(19) SCHWEIZERISCHE EIDGENOSSENSCHAFT
EIDGENÖSSISCHES INSTITUT FÜR GEISTIGES EIGENTUM

(11) CH 718 451 A2

(51) Int. Cl.: G06F 40/205 (2020.01)
G06F 40/284 (2020.01)
G06N 3/08 (2006.01)

Patentanmeldung für die Schweiz und Liechtenstein
Schweizerisch-liechtensteinischer Patentschutzvertrag vom 22. Dezember 1978

(12) PATENTANMELDUNG

(21) Anmeldenummer: 00280/21

(22) Anmeldedatum: 16.03.2021

(43) Anmeldung veröffentlicht: 30.09.2022

(71) Anmelder:
ELLA Media AG, Gotthardstrasse 28
6300 Zug (CH)

(72) Erfinder:
Katharina Wilbring, 50733 Köln (DE)
Andreas Funke, 40589 Düsseldorf (DE)
Nasrin Saef, 50670 Köln (DE)

(74) Vertreter:
Weinmann Zimmerli AG, Apollostrasse 2
8032 Zürich (CH)

(54) Verfahren zur Generierung von Dateien.

(57) Die Erfindung betrifft ein Verfahren zur Generierung von Dateien, insbesondere von Text- und Audiodateien sowie Dateien für Computerspiele oder Videos. Dabei wird die Verarbeitung sehr großer Mengen inhaltlich und strukturell unterschiedlicher Texte ermöglicht. Hierzu werden in einem ersten Schritt vorhandene Daten gefiltert und aufbereitet, und nachfolgend wird aus den aufbereiteten Daten ein im Hinblick auf die gewünschten Ergebnisse abgestimmtes Trainingskorpus erzeugt. Die gefilterten und aufbereiteten Daten werden normalisiert.

Beschreibung

[0001] Die Erfindung betrifft ein Verfahren zur Generierung von Dateien, insbesondere von Text- und Audiodateien sowie Dateien für Computerspiele und Videos

[0002] Die automatische Erstellung von Texten ist bekannt. So werden zum Beispiel Sport- oder Wetterberichte, Börsennachrichten oder Staumeldungen automatisiert erzeugt. Nachrichten-Portale setzen im Content-Mix zunehmend auf Roboterjournalismus und Online-Shops nutzen computergenerierte Texte zur Bewerbung von Produkten. Smarte Anwendungen kreieren automatisiert individuelle Berichte wie Geschäftsberichte, Immobilien-Exposes oder Fondsreports. Hierbei ist KI-basierte Software in der Lage, sprachliche Inhalte effizient zu generieren und in Echtzeit bereitzustellen.

[0003] Eine notwendige Bedingung für die Erstellung der oben aufgezählten automatisch generierten Textsorten sind strukturierte Daten, d. h. Informationen in Tabellenform, bei Wetterberichten zum Beispiel die Informationen zu Temperatur, Luftdruck oder Niederschlagswahrscheinlichkeit an einem Ort. Für menschliche Leser sind solche reinen Zahlenkolonnen weniger eingängig lesbar als Fliesstext.

[0004] Zur Verbesserung des Nutzererlebnisses übersetzt eine spezielle Software die Daten in Texte. Dabei werden die Informationen in inhaltlich relevante Bestandteile einer intelligenten Textvorlage eingefügt. Solche computergenerierten Beiträge auf Basis intelligenter Templates kommen in den Online-Auftritten der Leitmedien oder auch in lokalen Portalen zum Einsatz. Eine weitere Methode zur Textgenerierung ist die Erzeugung von Text über Language Models, welche mit Hilfe eines Neuronalen Netzes auf Grundlage eines grossen Trainingskorpus' ein Modell von Sprache lernen. Ausgehend von diesem gelernten Modell können sie dem Trainingskorpus' ähnliche Texte generieren. Solche Language Modelle schreiben jedoch standardmässig nur einen Anfang weiter.

