(12) NACH DEM VERTRAG ÜBER DIE INTERNATIONALE ZUSAMMENARBEIT AUF DEM GEBIET DES PATENTWESENS (PCT) VERÖFFENTLICHTE INTERNATIONALE ANMELDUNG

(19) Weltorganisation für geistiges Eigentum
Internationales Büro

(43) Internationales Veröffentlichungsdatum
08. Oktober 2020 (08.10.2020)

WIPO | PCT

(10) Internationale Veröffentlichungsnummer
WO 2020/201249 A1

(51) Internationale Patentklassifikation:
G06F 16/31 (2019.01)

(21) Internationales Aktenzeichen: PCT/EP2020/059042

(22) Internationales Anmeldedatum:
31. März 2020 (31.03.2020)

(25) Einreichungssprache: Deutsch

(26) Veröffentlichungssprache: Deutsch

(30) Angaben zur Priorität:
10 2019 108 858.5
04. April 2019 (04.04.2019) DE

(71) Anmelder: **BUNDESDRUCKEREI GMBH** [DE/DE]; Kommandantenstraße 18, 10969 Berlin (DE).

(72) Erfinder: **WILKE, Andreas**; Ziekowstr. 139, 13509 Berlin (DE). **KOMAROV, Ilya**; Am Tegeler Hafen 36, 13507 Berlin (DE). **PALM, Peter**; Tischlerstraße 1a, 30916 Isernhagen (DE). **PAESCHKE, Manfred**; An der Wildbahn 61, 16348 Wandlitz (DE).

(74) Anwalt: **RICHARDT PATENTANWÄLTE PARTG MBB**; Wilhelmstraße 7, 65185 Wiesbaden (DE).

(81) Bestimmungsstaaten *(soweit nicht anders angegeben, für jede verfügbare nationale Schutzrechtsart)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW,

(54) **Title:** MACHINE LEARNING BASED ON TRIGGER DEFINITIONS

(54) **Bezeichnung:** MASCHINELLES LERNEN AUF BASIS VON TRIGGER-DEFINITIONEN

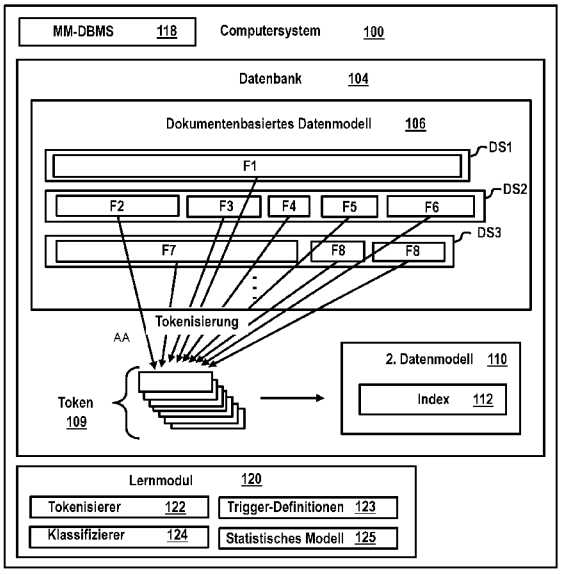

Fig. 1

100 Computer system
104 Database
106 Document-based data model
109 Token
110 Data model
112 Index
120 Learning module
122 Tokeniser
123 Trigger definitions
124 Classifier
125 Statistical model
AA Tokenisation

(57) **Abstract:** The invention relates to a computer-implemented method for machine learning. A pre-trained learning module (120) and a database (104) are provided. The learning module comprises a plurality of predetermined trigger definitions (123), which define triggers (121) for assigning tokens (109) to classes (111) of a group of classes. An additional dataset (108) is received and stored in a first data model (106) of the database, and one or more tokens are generated. Among the generated tokens, triggers are identified and each assigned to the relevant trigger definition. The identified triggers are used to assign further generated tokens to one or more classes in the group of classes. Remaining generated tokens, which have not been assigned to one of the trigger definitions nor assigned to a class due to one of the trigger definitions, are assigned to a fallback class. In a second data model (110), an index (112) is extended using the generated tokens, the class assignments of the generated tokens and a pointer (115) to the stored additional dataset.

(57) **Zusammenfassung:** Die Erfindung betrifft ein computerimpleme ntiertes Verfahren zum maschinellen Lernen. Ein vortrainiertes Lernmodul (120) und eine Datenbank (104) werden bereitgestellt. Das Lernmodul umfasst eine Mehrzahl von vorbestimmten Trigger-Definitionen (123), welche Trigger (121) für ein Zuordnen von Token (109) zu Klassen (111) einer Gruppe von Klassen definieren. Es wird ein zusätzlicher Datensatz (108) empfangen, in einem ersten Datenmodell (106) der Datenbank gespeichert, es werden ein oder mehreren Token erzeugt. Es wer-den unter den erzeugten Token Trigger identifiziert und jeweils der zugrundeliegenden Trigger-Definition zugeordnet. Die identifizierten Trigger werden verwendet, um weitere erzeugte Token zu ein oder mehreren Klassen der Gruppe von Klassen zu-zuordnen. Verbleibenden erzeugte Token, für welche keine Zuordnung zu

WO 2020/201249 A1

[Fortsetzung auf der nächsten Seite]

SA, SC, SD, SE, SG, SK, SL, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, WS, ZA, ZM, ZW.

