOUN	AUX, AUX	AUX, NOUN	AUX, AUX	AUX, AUX
Park	It is not possible to park inside of the National Park.	Es ist nicht möglich, innerhalb des Nationalparks zu parkieren.	VERB, NOUN	VERB, NOUN	VERB, NOUN	VERB, NOUN	VERB, NOUN
Close	My close friends always close my window when it starts raining while I am away.	Meine engen Freunde schliessen jeweils mein Fenster, wenn es zu regnen beginnt, während ich weg bin.	ADJ, VERB	ADJ, VERB	ADJ, VERB	ADJ, VERB	NOUN, VERB
Live	If you live in New Zealand, you can sometimes see the All Blacks play live	Wenn du in Neuseeland lebst, kannst du die All Blacks manchmal live spielen sehen	VERB, ADV	VERB, ADV	VERB, ADV	VERB, ADV	VERB, ADJ
Contract	She signed a contract not to contract the flu.	Sie hat einen Vertrag unterschrieben, dass es ihr nicht erlaubt ist, sich mit der Grippe anzustecken.	NOUN, VERB	NOUN, VERB	NOUN, VERB	NOUN, VERB	NOUN, VERB
Kind	This kind of people are kind by nature.	Diese Art von Menschen ist von Natur aus freundlich.	NOUN, ADJ	NOUN, NOUN	NOUN, ADJ	NOUN, ADJ	NOUN, NOUN
Project	The manager asks his employee to project the project agenda on the screen.	Der Manager bittet seinen Mitarbeiter, die Projektagenda auf die Leinwand zu projizieren.	VERB, NOUN	VERB, NOUN	VERB, NOUN	VERB, NOUN	NOUN, NOUN

Subject	Today's subject is whether a foreign company is subject to our environmental laws.	Das heutige Thema ist die Frage, ob ein ausländisches Unternehmen unserem Umweltrecht unterliegt.	NOUN, ADJ	NOUN, ADJ	NOUN, ADJ	NOUN, ADJ	NOUN; ADJ

Anhang 5: Klassifikations-Reports Deutsch

DistilBERT

	precision	recall	f1-score	support
ADJ	0.78	0.93	0.85	1029
ADP	0.99	0.99	0.99	1587
ADV	0.92	0.81	0.86	1255
AUX	0.94	0.96	0.95	678
CCONJ	0.96	0.94	0.95	453
DET	0.97	0.98	0.98	2007
INTJ	0.00	0.00	0.00	3
NOUN	0.94	0.95	0.95	3086
NUM	0.91	0.97	0.94	236
PART	0.97	0.95	0.96	205
PRON	0.93	0.90	0.92	896
PROPN	0.86	0.83	0.85	1009
SCONJ	0.95	0.87	0.90	159
VERB	0.96	0.94	0.95	1307
X	0.18	0.33	0.24	21
_	1.00	0.99	1.00	273
accuracy			0.93	14204
macro avg	0.83	0.83	0.83	14204
weighted avg	0.93	0.93	0.93	14204

NLTK

	precision	recall	f1-score	support
ADJ	0.74	0.88	0.80	1032
ADP	0.97	0.99	0.98	1590
ADV	0.93	0.76	0.84	1262
AUX	0.89	0.95	0.92	679
CCONJ	0.97	0.92	0.95	453
DET	0.96	0.97	0.97	2010
INTJ	0.00	0.00	0.00	3
NOUN	0.95	0.91	0.93	3090
NUM	0.90	0.92	0.91	238
PART	0.97	0.94	0.96	205
PRON	0.94	0.89	0.91	897
PROPN	0.73	0.86	0.79	1012
SCONJ	0.90	0.82	0.86	159
VERB	0.92	0.88	0.90	1311
accuracy			0.91	13941
macro avg	0.84	0.84	0.84	13941
weighted avg	0.91	0.91	0.91	13941

