Technische Hochschule Ingolstadt

Fakultät Informatik

Bachelorstudiengang Künstliche Intelligenz

Named Entity Recognition in Sporttweets

Seminararbeit

Abdullah Koukash

Betreuung: Steffen Freisinger

Datum: 28. Juni 2023

Zusammenfassung

Diese Seminararbeit stellt die Anwendung der Named Entity Recognition (NER) in Sport-Tweets unter Verwendung der SpaCy-Bibliothek vor. Die Arbeit beschreibt den Prozess des Datenlabelings, der Datenanalyse, des Trainings des Modells mit SpaCy und der Bewertung seiner Leistung. Die gelabelten Daten wurden mithilfe des Tools Label Studio vorbereitet, das den Annotierungsprozess erleichterte. Die Datenanalyse lieferte Einblicke in die Häufigkeit und Verteilung der Entitätstypen im Datensatz. Die SpaCy-Bibliothek, bekannt für ihre Geschwindigkeit und Effizienz, wurde für das Training des NER-Modells verwendet. Das vortrainierte Modell **en_core_web_sm** wurde mithilfe von benutzerdefinierten Trainingsdaten feinabgestimmt. Die Evaluation des Modells ergab beeindruckende Ergebnisse mit hoher Präzision, Recall und F1-Score, was auf seine Fähigkeit hinweist, genaue Vorhersagen zu treffen und relevante Entitäten zu identifizieren. Es wurden jedoch potenzielle Einschränkungen und Verbesserungsmöglichkeiten diskutiert, wie beispielsweise der Bedarf an vielfältigeren Trainingsdaten und der Umgang mit längeren Tweets. Insgesamt zeigen die Ergebnisse vielversprechende Resultate, erfordern jedoch weitere Forschung und Bewertung, um die Leistung des Modells zu verbessern.

Inhaltsverzeichnis

1 Einleitung

1.1 Hintergrund und Motivation

Mit der zunehmenden Verbreitung sozialer Medien und der wachsenden Anzahl von Nutzern, die Sportereignisse online diskutieren, wird die automatische Extraktion von Informationen aus Tweets über Sportereignisse immer wichtiger. Eine Methode, die dafür verwendet werden kann, ist die Named Entity Recognition (NER) [3], die das Erkennen und Kategorisieren von Entitäten wie Spielern, Mannschaften, Schiedsrichtern und Veranstaltungsorten in Tweets ermöglicht. Die Verwendung von NER in diesem Kontext kann dazu beitragen, Einblicke in Trends und Entwicklungen im Zusammenhang mit bestimmten Sportereignissen zu gewinnen.
Obwohl NER ein vielversprechender Ansatz für die automatische Textanalyse von Tweets über Sport ist, gibt es immer noch Herausforderungen bei der Entwicklung von effektiven NER-Systemen. Diese Herausforderungen können sich aus der Vielfalt der Entitätstypen, der Informalität von Tweets und der Notwendigkeit von großen Trainingsdatensätzen ergeben.

1.2 Forschungsfrage und Zielsetzung

In dieser Seminararbeit werde ich das Thema Named Entity Recognition (NER) im Allgemeinen vorstellen und die verschiedenen Ansätze kurz erläutern. Anschließend werde ich die Wirksamkeit eines feinabgestimmten spaCy-Modells anhand eines eigens erstellten Datensatzes von Tweets über Sport (Fußball) vergleichen. Mein Ziel ist es, zu untersuchen, wie gut das Modell Entitäten aus Sporttweets extrahieren kann. Dazu werde ich den Datensatz verwenden, um die NER-Systeme zu trainieren und zu evaluieren.
Die Ergebnisse dieser Arbeit sollen dazu beitragen, das Verständnis für die Anwendung von NER auf Sporttweets zu verbessern und über die Erfahrungen bei der Arbeit mit spaCy in diesem Kontext zu berichten.

2 Grundlagen von NER

2.1 Definition und Anwendungsbereiche von NER

NER steht für Named Entity Recognition, auch als Entity Extraction bezeichnet, und ist eine Technik des maschinellen Lernens, die dazu dient, benannte Entitäten wie Personen, Organisationen, Orte, Datum, Zeit, Mengen und andere spezielle Ausdrücke in einem Text zu erkennen und zu klassifizieren.
Die Anwendungsbereiche von NER sind vielfältig und reichen von der automatischen Klassifizierung von Texten bis hin zur Verbesserung der Suchmaschinenergebnisse. Einige der gängigen Anwendungsbereiche von NER umfassen:

- Informationsextraktion: NER kann verwendet werden, um wichtige Informationen aus großen Textmengen zu extrahieren, beispielsweise um wichtige Daten wie Namen von Personen, Firmen, Adressen oder Telefonnummern zu extrahieren.
- Automatisierte Chatbots: Chatbots können mit NER ausgestattet werden, um die Absichten des Benutzers besser zu verstehen und angemessene Antworten zu geben.