(84) **Bestimmungsstaaten** *(soweit nicht anders angegeben, für jede verfügbare regionale Schutzrechtsart)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), eurasisches (AM, AZ, BY, KG, KZ, RU, TJ, TM), europäisches (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Veröffentlicht:

— *mit internationalem Recherchenbericht (Artikel 21 Absatz 3)*

— *mit Informationen über die Einbeziehung von fehlenden Teilen und/oder Bestandteilen durch Verweis (Regel 20 Absatz 6)*

einer der Trigger-Definitionen und keine Klassenzuordnung aufgrund einer der Trigger-Definitionen erfolgt ist, werden einer Auffangklasse zugeordnet. In einem zweiten Datenmodell (110) wird ein Index (112) unter Verwendung der erzeugten Token, der Klassenzuordnungen der erzeugten Token und eines Zeigers (115) auf den gespeicherten zusätzlichen Datensatz ergänzt.

Maschinelles Lernen auf Basis von Trigger-Definitionen

Beschreibung

Die Erfindung betrifft ein Verfahren und ein Computersystem zum maschinellen Lernen.

Aus dem Stand der Technik sind Verfahren und Systeme zum maschinellen Lernen System bekannt. Solche Systeme lernen anhand von Beispielen und können diese Beispiele nach Beendigung der Lernphase auf bisher unbekannte Daten anwenden. Die zugrundeliegenden Beispiele stellen Muster und Gesetzmäßigkeiten bereit, welche im Zuge eines Lerntransfers zur Beurteilung bisher unbekannter Daten herangezogen werden.

Bekannte Verfahren und Systeme für maschinelles Lernen arbeiten aufgrund der verwendeten Datenspeicherstrukturen im Allgemeinen nicht auf der gesamten zur Verfügung stehenden Datenmenge. Für das Lernen wird eine Auswahl an Beispielen getroffen, mit welchen das System in der Lernphase trainiert wird. Die aus der beschränkten Auswahl im Zuge des Lernens erfassten Muster und Gesetzmäßigkeiten werden dann sukzessiv auf Teile des restlichen Datenbestands bzw. neuerfasste Daten angewendet.

Der Erfindung liegt die Aufgabe zugrunde, ein verbessertes Verfahren zum maschinellen Lernen zu schaffen.

Die der Erfindung zugrundeliegende Aufgabe wird jeweils mit den Merkmalen der unabhängigen Patentansprüche gelöst. Ausführungsformen der Erfindung sind in den abhängigen Patentansprüchen angegeben.

Ausführungsformen umfassen ein computerimplementiertes Verfahren zum maschinellen Lernen, wobei das Verfahren umfasst:

- Bereitstellen eines vortrainierten Lernmoduls zum maschinellen Lernen, wobei das vortrainierte Lernmodul eine Mehrzahl von vorbestimmten Trigger-Definitionen umfasst, welche Trigger für ein Zuordnen von Token zu Klassen einer ersten Gruppe von Klassen definieren,
- Bereitstellen einer Datenbank, welche von einem Multi-Modell-Datenbankmanagementsystem verwaltet wird, wobei die Datenbank eine Mehrzahl von Datensätze umfasst, welche in einem dokumentenorientierten Datenmodell gespeichert sind, wobei die gespeicherten Datensätze jeweils ein oder mehreren Feldwerte umfassen, wobei die einzelnen Feldwerte der gespeicherten Datensätze jeweils in einem Feld gespeichert sind,

  wobei die Datenbank ferner einen durchsuchbaren Index umfasst, welcher in einem weiteren Datenmodell gespeichert ist, wobei der Index eine Mehrzahl von aus den Feldwerten der gespeicherten Datensätze erzeugten Token umfasst, wobei Token in dem Index jeweils mit einem oder mehreren Zeigern auf

ein oder mehrere der in dem dokumentenorientierten Datenmodell gespeicherten Datensätze verknüpft ist, aus deren Feldwerten das entsprechende Token erzeugt wurde,

wobei erste Token in dem Index, welche von einer der Trigger-Definitionen als Trigger umfasst sind, jeweils der entsprechen Trigger-Definition zugeordnet sind, wobei zweite Token in dem Index jeweils ein oder mehreren Klassen der ersten Gruppe von Klassen zugeordnet sind und wobei die verbleibenden Token in dem Index zum Kennzeichnen der entsprechenden verbleibenden Token als unbekannte Daten einer Auffangklasse zugeordnet sind, wobei die Zuordnung zu der Auffangklasse eine Zuordnung zu einer der Trigger-Definitionen ebenso wie eine Zuordnung zu einer der Klassen der ersten Gruppe von Klassen ausschließt,