SpaCy

	precision	recall	f1-score	support
ADJ	0.94	0.77	0.85	1030
ADP	0.97	0.99	0.98	1587
ADV	0.83	0.94	0.88	1259
AUX	0.88	0.98	0.93	678
CCONJ	0.98	0.92	0.95	452
DET	0.90	0.99	0.94	2005
INTJ	0.00	0.00	0.00	3
NOUN	0.93	0.97	0.95	3084
NUM	0.86	0.95	0.90	237
PART	0.99	0.97	0.98	205
PRON	0.96	0.77	0.85	896
PROPN	0.91	0.79	0.84	1008
SCONJ	0.90	0.81	0.85	159
VERB	0.96	0.91	0.93	1308
X	0.00	0.00	0.00	0
accuracy			0.92	13911
macro avg	0.80	0.78	0.79	13911
weighted avg	0.92	0.92	0.92	13911

Stanford

	precision	recall	f1-score	support
ADJ	0.77	0.88	0.82	1032
ADP	0.97	0.98	0.97	1590
ADV	0.94	0.81	0.87	1262
AUX	0.86	0.96	0.91	679
CCONJ	0.98	0.91	0.94	453
DET	0.95	0.95	0.95	2010
INTJ	0.00	0.00	0.00	3
NOUN	0.97	0.92	0.95	3090
NUM	0.92	0.93	0.93	238
PART	0.94	0.97	0.95	205
PRON	0.90	0.86	0.88	897
PROPN	0.76	0.94	0.84	1012
SCONJ	0.93	0.79	0.85	159
VERB	0.95	0.87	0.91	1311
X	0.00	0.00	0.00	0
accuracy			0.91	13941
macro avg	0.79	0.79	0.78	13941
weighted avg	0.92	0.91	0.92	13941

Anhang 6: Klassifikations-Reports Französisch

DistilBERT

	precision	recall	f1-score	support
ADJ	0.95	0.94	0.94	2108
ADP	1.00	1.00	1.00	5486
ADV	0.98	0.98	0.98	1209
AUX	0.99	0.90	0.94	1079
CCONJ	0.99	1.00	1.00	886
DET	1.00	1.00	1.00	5279
INTJ	0.14	0.33	0.20	6
NOUN	0.95	0.98	0.96	6458
NUM	0.99	0.99	0.99	870
PRON	0.98	0.87	0.92	1489
PROPN	0.94	0.92	0.93	2507
SCONJ	0.99	0.97	0.98	241
SYM	0.14	0.27	0.18	11
VERB	0.97	0.94	0.96	2674
X	0.14	0.31	0.20	156
_	1.00	1.00	1.00	1007
accuracy			0.96	31466
macro avg	0.82	0.84	0.82	31466
weighted avg	0.97	0.96	0.97	31466

NLTK

	precision	recall	f1-score	support
ADJ	0.84	0.90	0.87	2111
ADP	0.98	0.97	0.98	5489
ADV	0.95	0.84	0.89	1210
AUX	0.95	0.94	0.95	1081
CCONJ	1.00	0.99	0.99	887
DET	0.98	0.98	0.98	5281
INTJ	0.50	0.33	0.40	6
NOUN	0.94	0.95	0.95	6465
NUM	0.98	0.98	0.98	870
PRON	0.98	0.80	0.89	1489
PROPN	0.91	0.95	0.93	2508
SCONJ	0.89	0.87	0.88	241
SYM	1.00	0.09	0.17	11
VERB	0.88	0.92	0.90	2675
X	0.00	0.00	0.00	0
_	0.00	0.00	0.00	0
accuracy			0.94	30324
macro avg	0.80	0.72	0.73	30324
weighted avg	0.95	0.94	0.94	30324

SpaCy

	precision	recall	f1-score	support
ADJ	0.73	0.80	0.76	2111
ADP	0.97	0.93	0.95	5489
ADV	0.86	0.79	0.83	1210
AUX	0.72	0.95	0.82	1081
CCONJ	0.99	0.99	0.99	887
DET	0.98	0.87	0.92	5281
INTJ	0.00	0.00	0.00	6
NOUN	0.79	0.88	0.83	6465
NUM	0.90	0.91	0.91	870
PART	0.00	0.00	0.00	0
PRON	0.91	0.82	0.87	1489
PROPN	0.89	0.72	0.80	2508
SCONJ	0.90	0.79	0.84	241
SYM	0.00	0.00	0.00	11
VERB	0.82	0.88	0.85	2675
X	0.00	0.00	0.00	0
accuracy			0.87	30324
macro avg	0.65	0.65	0.65	30324
weighted avg	0.88	0.87	0.87	30324