[0005] Die Aufgabe der Erfindung besteht nun darin, den Anwendungsbereich zu erweitern und ein Verfahren zur Generierung von Dateien, insbesondere von Text- und Audiodateien sowie Dateien für Computerspiele und Videos zu entwickeln, das die Verarbeitung wesentlich grösserer und unterschiedlicher Datenmengen ermöglicht, als es die sehr spezialisierten Template-basierten Verfahren zulassen.

[0006] Die Aufgabe ist mit den Merkmalen des Patentanspruchs 1 gelöst.

[0007] Es werden vorhandene Daten (Textdaten) über eine Preprocessing-Pipeline gefiltert, aufbereitet (gesäubert) und physikalisch normalisiert. Aus den aufbereiteten Daten wird ein im Hinblick auf die gewünschten Ergebnisse abgestimmtes Trainingskorpus erzeugt und das Training wird nach vorgegebenen Parametern gestartet. Aus den trainierten Modellen wird eine Datei, zum Beispiel ein Text, generiert, wobei die Bereitstellung über unterschiedliche APIs (Application Programming Interface) erfolgt.

[0008] Eine Normalisierung ist ein technischer Vorgang, wie er zum Beispiel aus dem Audiobereich (vergrössern oder verkleinern von Amplituden) oder bei der Herstellung von Stahl zur Erzeugung feinkörniger Strukturen bekannt ist.

[0009] Vorteilhafte Ausgestaltungen der Erfindung sind in den abhängigen Ansprüchen offenbart.

[0010] Die gefilterten und aufbereiteten (Roh-)Daten werden in einzelnen Token bzw. Ziffern dargestellt und verglichen bzw. verarbeitet. Dies unter Beibehaltung des Informationsgehalts der ursprünglichen Darstellung der Ziffernfolgen.

[0011] Die Erfindung wird nachfolgend in einem Ausführungsbeispiel näher beschrieben.

[0012] Beim Preprocessing werden Dateien, im Beispiel Texte, nach sprachlichen und inhaltlichen Parametern gefiltert, gesäubert und normalisiert. Dabei werden unter anderem Outlier-Texte, welche sprachlich stark vom Durchschnitt abweichen (z.B. durch äußerst viele Funktionswörter ohne inhaltliche Bedeutung) entfernt. Ausserdem werden unerwünschte bzw. unzulässige Inhalte entfernt. Die Texte werden gesäubert, indem Anmerkungen der Autoren, stilistische Eigenheiten (beispielsweise Formatierungen oder unübliche sprachliche Varianten), Zeichen aus nicht-lateinischen Alphabeten und gängige Rechtschreibfehler entfernt bzw. korrigiert werden. Sonderzeichen wie Anführungszeichen oder Trennlinien, aber auch für neuronale Netze schwer zu verarbeitende Ziffernfolgen, werden vereinheitlicht. Sentence Splitting und Tokenisierung - also das Markieren von Sätzen und Worteinheiten - werden vorgenommen.

[0013] Sentence Splitting erfolgt über die Library spacy, mit angelegten Zusatzregeln zum Parsen normalisierter Symbole für Absätze, Anführungszeichen etc. Das Tokenisieren, Säubern,

[0014] Filtern und Normalisieren basiert komplett auf eigenen Regeln und regulären Ausdrücken und resultiert in einem ausserordentlich sauberen und homogenen Korpus. Die hohe Standardisierung und das gründliche Säubern führen zu messbar besserem Textoutput der mit diesen Texten trainierten Modelle.