- Empfangen eines zusätzlichen Datensatzes,
- Speichern des zusätzlichen Datensatzes, welcher ein oder mehrere zusätzliche Feldwerte umfasst, durch das Multi-Modell-Datenbankmanagementsystem in dem dokumentenorientierten Datenmodell der Datenbank,
- Erzeugen von ein oder mehreren zusätzlichen Token aus den zusätzlichen Feldwerten,
- falls eines oder mehrere erste zusätzliche Token von einer der Trigger-Definitionen als Trigger umfasst sind, Identifizieren des entsprechenden Tokens als Trigger durch das Lernmodul,
- Verwenden der identifizierten Trigger zum Zuordnen von ein oder mehreren zweiten zusätzlichen Token zu ein oder mehreren Klassen der ersten Gruppe von Klassen durch das Lernmodul, falls die entsprechenden zweiten zusätzlichen Token von dem zusätzliche Datensatz in einer Kombination mit ein oder mehreren der identifizierten Trigger gemäß einer der Trigger-Definitionen umfasst sind, wobei die entsprechenden Trigger gemäß der entsprechenden Trigger-Definition eine entsprechende Klassenzuordnung triggern,
- Zuordnen der verbleibenden zusätzlichen Token, für welche keine Zuordnung zu einer der Trigger-Definitionen und keine Klassenzuordnung aufgrund einer der Trigger-Definitionen erfolgt ist, zu der Auffangklasse,

- Ergänzen des Index durch das Multi-Modell-Datenbankmanagementsystem unter Verwendung der zusätzlichen Token, der Klassenzuordnungen der zusätzlichen Token und eines Zeigers auf den zusätzlichen in dem dokumentenorientierten Datenmodell gespeicherten Datensatz.

Ausführungsformen können den Vorteil haben, dass es sich bei dem Lernmodul um ein vortrainiertes Lernmodul handelt. Das vortrainierte Lernmodul umfasst eine Mehrzahl inertial zur Verfügung gestellt bzw. festgelegt Trigger-Definitionen. Das Lernmodul ist dazu konfiguriert unter Verwendung dieser inertial festgelegten Trigger-Definitionen alle von der Datenbank bzw. dem Index umfassten Token zu klassifizieren. Ausführungsformen können den Vorteil haben, dass kein Zufall in den entscheidungs- bzw. Klassifizierungsprozess eingeht. Vielmehr beruht das Klassifizieren von Token auf den vorbestimmten Trigger-Definitionen und ist somit jederzeit nachvollziehbar. Auch wenn das Lernmodule beispielsweise auf Basis der Klassifizierung fortschreitet und weitere Muster und Gesetzmäßigkeiten anhand der dieser im Zuge eines Lerntransfers weitere Muster und Gesetzmäßigkeiten erlernt, so gehen die zugrundeliegenden Klassifizierung zurück auf die vorbestimmten Trigger-Definitionen.

Nach Ausführungsformen werden in Form der Klassifizierung Meta- und/oder Kontextinformationen zu den klassifizierten Token bereitgestellt. Diese Meta- und/oder Kontextinformationen werden anhand der Trigger gemäß den Trigger-Definitionen identifiziert und den entsprechenden Token in Form der Klassenzuordnung zugeordnet. Nach Ausführungsformen kann das Lernmodul dazu konfiguriert sein, unter Verwendung dieser Meta- und/oder Kontextinformationen weitere Muster und Gesetzmäßigkeiten zu erlernen.

Ausführungsformen können den Vorteil haben, dass die von der Datenbank empfangenen Datensätze alle in ihrer ursprünglichen Form in dem dokumentenorientierten Datenmodell abgespeichert werden. Hierdurch kann sichergestellt werden, dass der volle Informationsgehalt dieser Datensätze erhalten bleibt. Zusätzlich werden die von den in dem dokumentenorientierten Datenmodell abgespeicherten

Datensätzen umfassten Daten in Form des Indexes bereitgestellt. Dieser Index umfasst die entsprechenden Daten des dokumentenorientierten Datenmodells in Form von Token. Nach Ausführungsformen umfasst der Index alle von dem dokumentenorientierten Datenmodell umfassten elementaren Datenelemente in Form von elementaren Token. Nach weiteren Ausführungsformen umfasst der Index zusätzlich von dem dokumentenorientierten Datenmodell umfasste Kombination der elementaren Datenelemente in Form von Token-Kombinationen. Diese Token-Kombinationen umfassen jeweils eine Kombination einer Mehrzahl von elementaren Token. Nach weiteren Ausführungsformen umfasst der Index Token-Kombinationen bis zu einer vorbestimmten Komplexität. Die Komplexität einer Token-Kombination ist beispielsweise definiert durch die Anzahl und/oder Art der von dieser umfassten elementaren Token. Nach weiteren Ausführungsformen umfasst der Index alle von dem dokumentenorientierten Datenmodell umfassten Kombinationen elementarer Datenelemente in Form entsprechender Token-Kombinationen.