Insgesamt hat NER eine breite Palette von Anwendungsbereichen und kann dazu beitragen, den Prozess der Textanalyse und -verarbeitung zu automatisieren und zu verbessern. Ein Beispiel für die Anwendung von NER zur Erkennung von Datumsangaben könnte folgender Satz sein: »Das Meeting wurde für den 5. April 2023 um 10:00 Uhr geplant.« Hier würde NER erkennen, dass »5. April 2023« ein Datum ist und es entsprechend klassifizieren.

2.2 Arten von benannten Entitäten

NER-Systeme können eine Vielzahl von benannten Entitäten erkennen, darunter nicht nur Personen, Organisationen und Orte, sondern auch viele andere Arten. Es können auch weitere spezifische Entitäten hinzugefügt werden.
In dieser Seminararbeit werde ich die folgenden Entitäten berücksichtigen: Spielername, Trainername, Verein, Stadion, geo. Ort, Nationalität, Datum, Zeitangabe (Uhrzeit).

3 Methoden der NER

3.1 Regelbasierte Methoden

Regelbasierte Methoden sind einfache Techniken, die auf dem Abgleich von Mustern basiert, um benannte Entitäten im Text zu identifizieren. Diese Methode ist jedoch oft ungenau und erfordert eine hohe Anzahl von Regeln. Wichtig zu erwähnen ist, dass die Regeln manuell definiert werden müssen, was diese Methode sehr aufwendig macht.

3.2 Statistische Methoden

Hierbei werden maschinelle Lernverfahren wie Hidden Markov Models (HMMs) [4] oder Conditional Random Fields (CRFs) [5] verwendet, um benannte Entitäten im Text zu identifizieren. Diese Methode ist genauer als regelbasierte Techniken, erfordert jedoch ein umfangreiches manuelles Labeln von Trainingsdaten.

3.3 Deep Learning Methoden

Deep Learning-Methoden wie CNNs [6], RNNs [7] und Transformer wie BERT [8] verwenden tiefe neuronale Netzwerke, um Entitäten zu erkennen. Diese Modelle extrahieren automatisch hierarchische Merkmale und erkennen komplexe Zusammenhänge. Jedoch erfordern sie eine große Menge annotierter Daten und hohe Rechenleistung. Das Training kann zeitaufwändig und hardwareintensiv sein, insbesondere bei tiefen oder umfangreichen Modellen. Trotzdem bieten diese Methoden aufgrund ihrer Fähigkeit, komplexe Zusammenhänge zu erfassen, eine hohe Genauigkeit bei der Entitätenerkennung.

4 SpaCy Architektur

SpaCy ist eine NLP-Bibliothek [9],[10], die einen Mix aus regelbasierten und statistischen Methoden verwendet. Sie nutzt eine neuronale Netzwerkarchitektur namens "Entity Recognizer", die auf dem Modell des "linear chain conditional random field" basiert [11]. Diese Architektur ermöglicht die Erkennung von Entitäten in Texten.
Die zentralen Datenstrukturen in spaCy sind die Klasse Language, die Klasse Vocab und das Objekt Doc [12]. Siehe Abbildung 1. Die "Language"-Klasse wird verwendet, um einen Text zu verarbeiten und ihn in "Doc"-Objekte zu konvertieren. Unter nlp.pipeline wird ausgesucht was für ein Modell trainiert werden sollte (NER oder POS-Tagging etc.). Das

"Vocab" speichert das Vokabular und wandelt die Wörter in Vektoren um, während ein "Example" eine Sammlung von Trainingsannotationen darstellt, die Referenzdaten und Vorhersagen enthält. Ein "Doc" dient als Container für linguistische Annotationen. Ein "Token" repräsentiert ein einzelnes Element wie ein Wort oder Satzzeichen. Ein "Span" ist ein Ausschnitt aus einem "Doc"-Objekt, der aus mehreren Tokens besteht. Das "Lexem" bezieht sich auf einen Worttyp im Vokabular und enthält keine Kontextinformationen. Zum Beispiel hat das Lexem "laufen" verschiedene Wortformen wie "läuft", "lief" und "gelaufen" [12].