Stanford

	precision	recall	f1-score	support
ADJ	0.92	0.90	0.91	2111
ADP	0.98	0.99	0.99	5489
ADV	0.93	0.84	0.89	1210
AUX	0.95	0.97	0.96	1081
CCONJ	1.00	0.95	0.98	887
DET	0.98	0.99	0.98	5281
INTJ	0.57	0.67	0.62	6
NOUN	0.96	0.95	0.95	6465
NUM	0.98	0.99	0.98	870
PRON	0.95	0.78	0.85	1489
PROPN	0.90	0.95	0.92	2508
SCONJ	0.74	0.86	0.79	241
SYM	0.23	0.27	0.25	11
VERB	0.93	0.95	0.94	2675
X	0.00	0.00	0.00	0
accuracy			0.95	30324
macro avg	0.80	0.80	0.80	30324
weighted avg	0.95	0.95	0.95	30324

Anhang 7: Klassifikations-Reports Englisch

DistilBERT

	precision	recall	f1-score	support
ADJ	0.90	0.94	0.92	1265
ADP	0.98	0.96	0.97	1919
ADV	0.88	0.87	0.88	847
AUX	0.98	0.87	0.92	894
CCONJ	0.99	1.00	0.99	664
DET	0.99	0.98	0.99	1647
INTJ	0.42	0.80	0.56	145
NOUN	0.91	0.96	0.93	3372
NUM	0.95	0.98	0.97	332
PART	0.98	0.99	0.99	413
PRON	0.99	0.99	0.99	1380
PROPN	0.96	0.74	0.83	1272
SCONJ	0.92	0.92	0.92	296
SYM	0.00	0.00	0.00	2
VERB	0.93	0.97	0.95	2035
X	0.37	0.27	0.31	26
_	1.00	0.99	1.00	256
accuracy			0.94	16765
macro avg	0.83	0.84	0.83	16765
weighted avg	0.94	0.94	0.94	16765

NLTK

	precision	recall	f1-score	support
ADJ	0.85	0.85	0.85	1268
ADP	0.94	0.97	0.95	1923
ADV	0.87	0.83	0.85	847
AUX	0.96	0.91	0.93	895
CCONJ	0.98	1.00	0.99	665
DET	0.98	0.99	0.98	1649
INTJ	0.90	0.68	0.78	145
NOUN	0.92	0.94	0.93	3376
NUM	0.95	0.96	0.96	334
PART	0.95	0.76	0.85	413
PRON	0.98	0.98	0.98	1380
PROPN	0.92	0.93	0.93	1276
SCONJ	0.83	0.76	0.79	296
SYM	0.00	0.00	0.00	2
VERB	0.90	0.92	0.91	2036
X	0.00	0.00	0.00	0
–	0.00	0.00	0.00	0
accuracy			0.93	16505
macro avg	0.76	0.73	0.75	16505
weighted avg	0.93	0.93	0.93	16505

SpaCy

	precision	recall	f1-score	support
ADJ	0.92	0.89	0.90	1268
ADP	0.93	0.96	0.95	1923
ADV	0.87	0.89	0.88	847
AUX	0.95	0.82	0.88	895
CCONJ	0.99	1.00	1.00	665
DET	0.90	1.00	0.94	1649
INTJ	0.93	0.78	0.85	145
NOUN	0.96	0.94	0.95	3376
NUM	0.98	0.99	0.98	334
PART	0.98	0.96	0.97	413
PRON	0.97	0.87	0.91	1380
PROPN	0.87	0.95	0.91	1276
SCONJ	0.68	0.50	0.57	296
SYM	0.00	0.00	0.00	2
VERB	0.89	0.94	0.91	2036
X	0.00	0.00	0.00	0
accuracy			0.92	16505
macro avg	0.80	0.78	0.79	16505
weighted avg	0.92	0.92	0.92	16505