Bei den von dem Index umfassten Token kann es sich beispielsweise um Trigger gemäß den vorbestimmten Trigger-Definitionen handeln. Ein entsprechendes Token wird, wenn es erstmals beispielsweise im Zuge der Verarbeitung eines zusätzlichen Datensatzes erzeugt wird, anhand einer der Trigger-Definitionen als Trigger identifiziert, dem Index hinzugefügt und der entsprechenden Trigger-Definition zugeordnet. Erkennt das Lernmodul dasselbe Token, welchen der Index als Trigger definiert, innerhalb eines weiteren Datensatzes wieder, so greift das Lernmodul auf die dem Triggern in dem Index zugeordnete Trigger-Definition zurück und ordnet der entsprechenden Trigger-Definition folgenden ein oder mehrere Token aus einem Kontextumfeld des als Trigger gespeicherten Tokens in dem weiteren Datensatz ein oder mehreren Klassen der ersten Gruppe von Klassen zu.

Ferner umfasst der Index eine Mehrzahl von Token, welche jeweils ein oder mehreren Klassen der ersten Gruppe von Klassen zugeordnet sind. Nach Ausführungsformen werden durch die Zuordnung zu den Klassen Meta- und/oder Kontextinformationen zu den entsprechenden Token bereitgestellt. Die entsprechenden Meta- und/oder Kontextinformationen können beispielsweise für eine Verarbeitung der

entsprechenden Token und/oder der die entsprechenden Token umfassenden Datensätze in dem dokumentenorientierten Datenmodell verwendet werden. Beispielsweise werden die entsprechenden Meta- und/oder Kontextinformationen im Zuge einer Suchanfrage zur Identifikation relevanter Token und/oder Datensätze verwendet oder im Zuge eines den Index verwendenden weiteren Verfahrens zum maschinellen Lernen. So können unter Verwendung der entsprechenden Meta- und/oder Kontextinformationen im Zuge eines weiteren Lerntransfers zusätzliche Muster und Gesetzmäßigkeiten erlernt werden. Dieses weitere Verfahren zum maschinellen Lernen wird beispielsweise durch das die Trigger-Definitionen verwendende Lernmodul oder ein weitere Lernmodul ausgeführt. Beispielsweise handelt es sich bei dem weiteren Verfahren zum maschinellen Lernen um ein KI-Verfahren, welches von einem KI-Modul ausgeführt wird.

Schließlich umfasst der Index Token, welche unter keine der vorbestimmten Trigger-Definitionen fallen. Bei diesen Token handelt es sich weder um Trigger, noch lassen sie sich anhand der von den Trigger-Definitionen definierten Trigger Klassen zuordnen. Vielmehr handelt es sich bei diesen Token um unbekannte Daten, welche nicht zuordenbar sind und für welche damit Meta- bzw. Kontextinformationen fehlen. Diese Token werden als unbekannte Daten einer Auffangklasse zugeordnet. Dabei schließt eine Zuordnung zu der Auffangklasse eine Zuordnung zu einer der Trigger-Definitionen ebenso wie eine Zuordnung zu einer der Klassen der ersten Gruppen von Klassen aus.

Ausführungsformen können den Vorteil haben, dass anhand der Token-Zuordnungen in einfacher Form erkannt werden kann, bei welchen Token es sich um unbekannte Daten handelt und bei welchen Token es sich um bekannte Daten, d.h. Trigger oder klassifizierbare Daten, handelt. Somit können beispielsweise Suchabfragen derart definiert werden, dass sie ausschließlich bekannte Daten berücksichtigen. Zusätzliche Lernalgorithmen können beispielsweise derart konfiguriert werden, dass sie ausschließlich auf bekannten Daten arbeiten. Mithin kann beispielsweise eine Verwendung von Zufall in einem Entscheidungs- bzw. Klassifizierungsprozess verwendet, selbst wenn zusätzliche Lernalgorithmen zum Einsatz kommen. Grundlage

für alle Lernprozesse und/oder KI-Prozesse bieten in diesem Fall die initial festgelegten Trigger, anhand derer die von der Datenbank empfangenen Daten klassifiziert werden. Nach Ausführungsformen bieten die vorbestimmenden Trigger-Definitionen eine Grundlage für ein überwachtes Lernen. Durch die Verwendung der vorbestimmenden Trigger-Definitionen lassen sich Fehlentwicklungen in selbstklebenden Systemen und/oder bei der Verarbeitung von Suchanfragen vermeiden, welche die in der Datenbank gespeicherten Daten verwenden.

Ausführungsformen können ferner den Vorteil haben, dass zusätzliche Datensätze, welche der Datenbank hinzugefügt werden, jeweils dahingehend analysiert werden, welche der von ihnen umfassten Daten bekannte Daten sind und welche Daten unbekannte Daten sind. In diesem Kontext werden unter bekannten Daten solche Daten verstanden, welche als Trigger bekannt sind, für welche Meta- bzw. Kontextinformationen vorliegen und/oder für welche Meta- bzw. Kontextinformationen unter Verwendung der Trigger-Definitionen aus dem Kontext der Datensätze abgeleitet werden können. Daten, bei welchen es sich weder um Trigger handelt noch um Daten, welche anhand der Trigger-Definitionen klassifizierbar sind, sind unbekannte Daten. Unbekannte Daten werden der Auffangklasse zugeordnet.