Abbildung 1: Library Architecture [12]

5 Datensatz

5.1 Beschreibung des Datensatzes

Ich habe meinem Datensatz den Namen *Football Tweets Dataset* gegeben, kurz *FTD-01*. Der Datensatz umfasst über 200 Tweets zum Thema Fußball. Die überwiegende Mehrheit der Tweets sind aktuelle Tweets, was dazu führt, dass aktuelle Trends oder bestimmte Mannschaften oder Spieler, die zum Zeitpunkt der Erstellung des Datensatzes im Trend waren, häufiger im Datensatz vorkommen.
Der Datensatz *FTD-01* wurde manuell erstellt, indem ich eigenständig Tweets zum Thema Fußball gesucht und ausgewählt habe, die bestimme Kriterien erfüllen. Zu den Kritiren gehörten insbesondere englischsprachige Tweets, die sich auf das Thema Fußball beziehen, gut verständlich und grammatikalisch korrekt sind und mindestens eine der benannten Entitäten enthalten, wie z.B. Mannschaften, Spieler usw. Darüber hinaus habe ich darauf geachtet, dass die Länge der Tweets weder zu kurz noch zu lang ist, um eine einheitliche Datenbasis zu gewährleisten.

Ich habe die Daten bewusst kaum vorverarbeitet, um das Potenzial der Daten für weitere Analysen und Modelle bestmöglich zu erhalten. Die einzige Vorverarbeitung, die ich durchgeführt habe, war das Entfernen der meisten Emojis aus den Tweets. Nur bei einigen Tweets habe ich bewusst die Emojis belassen, um eine gewisse Rauschigkeit in den Daten zu behalten. Das Ziel war es, eine realistische und unverfälschte Darstellung der Twitter-Konversationen im Bereich des Fußballs zu erhalten und mögliche Verzerrungen zu vermeiden, die durch unnötige Vorverarbeitungsschritte verursacht werden könnten.

Außerdem war es notwendig, die Labels der Daten in BIO-Kodierung (Begin-Inner-Outside encoding) zu bringen, um die Entitäten mit mehr als ein Token in Texten zu identifizieren. Ein Beispiel dafür ist die Entität **Real Madrid** was in einem Tweet vorkommt. Um diese Entität zu labeln, können wir die BIO-Kodierung verwenden, wobei *B-Verein* (B= Beginning) für das erste Token **Real** und *I-Verein* (I = Inside) für das zweite Token **Madrid** steht.
Durch die Verwendung der BIO-Kodierung können wir komplexe Entitäten effektiv kennzeichnen, die aus mehreren Tokens bestehen, und gleichzeitig die Tokens, die keiner Entität zugeordnet sind, mit *O* markieren. Es ist wichtig zu beachten, dass es bei der Anwendung

der BIO-Kodierung auf Textdaten auf die korrekte Reihenfolge der Tags ankommt. Wenn beispielsweise die Reihenfolge der Tags *I-Verein* und *B-Verein* vertauscht wird, kann dies zu Fehlern in der Entitätserkennung führen.

5.2 Labeling-Methode

Für den Labelingprozess habe ich das Tool Label Studio [13], [14] verwendet, das eine benutzerfreundliche Oberfläche bietet und die Arbeit sehr erleichtert hat. Mit Label Studio war es einfach, die benötigten Labels und Tags für meine Textdaten zu definieren und zuzuweisen. Zusätzlich bot das Tool verschiedene Exportdateitypen an, die eine nahtlose Integration meiner gelabelten Daten in weitere Schritte meines Workflows ermöglichten. Insgesamt hat Label Studio den Labelingprozess optimiert und meine Arbeit mit den Textdaten erheblich erleichtert. Siehe Abbildung 2.

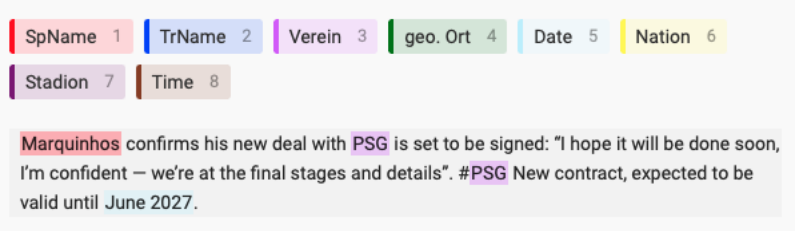

Abbildung 2: Label Studio [14], [**?**]