[0015] Der CorpusComposer (Modul) stellt aus einem vorverarbeiteten Textkorpus ein Trainingskorpus zusammen. Die zu berücksichtigenden Texte können nach Metadaten, wie
z. B. der Textlänge, gefiltert werden. Es werden Anpassungen an das zu trainierende Modell vorgenommen, wie z. B. das Ersetzen nicht vorgesehener Sonderzeichen oder Subword Encoding. Weiterhin können die gewünschten Eingabesequenzen festgelegt und Control-Token vorangestellt werden. Während bisherige Language Models für gewöhnlich nur einen Eingabesatz fortschreiben, wird erfindungsgemäss zusätzlich die Eingabe von Prompts, Titeln, Zusammenfassungen, sowie Endsätzen ermöglicht. Control-Token steuern beispielsweise die Textlänge oder das Genre.

[0016] Für Subword Eencoding sowie für die Binarisierung von Korpora zwecks Trainings von sequence-to-sequence Modellen werden die Librarys transformers und fairseq herangezogen. Das Erstellen eines Trainingskorpus basiert ansonsten auf eigenen Filtern und Verarbeitungsregeln. Durch den CorpusComposer ist das Zusammenstellen eines Trainingskorpus' komfortabel konfigurierbar und das Ergebnis reproduzierbar.

[0017] Das Training erfolgt über die bereits genannten Librarys transformers (Language Models) und fairseq (sequence to sequence-Modelle). Das Storymodel enthält fertige Shell-Skripte, die an das zu startende Training angepasst werden können (Pfade, Hyperparameter).

[0018] Für die Generierung ist das Modul StoryGenerator zuständig. Hier werden Kontexte sowie Vorgaben zu Story-Eigenschaften mitgegeben und damit Storys erzeugt.

[0019] Die Grundlage dafür ist eine erweiterte Textgenerierung aus den Librarys transformers bzw. fairseq. Die zusätzlichen Decoder-Features umfassen u.a. eine Blocklist, die die Generierung bestimmter Wörter oder Wortfolgen verhindert. Es kann zwischen einer basalen und einer erweiterten Blocklist ausgewählt werden, außerdem können modellspezifische Blocklists eingebunden werden. Weiterhin ist ein Reduzieren der Häufigkeit von Wiederholungen und direkter Rede sowie die Vorgabe der gewünschten Stimmungslage der Story möglich. Zudem kann dort die Spannungskurve beeinflusst werden. Auch sind die Figurennamen bei einigen Modellen konfigurierbar. Diese Modelle wurden auf generalisierten Varianten unserer Korpora trainiert, die Entity-Klassen (z. B. Protagonist, Love Interest oder Person) statt individueller Namen enthalten. Sie können nach der Generierung mit Wunschnamen ersetzt werden. Durch die zusätzlichen Features werden viele bekannte Probleme von maschinell generiertem Text vermieden und es wird eine hohe Baseline-Qualität beim sprachlichen Output erzielt. Zudem ermöglichen sie mehr Kontrolle über die generierten Storys.

[0020] Um dem Problem zu begegnen, dass neuronale Netze Blackboxes mit schwer zu beeinflussendem Output sind, werden Regeln und ausformuliertes Wissen eingesetzt. Dies verbessert das Lernen von Figuren, indem die Rollen analysiert und in allen Storys gleiche Platzhalter ins Training gegeben werden (anstelle von Namen). Die Spannungskurve in Trainingsstorys wird analysiert und dazu werden Informationen ins Training gegeben, um bei der Generierung die Spannung zu beeinflussen. Das im Modell implizit vorhandene Weltwissen wird erweitert, indem tabellarisches Wissen und Beispiele, wie dieses vom Modell abgerufen werden kann, in die Trainingsdaten integriert werden. Ebenfalls soll explizit formuliertes Wissen aus Wissensgraphen in Training oder Decoder eingebunden werden, so dass einerseits faktisch korrekte Texte generiert werden, und andererseits über Story-Graphen auch Einfluss auf für eine Story (oder ein Modell) relevante Fakten genommen werden kann, wie z. B. Besonderheiten des Settings.

[0021] Semi-automatisierte Tests werden vor jedem Upgrade der API ausgeführt und stellen sicher, dass die neue Version wie erwartet funktioniert.