Ausführungsformen können den Vorteil haben, dass ein für das maschinelle Lernen optimiertes Datenbanksystem verwendet wird. Das entsprechende Datenbanksystem umfasst alle dem maschinellen Lernen zu Grunde liegenden Daten, d.h. sowohl zum Klassifizieren von Daten verwendete Trigger-Definitionen, als auch die Daten, welche unter Verwendung der Trigger-Definitionen verarbeitet werden. Somit wird ein kontinuierliches Lernen durch das Lernmodul unter Berücksichtigung aller von dem System bzw. dem Lernmodul gesehenen Daten ermöglicht.

Die Datenbank speichert alle empfangenen Datensätze in einem dokumentenorientierten Datenmodell. Ein dokumentenorientiertes Datenmodell bedeutet, dass das Datenmodell keine strukturellen Vorgaben an die zu speichernden Daten stellt. Vielmehr werden die Daten in Dokumenten bzw. Datencontainern in der Form gespeichert, in der empfangen werden. In diesem Sinne handelt es sich bei den in dem

dokumentenorientierten Datenmodell gespeicherten Daten um Rohdaten. Rohdaten bedeutet, dass die Daten in der Form abgespeichert werden, in der sie empfangen werden, ohne eine zusätzliche Datenverarbeitung durch das Datenbankmanagementsystem, insbesondere keine Umstrukturierung der Daten. Ausführungsformen können den Vorteil haben, dass somit der gesamte Informationsgehalt der empfangenen Daten (nahezu) vollständig beibehalten werden kann, ohne dass Vorannahmen des Datenbankmanagementsystems einfließen. Das Datenbankmanagementsystem ebenso wie das Lernmodul können jederzeit auf die ursprünglichen Datenbestände zurückgreifen und diese in bei der weiteren Verarbeitung berücksichtigen.

Basierend auf diesem Datenpool an Rohdaten, welchen das dokumentenbasierte Datenmodell bereitstellt, wird ein Index erzeugt. Erst auf dieser Ebene werden aus den Datensätze strukturelle Information bzw. Sinnzusammenhänge extrahiert. Diese strukturellen Informationen bzw. Sinnzusammenhängen werden in Form von Klassenzuordnungen der indexierten Daten berücksichtigt.

Hierzu werden die Datensätze durch eine Tokenisierung auf ein elementares Niveau heruntergebrochen, welches die elementaren Bestandteile der Datensätze in Form der Token berücksichtigt. Die Token werden durch das Lernmodul als Trigger einer der Trigger-Definitionen zugeordnet oder als unter Verwendung der Trigger-Definitionen klassifiziert. Alle Token, welche weder als Trigger identifiziert werden, noch sich unter Verwendung einer der Trigger-Definitionen klassifiziert lassen, werden als unbekannte Daten der Auffangklasse zugeordnet. Beispielsweise umfasst das Lernmodul einen Klassifizierer und ist zur Klassifizierung der Token unter Verwendung der vorbestimmten Trigger-Definitionen konfiguriert. Die entsprechende Klassifikation kann beispielsweise Teil einer Mustererkennung sein, bei welcher eine Merkmalsgewinnung durch die Tokenisierung implementiert wird. Basierend auf diese Merkmalsgewinnung erfolgt jedoch keine Merkmalsreduktion im klassischen Sinne, da der vollständige Datenbestand indexiert wird und somit jedes Token als Trigger erkannt oder einer Klasse, zumindest der Auffangklasse, zugeordnet wird.

Zudem ist jedes Token in dem Index mit einem oder mehreren Zeigern verknüpft, welche anzeigen, in welchen Datensätzen das entsprechende Token vorkommt. Somit kann jederzeit auf die für einen Token relevanten Rohdaten zugegriffen werden und diese Rohdaten können zur Auswertung in Hinblick auf diesen Token herangezogen werden.

Hierdurch werden die Token-Zuordnungen nach bekannten Daten, welche gesicherten Fakten darstellen, und unbekannte Daten differenziert. Ausführungsformen können den Vorteil haben, dass die Verwendung der von dem Lernmodul bestimmten Strukturen und Gesetzmäßigkeiten in den Datensätzen, welche sich in den Token-Zuordnungen niederschlagen, auf den einer Verwendung der vorbestimmten Trigger Definitionen beruht. Unbekannt Daten werden demgegenüber als solche erfasst und solange außen vorgelassen werden, bis diese ebenfalls klassifiziert und mithin als gesicherte Fakten angesehen werden können. Eine solche zusätzliche Klassifikation kann beispielsweise durch zusätzliche Trigger-Definitionen implementiert werden. Insbesondere können gezielt zusätzliche Trigger-Definitionen zur Reduktion des von der Auffangklasse umfassten Token-Menge ergänzt werden. Das Verfahren ermöglicht somit ein Lernen und/oder Klassifizieren unter Vorbehalt.