5.3 Datenanalyse

Als Nächstes wird die Anzahl aller Entitäten je Entitätstyp gezählt. Das gibt uns einen Überblick darüber, welche Entitätstypen in dem Datensatz am häufigsten vorkommen und welche weniger häufig sind. Diese Informationen können uns helfen, die Modellierungsbemühungen zu priorisieren und zu entscheiden, welche Entitätstypen möglicherweise mehr Aufmerksamkeit und Fine Tuning benötigen. Zuallererst habe ich die Anzahl aller Entitäten je Entitätstyp gezählt, siehe Abbildung 3. Dabei habe ich für jeden Entitätstyp, wie zum Beispiel Vereine, die Anzahl für B-Verein und I-Verein zusammengefasst, denn die Zusammenfassung von B- und I-Entitäten innerhalb desselben Entitätstyps ist sinnvoll, da sie beide das gleiche Konzept repräsentieren und somit eine gemeinsame Analyse und Statistik ermöglichen. Anzahl der Entitäten insgesamt ist **929**.

In Abbildung 4 sind die häufigsten Werte je Entitätstyp aufgeführt. Die Tabelle zeigt jeweils die Top 3 Werte für alle Entitätstypen. Die Tabelle wurde zur besseren Übersicht in die Spalten *Entity Label, Top 1 Value, Top 1 count, Top 2 Value, Top 2 count, Top 3 Value und Top 3 count* unterteilt.

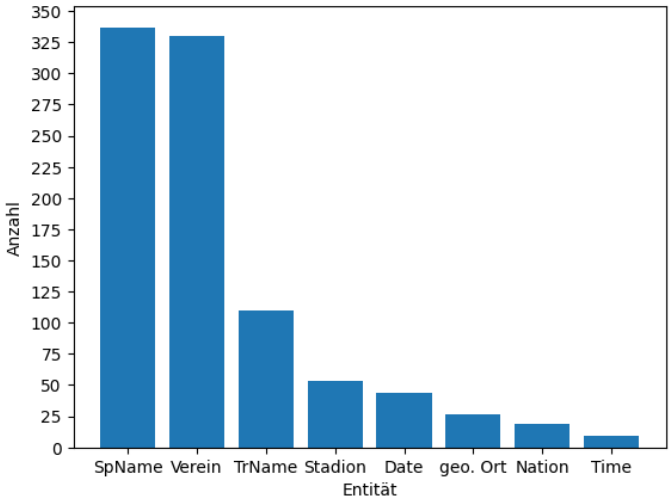

Abbildung 3: Anzahl Entitäten je Entitätstyp

entity_counts

Entity Label	Top 1 value	Top 1 count	Top 2 value	Top 2 count	Top 3 value	Top 3 count
B-SpName	Messi	9	Cristiano	8	Ronaldo	8
I-SpName	Hazard	5	Ronaldo	5	Bale	3
B-Verein	Real	29	Chelsea	22	Arsenal	18
I-Verein	Madrid	28	United	8	Milan	4
B-TrName	Nagelsmann	8	Julian	5	Jose	5
I-TrName	Nagelsmann	4	Mourinho	4	Tuchel	3
B-Stadion	Old	7	Allianz	6	Anfield	4
I-Stadion	Trafford	7	Arena	6	Nou	1
B-Nation	English	4	French	4	Spanish	2
B-geo. Ort	England	4	Paris	2	Munich	2
B-Date	2018	4	June	3	2021	3
I-Date	2027	2	2018	2	2028	1
B-Time	16:30	1	8:30	1	11:45	1
I-Time	PM	1				

Abbildung 4: Top 3 Werte je Entitätstyp

6 Training mit SpaCy

SpaCy ist eine kostenlose Open-Source-Bibliothek für Natural Language Processing in Python. Es bietet NER, Part-of-Speech-Tagging, Abhängigkeitsanalyse, Wortvektoren und mehr [9]. Im Folgenden werden die wichtigsten Aspekte von SpaCy vorgestellt und auf die Named Entity Recognition (NER) in SpaCy sowie das Fine-Tuning eingegangen.

6.1 Wichtige Merkmale von SpaCy [1] [2]

- Schnelle und effiziente Verarbeitung von Textdaten: SpaCy wurde für eine hohe Geschwindigkeit und effiziente Nutzung von Ressourcen entwickelt. Es ist in Cython geschrieben, was es zu einer der schnellsten NLP-Bibliotheken auf dem Markt macht.
- Vorkonfigurierte Modelle: SpaCy verfügt über vorgefertigte Modelle, die für verschiedene Aufgaben wie POS-Tagging, Parsing und Named Entity Recognition optimiert sind. Diese Modelle können auf einfache Weise für spezifische Anwendungsfälle angepasst werden.
- Unterstützung für viele Sprachen: SpaCy unterstützt mehrere Sprachen, darunter Englisch, Deutsch, Spanisch, Portugiesisch, Französisch und Italienisch.
- Integration in andere Tools: SpaCy kann in andere NLP-Tools wie NLTK und Gensim integriert werden und ist auch in der Lage, mit anderen Datenanalyse- und Machine-Learning-Bibliotheken wie Scikit-Learn, TensorFlow und PyTorch zu arbeiten.