Stanford

	precision	recall	f1-score	support
ADJ	0.92	0.90	0.91	1268
ADP	0.22	0.04	0.06	1923
ADV	0.77	0.88	0.82	847
AUX	0.99	0.20	0.34	895
CCONJ	0.99	0.99	0.99	665
DET	0.90	1.00	0.94	1649
INTJ	0.95	0.70	0.80	145
NOUN	0.95	0.96	0.96	3376
NUM	0.98	0.98	0.98	334
PART	0.97	0.17	0.29	413
PRON	0.99	0.85	0.92	1380
PROPN	0.85	0.93	0.89	1276
SCONJ	0.11	0.79	0.19	296
SYM	0.00	0.00	0.00	2
VERB	0.72	0.94	0.82	2036
X	0.00	0.00	0.00	0
accuracy			0.77	16505
macro avg	0.71	0.65	0.62	16505
weighted avg	0.81	0.77	0.75	16505

Bisher erschienene Schriften

Ergebnisse von Forschungsprojekten erscheinen jeweils in Form von Arbeitsberichten in Reihen. Sonstige Publikationen erscheinen in Form von alleinstehenden Schriften.

Derzeit gibt es in den Churer Schriften zur Informationswissenschaft folgende Reihen:
Reihe Berufsmarktforschung

Weitere Publikationen

Churer Schriften zur Informationswissenschaft – Schrift 140
Herausgegeben von Wolfgang Semar
Noemi Andres
Status quo des Social-Media-Einsatzes in Schweizer Tambouren-, Clairon- und Pfeifervereinen
Chur, 2021
ISSN 1660-945X

Churer Schriften zur Informationswissenschaft – Schrift 141
Herausgegeben von Wolfgang Semar
Rachel Noëmi Thommen
Lärmmanagement an Deutschschweizer Hochschulbibliotheken
Evaluation der Wahrnehmung des Geräuschpegels von Studierenden in Hochschulbibliotheken und Einfluss von Covid-19
Chur, 2021
ISSN 1660-945X

Churer Schriften zur Informationswissenschaft – Schrift 142
Herausgegeben von Wolfgang Semar
Daria Gloor
Berichterstattung von CO_2-Emissionen im Scope 3 des GHG Protocol
Eine Fallstudie zur Ableitung von digitalen Best Practices für Unternehmen zur Messung und Angabe von CO_2-Emissionen der Kriterien im Scope 3
Chur, 2022
ISSN 1660-945X

Churer Schriften zur Informationswissenschaft – Schrift 143
Herausgegeben von Wolfgang Semar
Leonardo Personini
What role have academic libraries and librarians had in the fight against the COVID-19 pandemic?
Chur, 2022
ISSN 1660-945X

Churer Schriften zur Informationswissenschaft – Schrift 144
Herausgegeben von Wolfgang Semar
Jasmin Suter
TikTok User sind einfacher manipulierbar
Einfluss von Videoplattformen auf das Verhalten in der Pre-Purchase Phase am Beispiel TikTok
Chur, 2022
ISSN 1660-945X

Churer Schriften zur Informationswissenschaft – Schrift 145
Herausgegeben von Wolfgang Semar
Lea Bächli
Die Veränderungen der Angebote öffentlicher Bibliotheken in der Deutschschweiz durch die COVID-19-Pandemie
Chur, 2022
ISSN 1660-945X

Churer Schriften zur Informationswissenschaft – Schrift 146
Herausgegeben von Wolfgang Semar
Jeffrey Santana de Jesus
Mithilfe von Digital Nudging mehr Privatsphäre in sozialen Netzwerken?
Digital Nudging in sozialen Netzwerken
Chur, 2022
ISSN 1660-945X

Churer Schriften zur Informationswissenschaft – Schrift 147
Herausgegeben von Wolfgang Semar
Regina Eicher
Die Entwicklung inhaltlicher Sprachbegriffe für eine verbesserte Erschliessung von Kinder- und Jugendzeichnungen
Eine qualitative Inhaltsanalyse von 12 ausgewählten Märchen
Chur, 2022
ISSN 1660-945X

Churer Schriften zur Informationswissenschaft – Schrift 148
Herausgegeben von Wolfgang Semar
Andrej Kilian
"Die Bibliotheksthematik hat sich in den letzten Jahren stark relativiert"
Interne Bibliotheken in der Deutschschweiz und in Lichtenstein – Versuch eines Einblicks
Chur, 2022
ISSN 1660-945X

Churer Schriften zur Informationswissenschaft – Schrift 149
Herausgegeben von Wolfgang Semar
Sandra Freiburghaus
Untersuchung von Anzeige- und Reservationssystemen zur Lernplatzorganisation in Bibliotheken
Chur, 2022
ISSN 1660-945X