Ausführungsformen können daher den Vorteil haben, dass sie dem Lernmodul erlauben auf dem gesamten zur Verfügung stehenden Datenbestand zu arbeiten. Insbesondere können sie den Vorteil haben, ein kontinuierliches Lernen zu ermöglichen, welches sowohl zusätzliche Datensätze als auch bereits gespeicherte Datensätze berücksichtigt. Ausführungsformen können daher den Vorteil haben, dass sie nicht darauf beschränkt sind, dass aus einer zur Verfügung stehenden Gesamtdatenmenge eine willkürliche Untermenge herausgegriffen wird, um auf dieser zu trainieren. Vielmehr werden alle von der Datenbank umfassten Daten unter Verwendung der Trigger-Definitionen verarbeitet. Durch Ergänzungen der Trigger-Definitionen kann nach Ausführungsformen zudem erreicht werden, dass alle Token entweder als Trigger identifiziert oder unter Verwendung der (ergänzten) Trigger-Definitionen klassifiziert werden. Werden unbekannte Daten von Suchanfragen und/oder

weiteren Lernprozessen ausgeschlossen, so erfolgt dieser Ausschluss nicht willkürlich, sondern basierend auf den bereitgestellten Trigger-Definitionen.

Ausführungsformen können den Vorteil haben, dass keine zufällige Initialisierung erforderlich ist, wie etwa bei bekannten selbstlernenden Systemen, z.B. neuronalen Netzwerken. Die Initialisierung beruht vielmehr auf den vorbestimmten Trigger-Definitionen. Durch das aus dieser zufälligen Initialisierung resultierende Zufallsmoment werden die Entscheidungen/Klassifizierungen eines entsprechenden neuronalen Netzes nicht transparent und nicht nachvollziehbar. Demgegenüber können Ausführungsformen den Vorteil haben, vollständig deterministisch zu sein.

Ausführungsformen können den Vorteil haben, dass ein bereits trainiertes System, d.h. das vortrainierte Lernmodul nachtrainiert bzw. weiter trainiert wird. So können Trigger-Definitionen ergänzt, entfernt oder geändert werden. Dadurch lassen sich beispielsweise auch die bei der Klassifizierung verwendeten Klassen ergänzen, entfernen oder ändern. Werden Trigger-Definitionen ergänzt, entfernt oder geändert, so sind alle auf diesen basierenden Zuordnungen von Token zu den entsprechenden Trigger-Definitionen oder zu einer der Klassen entsprechend anzupassen. Durch ein Ergänzen von Trigger-Definitionen können beispielsweise neue Strukturen erfasst werden, welche sich bisher noch nicht gezeigt haben. Dadurch kann insbesondere das Faktenwissen nachträglich erweitert werden, d.h. der Auffangklasse zugeordneten Token können andere Klassen zugeordnet werden.

Nach Ausführungsformen implementiert das Lernmodul einen Algorithmus zum maschinellen Lernen, wobei das Verfahren nicht beschränkt ist auf einen bestimmten Algorithmus. Nach Ausführungsformen umfasst der Algorithmus zum maschinellen Lernen zumindest einen Klassifizierungsalgorithmus zum Klassifizieren von Token. Bei dem maschinellen Lernen kann es sich um ein überwachtes oder ein unüberwachtes Lernen handeln. Das maschinelle Lernen kann eine Klassifizierung und/oder eine Regressionsanalyse umfassen. Ein Lernalgorithmus versucht, eine Hypothese bzw. eine Abbildung zu finden, welche jedem Eingabewert den (vermuteten) Ausgabewert zuordnet. Liegen die zuzuordnenden Ausgabewerte in einer

kontinuierlichen Verteilung vor, deren Ergebnisse beliebige quantitative Werte eines vorgegebenen Wertebereiches annehmen kann, wird im Allgemeinen von einem Regressionsproblem gesprochen. Liegen die zuzuordnenden Ausgabewerte hingegen in diskreter Form vor bzw. sind die Werte qualitativ, wird im Allgemeinen von einem Klassifikationsproblem gesprochen. Nach Ausführungsformen stützt sich das maschinelle Lernen auf die Klassifikation der indizierten Token. Gemäß Ausführungsformen der Erfindung umfasst das Lernmodul einen speziell für das maschinelle Lernen entwickelten Algorithmus, wie zum Beispiel, ohne darauf beschränkt zu sein, eine dichtenbasierte multidimensionale Ausreißererkennung (engl. „local outlier detection"), ein Random-Forrest-Algorithmus, ein Neuronales Netz, eine Support-Vektor-Maschine, ein Naive-Bayes-Klassifikator oder eine Rückkopplung ähnlich der Rückkopplung eines linearen oder nichtlinearen Reglers.

Unter einer Multi-Modell-Datenbank wird hier eine Datenbank verstanden, welche dazu konfiguriert ist eine Mehrzahl von unterschiedlichen Datenmodellen zu unterstützen. Eine Multi-Modell-Datenbank ist also dazu konfiguriert Daten in mehr als einem Datenmodell zu speichern, zu indexieren und abzufragen. Datenmodelle sind beispielsweise relational, spaltenorientiert, dokumentenorientiert, graphbasiert, Key-Value-basiert etc. Ein Datenbankmodell legt fest, in welcher Struktur Daten in einem Datenbanksystem gespeichert werden, d.h. in welcher Form die Daten organisiert, gespeichert und bearbeitet werden.