6.2 Fine-Tuning in SpaCy

In meiner Seminararbeit habe ich das Modell **en_core_web_sm** [15] verwendet und entsprechend fine-tuned. **en_core_web_sm** ist ein vortrainiertes Modell von spaCy für die englische Sprache. Es bietet NER, POS-Tagging, Abhängigkeitsanalyse und vieles mehr[15]. Im Allgmeinen wird beim Fine-Tuning ein vorgefertigtes Modell mit zusätzlichen Daten trainiert, um es für spezifische Anwendungsfälle zu optimieren. Mit SpaCy besteht die Möglichkeit für Benutzer ein vorgefertigtes Modell mit eigenen Daten für die NER optimieren [16]. Das Fine-Tuning erfolgte in drei Schritten:

- Vorbereitung der Trainingsdaten

- Anpassung des vorgefertigten Modells
- Evaluierung des optimierten Modells

6.3 Training

Die Daten für das Training wurden wie folgt vorbereitet und an das Modell für Fine Tuning weitergegeben:

```
Beispiel-Sample:
("Today we look at striker Robert Lewandowski who plays for FC Barcelona.
What do you think of this player?",
{'entities': [(25, 31, 'B-SpName'), (32, 43, 'I-SpName'),
              (58, 60, 'B-Verein'), (61, 70, 'I-Verein')]})
```

Der Datensatz wurde in zwei Teile aufgeteilt: den Trainingsteil und den Testteil. Der Trainingsteil umfasst 75% des gesamten Datensatzes, was 150 Samples entspricht. Der Testteil besteht aus 25% des Datensatzes, was 50 Samples entspricht.

SpaCy verwendet beim Fine-Tuning den Optimierungsalgorithmus Adam[17]. Anstelle einer spezifischen Verlustfunktion verwendet Spacy kein bestimmtes Loss-Modell [18],[19] . Stattdessen setzt Spacy auf ein iteratives Trainingsverfahren, bei dem das Modell schrittweise optimiert wird. Das NER-Modell von Spacy basiert auf Übergängen und verfolgt ein Imitationslernziel. Um eine detaillierte Beschreibung des Algorithmus zu erhalten, empfiehlt sich das Video [20]. Insbesondere der Abschnitt über strukturierte Vorhersagen bietet eine umfassende Erklärung. Außerdem wäre das Paper [21] für ein besseres Verständnis hilfreich, insbesondere Abschnitt 4. Der Quellcode zu der Berechnung ist unter Quelle [22] verfügbar.
Für die Anpassung habe ich die Lernrate auf 0.001 festgelegt und alle anderen Parameter auf ihren Standardwerten belassen.
Um gute Trainingsergebnisse zu erzielen, habe ich das Modell 15 Epochen trainiert. Nach jeder Epoche wurden die Daten erneut gemischt. Für eine visuelle Darstellung der Trainingsergebnisse siehe Abbildung 5.