Churer Schriften zur Informationswissenschaft – Schrift 150
Herausgegeben von Wolfgang Semar
Nicole Fässler
User Adoption bei der Einführung einer Kollaborations- und Kommunikationssoftware im Modern Workplace Umfeld
Chur, 2022
ISSN 1660-945X

Churer Schriften zur Informationswissenschaft – Schrift 151
Herausgegeben von Wolfgang Semar
Marina Inglin
Re- und Upskilling-Empfehlung. Kriterien für die automatische Auswahl von Re- und Upskilling-Angebote
Chur, 2022
ISSN 1660-945X

Churer Schriften zur Informationswissenschaft – Schrift 152
Herausgegeben von Wolfgang Semar
Lisa Heller
Zur Genese eines nationalen Bibliotheksprojekts: Swiss Library Service Platform (SLSP)
Chur, 2022
ISSN 1660-945X

Churer Schriften zur Informationswissenschaft – Schrift 153
Herausgegeben von Wolfgang Semar
Antonin Friberg
Die Effektivität von Social Media Norms Nudging in der Customer Journey
Chur, 2022
ISSN 1660-945X

Churer Schriften zur Informationswissenschaft – Schrift 154
Herausgegeben von Wolfgang Semar
Curdin Marxer
«Drug Repurposing» Wie können unstrukturierte Textdaten für die Ermittlung neuer «Drug Repurposing» nutzbar gemacht werden und wie können sie Datenbanken ergänzen?
Chur, 2022
ISSN 1660-945X

Churer Schriften zur Informationswissenschaft – Schrift 155
Herausgegeben von Wolfgang Semar
Samir Limani
Sicht der administrativen Mitarbeitenden von Bündner Spitälern und Kliniken auf den Digitalisierungsstand ihres Unternehmens
Chur, 2022
ISSN 1660-945X

Über die Informationswissenschaft der Fachhochschule Graubünden

Die Informationswissenschaft ist in der Schweiz noch ein relativ junger Lehr- und Forschungsbereich. International weist diese Disziplin aber vor allem im anglo-ameri-kanischen Bereich eine jahrzehntelange Tradition auf. Die klassischen Bezeichnungen dort sind Information Science, Library Science oder Information Studies. Die Grundfragestellung der Informationswissenschaft liegt in der Betrachtung der Rolle und des Umgangs mit Information in allen ihren Ausprägungen und Medien sowohl in Wirtschaft und Gesellschaft. Die Informationswissenschaft wird in Chur integriert betrachtet.

Diese Sicht umfasst nicht nur die Teildisziplinen Bibliothekswissenschaft, Archiv-wissenschaft und Dokumentationswissenschaft. Auch neue Entwicklungen im Bereich Medienwirtschaft, Informations- und Wissensmanagement und Big Data werden gezielt aufgegriffen und im Lehr- und Forschungsprogramm berücksichtigt.

Der Studiengang Informationswissenschaft wird seit 1998 als Vollzeitstudiengang in Chur angeboten und seit 2002 als Teilzeit-Studiengang in Zürich. Seit 2010 rundet der Master of Science in Business Administration das Lehrangebot ab.

Der Arbeitsbereich Informationswissenschaft vereinigt Cluster von Forschungs-, Entwicklungs- und Dienstleistungspotenzialen in unterschiedlichen Kompetenzzentren:

- Information Management & Competitive Intelligence
- Collaborative Knowledge Management
- Information and Data Management
- Records Management
- Library Consulting
- Information Laboratory
- Digital Education

Diese Kompetenzzentren werden im Swiss Institute for Information Science (SII) zusammengefasst.

Impressum

Impressum

FHGR - Fachhochschule
Graubünden
Information Science
Pulvermühlestrasse 57
CH-7000 Chur

www.informationscience.ch
www.fhgr.ch
ISSN 1660-945X

Institutsleitung

Prof. Dr. Ingo Barkow
Telefon: +41 81 286 24 61
Email: ingo.barkow@fhgr.ch

Sekretariat

Telefon: +41 81 286 24 24
Fax: +41 81 286 24 00
Email: clarita.decurtins@fhgr.ch