Unter einer Datenbank wird im Folgenden eine (typischerweise große) Menge von Daten verstanden, die in einem Computersystem von einem Datenbankmanagementsystem (DBMS) nach bestimmten Kriterien verwaltet wird. Die Daten sind dabei in einer Vielzahl von Datensätzen organisiert. Unter einem Datenbankmanagementsystem oder DBMS wird im Folgenden ein elektronisches System zur Speicherung und Wiedergewinnung von Daten verstanden. Vorzugsweise werden die Daten in dem DBMS widerspruchsfrei und dauerhaft gespeichert und verschieden Anwendungsprogrammen und Nutzern in bedarfsgerechter Form effizient zur Verfügung gestellt. Ein DBMS kann typischerweise ein oder mehrere Datenbanken beinhalten und die darin enthaltenen Datensätze verwalten. Bei dem DBMS kann es sich

vorzugsweise um ein feldorientiertes DBMS handeln, also um ein DBMS, das dazu konfiguriert ist, Teile einzelner Datensätze, sogenannte Feldwerte, in mehreren unterschiedlichen Feldern zu speichern.

Unter einem Datensatz wird im Folgenden eine dem Datenbanksystem bereitgestellte zusammenhängende Menge von Daten verstanden, welche von dem Datenbankmanagementsystem als zusammenhängende Menge von Daten verwaltet wird. Ein Datensatz umfasst beispielsweise eine Menge inhaltlich zusammenhängender Daten. Nach Ausführungsformen werden Datensätze jeweils als zusammenhängende Datenmengen in dem dokumentenorientierten Datenmodell gespeichert. Beispielsweise kann ein einzelner Datensatz ein bestimmtes physisches Objekt, z.B. eine natürliche Person oder eine Vorrichtung, repräsentieren. Bei der Person kann es sich z.B. um einen Angestellten, einen Patienten, einen Kunden, etc. handeln. Bei der Vorrichtung kann es sich beispielsweise um eine Produktionsvorrichtung, eine Computervorrichtung, ein Computer- oder Netzwerkelement oder eine Transportvorrichtung handeln. Der entsprechende Datensatz kann eine vordefinierte Menge von Attributwerten dieser Person oder Vorrichtung beinhalten (z.B. Name oder Pseudonym, Alter, Größe, Gewicht, Geburtsdatum, Ausweisnummern, Sicherheitszertifikate, Authentifizierungscodes, biometrische Daten, Identifikator, Herstellungsdatum, Inbetriebnahmedatum, Konfigurationsdaten, und andere). Beispielsweise kann ein Datensatz eine Gruppe von inhaltlich zusammenhängenden (zu einem Objekt gehörenden) Datenfeldern repräsentieren, z. B. Artikelnummer, Artikelgröße, Artikelfarbe, Artikelname oder ähnliches. Die Klassen ‚Name', ‚Adresse' und ‚Geburtsdatum' könnten z.B. die logische Struktur eines Datensatzes zum Objekttyp „Person" bilden. In der Datenverarbeitung werden Daten in Form von Datensätzen in Datenbanken gespeichert, wobei sie sind Gegenstand der Verarbeitung von Computerprogrammen und werden von diesen erzeugt, gelesen, verändert und gelöscht.

Ein „NoSQL" (englisch für Not only SQL) DBMS ist ein DBMS, welches einem nichtrelationalen Ansatz der Datenspeicherung folgt und keine festgelegten Tabellenschemata benötigt. Zu den NoSQL DBMSs gehören insbesondere

dokumentenorientierte DBMSs wie Apache Jackrabbit, BaseX, CouchDB, IBM Notes, MongoDB, Graphdatenbanken wie Neo4j, OrientDB, InfoGrid, HyperGraphDB, Core Data, DEX, AllegroGraph, und 4store, verteilte ACID-DBMSs wie MySQL Cluster, Key-Value-Datenbanken wie Chordless, Google BigTable, GT.M, InterSystems Caché, Membase, Redis, sortierte Key-Value-Speicher, Multivalue-Datenbanken, Obektdatenbanken wie Db4o, ZODB, spaltenorientierte Datenbanken und temporale Datenbanken wie Cortex DB.

Ein Index ist eine Datenstruktur, welche eine Suche nach bestimmten Datenwerten durch ein Datenbankmanagementsystem beschleunigt. Ein Index besteht aus einer Ansammlung von Zeigern (Verweisen), die eine Ordnungsrelation auf mehrere (in dem Index gespeicherte) „indizierte“ Datenwerte definieren. Beispielsweise werden hierfür B+-Bäume verwendet. Jeder indizierte Datenwert ist mit weiteren Zeigern verknüpft, die auf Datensätze verweisen, in welchen der gefundene indizierte Datenwert enthalten ist und welche die Datenbasis für die Erstellung des Index darstellten. Datenbankmanagementsysteme verwenden Indizes um als Antwort auf eine Suchanfrage die gewünschten Datensätze schnell zu identifizieren, indem zunächst der Index entlang der Zeiger nach einem Datenwert durchsucht wird, welcher identisch zu einem in der Suchanfrage enthaltenen Referenzwert ist. Ohne Index müssten die von dem DBMS verwalteten Datenwerte eines Feldes sequenziell durchsucht werden, während eine Suche mit Hilfe des Index, z.B. eines B+-Baums, oft nur logarithmische Komplexität hat.