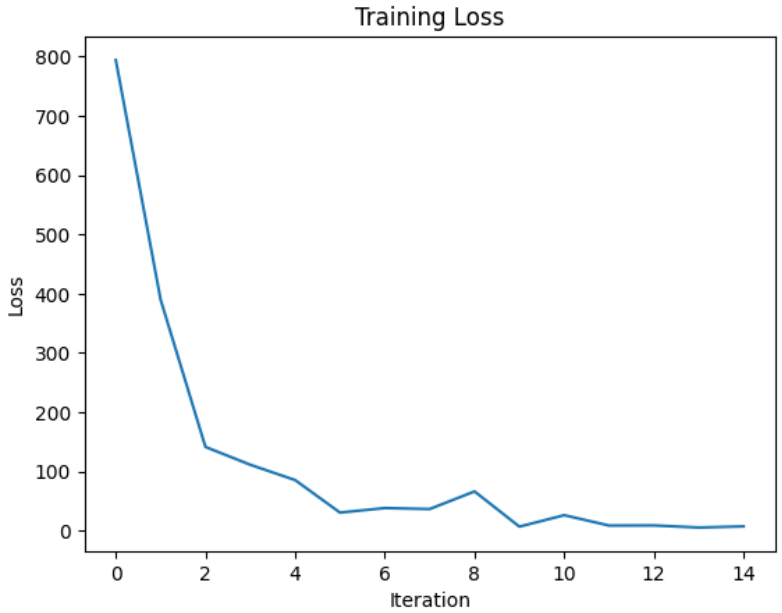

Abbildung 5: Loss Verlauf

6.4 Evaluation

Die Evaluierung des Modells ergab beeindruckende Testergebnisse mit einer Präzision von 0.981, einem Recall von 0.987 und einem F1-Score von 0.984. Diese Metriken zeigen die Fähigkeit des Modells, präzise Vorhersagen zu treffen und eine hohe Vollständigkeit bei der Identifizierung relevanter Ergebnisse zu erreichen. Die hervorragende Leistung des Modells bestätigt seine Effektivität und Zuverlässigkeit bei der Lösung der Aufgabestellung.
Ein hohes Maß an Präzision bedeutet, dass das Modell nur wenige falsch positiv Vorhersagen macht, während ein hoher Recall darauf hinweist, dass das Modell nur wenige falsch negativ Vorhersagen macht. Der F1-Score ist eine gewichtete Kombination aus Präzision und Recall und gibt ein ausgewogenes Maß für die Modellleistung.

Metrik	Wert
Präzision	0.981
Recall	0.987
F1-Score	0.984

Tabelle 1: Testergebnisse

6.5 Diskussion der Ergebnisse

Die Werte für Präzision, Recall und F1-Score liegen sehr nahe beieinander und sind ziemlich hoch. Dies deutet darauf hin, dass das Modell in der Lage ist, sowohl genaue Vorhersagen zu treffen als auch einen Großteil der relevanten Ergebnisse zu identifizieren. Es ist ein positives Zeichen dafür, dass das Modell gut funktioniert und eine ausgewogene Leistung aufweist. Es ist jedoch wichtig, einige mögliche Einschränkungen und Verbesserungsmöglichkeiten zu berücksichtigen.
Jedoch sollte man die Ergebnisse kritisch hinterfragen und mögliche Gründe erörtern. Es könnte durchaus sein, dass die guten Ergebnisse aufgrund der begrenzten Anzahl verschiedener Entitäten und einer möglichen Auswendiglernung (Overfitting) der Entitäten durch das Modell erreicht wurden. Dies könnte bedeuten, dass das Modell spezifische Entitäten gut erkennt, aber möglicherweise Schwierigkeiten hat, neue, unbekannte Entitäten korrekt zu klassifizieren.
Um sicherzustellen, dass das Modell eine allgemeine Fähigkeit zur Erkennung von Entitäten besitzt - also die Fähigkeit, die allgemeinen Muster und Eigenschaften von Entitäten zu verstehen und nicht nur spezifische Beispiele auswendig gelernt hat - ist es ratsam, eine vielfältigere Trainingsdatenmenge einzusetzen, die verschiedene Entitätstypen und Variationen umfasst. Dadurch kann das Modell lernen, Entitäten auf eine breitere und robusterere Weise zu erkennen und zu klassifizieren. Außerdem fehlt mir eine Vergleichsgrundlage. Es wäre hilfreich, das Modell mit anderen ähnlichen Modellen oder Benchmark-Datensätzen zu vergleichen, um die Leistung im Kontext der breiteren Forschungsgemeinschaft zu bewerten. Ein solcher Vergleich könnte dabei helfen, die Leistung des Modells besser einzuschätzen.
Außerdem habe ich bei einigen manuellen Tests festgestellt, dass das Modell anfällig für Fehler bei langen Tweets ist. Es gibt mehrere Gründe, warum das Modell auf lange Tweets fehleranfällig sein kann. Einer der Hauptgründe ist, dass längere Texte eine größere Vielfalt an Ausdrücken, Satzstrukturen und Kontexten enthalten können. Dadurch steigt die Komplexität der Aufgabe für das Modell, da es mehr Informationen verarbeiten und interpretieren muss. Dies kann zu einer erhöhten Unsicherheit bei der Klassifizierung von Entitäten führen. Hier ist ein Beispiel (R: Richtig, F: Falsch):

```
Beispiel-Sample:("Understand Chelsea haven't sent €80m bid for Dusan
Vlahović, as of now.  He's one of many strikers appreciated at the club
but no bid/talks. #FCBayern and Man United remain in the race for
```

```
Vlahović - but still waiting for Juventus decision."),
```

Entität	Typ	R/F
Chelsea	B-Verein	R
Dusan	B-SpName	R
Vlahović	I-SpName	R
FCBayern	B-Verein	R
Man	B-Verein	R
United	I-Verein	R
Vlahović	B-Stadion	F
Juventus	B-Verein	R

Zusammenfassend lässt sich sagen, dass die vorliegenden Ergebnisse vielversprechend sind, aber eine kritische Betrachtung und weitere Untersuchungen erforderlich sind, um die Leistung des Modells umfassend zu bewerten und mögliche Verbesserungen zu identifizieren.