[0022] Die generierten Texte werden über format-individuelle technische Schnittstellen für die automatische Erzeugung von weiteren Unterhaltungsformen genutzt. Hierbei können insbesondere die Schnittstellen text-to-video und text-to-game entwickelt werden.

[0023] Das Vorhandensein einer gut dokumentierten Programmierschnittstelle (API) ist ein erheblicher Vorteil für ein Software- oder ein die Software enthaltendes Hardware-Produkt, da auf diese Weise Nutzer in die Lage versetzt werden, zusätzliche Software für dieses System zu erstellen. Dadurch wiederum steigt die Attraktivität des Ausgangssystems, zum Beispiel eines Computersystems. Ein weiterer Faktor ist die Langzeit-Stabilität der API.

[0024] Entwickelt wurde eine Storytelling-AI, die zum Beispiel englische Kurzgeschichten im Umfang von bis zu 1.000 Token schreibt.

[0025] Das Trainingskorpus der Anmelderin enthält ca. 700.000 Kurzgeschichten mit je maximal 1.000 Token, die normalisiert und nach sprachlichen und inhaltlichen Parametern gefiltert sind. Dabei werden unter anderem Outlier-Texte sowie pornografische, gewalttätige, rassistische und stark politische Inhalte entfernt. Die Geschichten werden gesäubert, indem Autorenkommentare, stilistische Eigenheiten, Zeichen aus nicht-lateinischen Alphabeten und gängige Rechtschreibfehler entfernt bzw. korrigiert werden. Sonderzeichen werden vereinheitlicht.

[0026] Der Anwender kann darüber hinaus auch einen Prompt, einen Titel, eine Zusammenfassung sowie das Ende als Kontext vorgeben. Die gewünschte Textlänge wird über beim Training gelernte Kontroll-Token bestimmt. Darüber hinaus ist die Sampling-Architektur um eigene Algorithmen erweitert worden, mit denen unerwünschte Wortfolgen über eine Blocklist verboten werden, Wiederholungen gezielter reduziert sowie die Häufigkeit direkter Rede und die gewünschte Stimmung (über Sentiment-Scores) gesteuert werden. Bei einigen Modellen können ausserdem Personennamen vorgegeben werden, da sie auf generalisierten Varianten der Korpora trainiert wurden, die Entity-Klassen statt individueller Namen enthalten. Diese können nach der Generierung zu Wunschnamen de-anonymisiert werden.

[0027] Auf Basis regelmäßig erhobener menschlicher Annotationen entwickelte die Anmelderin eine automatisierte QA, um mithilfe eigener Metriken und Diskriminatoren die generierten Geschichten vorzufiltern sowie den Prozess der Modellselektion zu beschleunigen.

Patentansprüche

1. Verfahren zur Generierung von Dateien, insbesondere von Text- und Audiodateien sowie Dateien für Computerspiele oder Videos, wobei in einem ersten Schritt vorhandene Daten gefiltert und aufbereitet, gesäubert, werden und nachfolgend aus den aufbereiteten Daten ein, im Hinblick auf die gewünschten/angestrebten Ergebnisse, abgestimmtes Trainingskorpus erzeugt wird, dadurch gekennzeichnet, dass die gefilterten und aufbereiteten Daten normalisiert werden.
2. Verfahren nach Anspruch 1, dadurch gekennzeichnet, dass die Daten für die Normalisierung in einzelnen Token bzw. Ziffern zerlegt, dargestellt und verglichen sowie verarbeitet werden.
3. Verfahren nach Anspruch 1 oder 2, dadurch gekennzeichnet, dass mittels der normalisierten Daten neue Daten aus dem Training erzeugt werden.
4. Verfahren nach einem der Ansprüche 1 bis 3, dadurch gekennzeichnet, dass die Dateien Texte, Audioformate oder Videos sind.
5. Verfahren nach einem der Ansprüche 1 bis 4, dadurch gekennzeichnet, dass die Daten- und Dateierzeugung über mindestens eine API erfolgt.