7 Zusammenfassung und Ausblick

7.1 Zusammenfassung der wichtigsten Ergebnisse und Erkenntnisse

In dieser Seminararbeit wurde die Anwendung der Named Entity Recognition (NER) in Sport-Tweets mithilfe der SpaCy-Bibliothek untersucht. Der gesamte Prozess, einschließlich Datenlabeling, Datenanalyse, Modelltraining und Evaluation, wurde durchgeführt. Die Datenlabeling-Phase wurde mithilfe des Tools Label Studio durchgeführt, was den Annotierungsprozess vereinfachte. Die Datenanalyse lieferte Einblicke in die Häufigkeit und Verteilung der Entitätstypen im Datensatz, was wichtig ist, um das Modell angemessen zu trainieren. Das Training des NER-Modells erfolgte mit der SpaCy-Bibliothek, die für ihre Geschwindigkeit und Effizienz bekannt ist. Das vortrainierte Modell **en_core_web_sm** wurde mithilfe von benutzerdefinierten Trainingsdaten verfeinert. Die Evaluation des Modells ergab beeindruckende Ergebnisse mit hoher Präzision, Recall und F1-Score, was darauf hinweist, dass das Modell genaue Vorhersagen treffen und relevante Entitäten identifizieren kann. Allerdings wurden auch potenzielle Einschränkungen und Verbesserungsmöglichkeiten identifiziert. Es besteht ein Bedarf an vielfältigeren Trainingsdaten, um die Leistung des Modells weiter zu verbessern. Darüber hinaus ist der Umgang mit längeren Tweets eine Herausforderung, da das Modell möglicherweise Schwierigkeiten hat, komplexe Zusammenhänge in solchen Texten zu verstehen.
Insgesamt liefern die Ergebnisse vielversprechende Resultate und zeigen das Potenzial der NER in der Analyse von Sporttweets. Es ist jedoch weitere Forschung und Bewertung erforderlich, um die Leistung des Modells zu optimieren und seine Anwendung auf andere Textarten zu untersuchen.

7.2 Ausblick auf weitere Forschungsmöglichkeiten

Trotz der vielversprechenden Ergebnisse bei der Anwendung von finegetunten SpaCy-Modellen auf Sporttweets gibt es noch Raum für weitere Forschung und Verbesserungen. Hier sind einige Bereiche, die in Zukunft erkundet werden könnten:

- Erweiterung des Trainingsdatensatzes: Um die allgemeine Fähigkeit des Modells zur Erkennung von Entitäten zu verbessern, ist es wichtig, einen größeren und vielfältigeren Trainingsdatensatz einzusetzen. Dies könnte verschiedene Sportarten, Ligen und Spieler umfassen, um ein breiteres Spektrum an Entitäten abzudecken.

- Berücksichtigung von Kontext: Die Berücksichtigung des Kontexts in den Tweets kann die Genauigkeit der Entitätserkennung weiter verbessern. Indem man den Zusammenhang zwischen Wörtern und Sätzen analysiert, kann man das Modell dazu bringen, die richtigen Entitäten in bestimmten Kontexten besser zu erkennen.
- Berücksichtigung von Emotionen und Meinungen: Sporttweets enthalten oft Emotionen und Meinungen der Nutzer. Die Integration von Sentiment-Analyse-Techniken kann helfen, die emotionale Bewertung von Entitäten zu verstehen und somit eine tiefere Analyse des Inhalts der Tweets zu ermöglichen.
- Verbesserung der Robustheit gegenüber Rauschen: Tweets enthalten häufig Tippfehler, Abkürzungen und informelle Sprache, was die Entitätserkennung erschweren kann. Durch die Entwicklung von Modellen, die robust gegenüber Rauschen sind und in der Lage, solche Abweichungen zu berücksichtigen, kann die Leistung der Entitätserkennung weiter verbessert werden.
- Integration von Multi-Modalität: Sport-Tweets werden oft mit Bildern, Videos oder anderen Medieninhalten begleitet. Die Integration von Multi-Modalität in die Entitätserkennung kann eine umfassendere und genauere Analyse des Tweet-Inhalts ermöglichen.

Insgesamt bietet die Anwendung der NER-Techniken auf Sport-Tweets ein großes Potenzial für weitere Forschung und Verbesserungen. Durch die Berücksichtigung der genannten Aspekte können zukünftige Studien bzw. Arbeiten zur Entwicklung fortschrittlicherer Modelle beitragen, die eine präzisere und umfassendere Analyse von Sport-Tweets ermöglichen.

Literatur

[1] D.-I. F. S. Luber, "Was ist spacy?" [Online]. Available: https://www.bigdata-insider.de/was-ist-spacy-a-980360/

[2] SpaCy, "Spacy merkmale." [Online]. Available: https://spacy.io/usage/spacy-101

[3] J. Li, A. Sun, J. Han, and C. Li, "A survey on deep learning for named entity recognition," *CoRR*, vol. abs/1812.09449, 2018. [Online]. Available: http://arxiv.org/abs/1812.09449

[4] S. Morwal and N. Jahan, "Named entity recognition using hidden markov model (hmm): An experimental result on hindi, urdu and marathi languages," 2013.

[5] W. Khan, A. Daud, K. Shahzad, T. Amjad, A. Banjar, and H. Fasihuddin, "Named entity recognition using conditional random fields," *Applied Sciences*, vol. 12, no. 13, p. 6391, Jun 2022. [Online]. Available: http://dx.doi.org/10.3390/app12136391

[6] K. O'Shea and R. Nash, "An introduction to convolutional neural networks," *CoRR*, vol. abs/1511.08458, 2015. [Online]. Available: http://arxiv.org/abs/1511.08458

[7] A. Sherstinsky, "Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network," *CoRR*, vol. abs/1808.03314, 2018. [Online]. Available: http://arxiv.org/abs/1808.03314

[8] H. Darji, J. Mitrović , and M. Granitzer, "German BERT model for legal named entity recognition," in *Proceedings of the 15th International Conference on Agents and Artificial Intelligence*. SCITEPRESS - Science and Technology Publications, 2023. [Online]. Available: https://doi.org/10.5220%2F0011749400003393

[9] M. Honnibal, I. Montani, S. Van Landeghem, and A. Boyd, "spaCy: Industrial-strength Natural Language Processing in Python," 2020.

[10] SpaCy. [Online]. Available: https://spacy.io

[11] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer, "Neural architectures for named entity recognition," *CoRR*, vol. abs/1603.01360, 2016. [Online]. Available: http://arxiv.org/abs/1603.01360

[12] SpaCy, "Spacy api." [Online]. Available: https://spacy.io/api

[13] M. Tkachenko, M. Malyuk, A. Holmanyuk, and N. Liubimov, "Label Studio: Data labeling software," 2020-2022, open source software available from https://github.com/heartexlabs/label-studio. [Online]. Available: https://github.com/heartexlabs/label-studio

[14] Labelstudio, "Labelstudio-ner." [Online]. Available: https://labelstud.io/templates/named_entity.html

[15] SpaCy, "Spacy models." [Online]. Available: https://spacy.io/models/en

[16] ——, "Spacy embeddings-transformers." [Online]. Available: https://spacy.io/usage/embeddings-transformers/

[17] ——, "Spacy training." [Online]. Available: https://spacy.io/usage/training

[18] ——, "Spacy quellcode github." [Online]. Available: https://github.com/explosion/spaCy/blob/v2.3.x/spacy/syntax/nn_parser.pyx#L566

[19] ——, "Spacy entityrecognizer loss." [Online]. Available: https://spacy.io/api/entityrecognizer#get_loss

[20] Explosion, "loss in spacy." [Online]. Available: https://www.youtube.com/watch?v=sqDHBH9IjRU

[21] E. Charniak and M. Johnson, "Coarse-to-fine n-best parsing and MaxEnt discriminative reranking," in *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*. Ann Arbor, Michigan: Association for Computational Linguistics, Jun. 2005, pp. 173–180. [Online]. Available: https://aclanthology.org/P05-1022

[22] SpaCy, "Spacy quellcode github 2." [Online]. Available: https://github.com/explosion/spaCy/blob/0367f864fe90dfa1dcdd0bfaf8f06dbcd5e97e45/spacy/syntax/_parser_model.pyx#L153

Eidesstattliche Erklärung

Ich versichere, dass ich die Arbeit ohne fremde Hilfe und ohne Benutzung anderer als der angegebenen Quellen angefertigt habe und dass die Arbeit in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegen hat und von dieser als Teil einer Prüfungsleistung angenommen wurde. Alle Ausführungen, die wörtlich oder sinngemäß übernommen wurden, sind als solche gekennzeichnet.

Ingolstadt, 26. Juni 2023

Unterschrift