Ferner ordnet der Index die indizierten Daten, d.h. Token, Klassen zu, wodurch die entsprechenden Daten mit Meta- bzw. Kontextinformationen verknüpft werden. Diese Meta- bzw. Kontextinformationen können bei einer Suche und/oder bei einem maschinellen Lernprozess auf den Daten der Datenbank verwendet werden.

Unter einem Feld wird im Folgenden ein Bereich auf einem logischen oder physikalischen Datenträger bezeichnet, der von einem DBMS verwaltet wird, der einem vordefinierten Feldtyp zugeordnet ist und der zur Speicherung eines Feldwertes eines Datensatzes angelegt und bestimmt ist. Ein Feld ist also ein Element zur

Speicherung eines Feldwertes eines Datensatzes gemäß obiger Definition. Felder eines Datensatzes werden von einem DBMS gemeinsam verwaltet.

Ein Feldwert ist ein Datenwert, der Bestandteil eines Datensatzes ist und in einem Feld des Datensatzes gespeichert ist. Ein Feldwert kann aus einem einzigen Wort, einer einzigen Zahl, oder einer Kombination aus mehreren Wörtern und/oder Zahlen und/oder anderen Datenformaten bestehen, wobei verschiedene Ausführungsformen der Erfindung verschiedene Grade an Flexibilität im Hinblick auf die Art und Kombinierbarkeit von Datentypen innerhalb des gleichen Feldwertes umfassen.

Ein „Tokenisierer“ ist eine Programlogik, die Daten, zum Beispiel einen Feldwert, als Input erhält, die Daten analysiert, z.B. um Delimiter oder andere Zerlegungskriterien und Muster zu erkennen, und die Daten dann in ein oder mehrere Token als Ergebnis der Analyse zerlegt und die Token zurückgibt. Es ist auch möglich, dass nicht alle Daten als Token zurückgegeben werden. Beispielsweise kann ein Volltextindizierer semantisch unbedeutende Stoppwörter erkennen und herausfiltern, sodass diese nicht indiziert werden. Alternativ werden alle Daten als zurückgegeben. Einen Datenwert zu „tokenisieren“ bedeutet also, den Datenwert nach einem bestimmten Schema in mehrere Bestandteile zu zerteilen. Die Bestandteile stellen die Token dar. So können z.B. natürlichsprachige Texte an vordefinierten Trennzeichen, z.B. Leerzeichen, Punkten oder Kommata, aufgeteilt werden, die so generierten Bestandteile (Wörter) werden als Token verwendet. Nach Ausführungsformen werden alle Token für die Indizierung verwendet. Es ist auch möglich, dass manche Token nicht für die Indizierung verwendet werden (z.B. Stopwörter) oder die Token vor der Indizierung zusätzlich verarbeitet werden (z.B. Reduzierung von Wörtern auf den Wortstamm). In diesem Fall erfolgt für eine Verarbeitung von Suchanfragen eines Client-Computersystems an ein Server-Computersystem vorzugsweise eine gleichartige Verarbeitung des Suchwerts durch das Client-Computersystem oder das Server-Computersystem um sicherzustellen, dass die Suchwerte der Suchanfragen den in dem Index enthaltenen Token entsprechen.

Ein Klasse definiert beispielsweise eine Kategorie bzw. ein Typ, dem ein Token angehört. Die Klasse ordnet dem Token mithin Meta- bzw. Kontextinformationen, etwa in Form einer Eigenschaft zu. Beispielsweise kann eine Klasse ein bestimmtes Attribut eines physischen Objekts in Form eines Token repräsentieren. Beispielsweise können zu speichernde Datensätzen, die Attribute von Angestellten enthalten, welche Klassen wie „Name“, „Pseudonym“, „Ausweisnummer“; „Zugriffszertifikat für Raum R“, „Zugriffszertifikat für Gerät G“, „Zugriffszertifikat für Gebäude GB“, „Alter“ repräsentieren. Jedes Token kann ein oder mehreren Klassen zugeordnet sein. Ferner können Kombinationen von Token als eigenständige Token wiederum ein oder mehreren weiteren Klassen zugeordnet sein.

Die empfangenen Datensätze werden unter Verwendung eines dokumentorientierten Datenmodell abgespeichert. Beispielsweise werden alle Feldwerte der abgespeicherten Datensätze als Token in einen mehrdimensionalen Schlüssel/Wert-Speicher (Key/Value-Store) bzw. Key-Value-Datenbanken übertragen. Dabei werden die Token Tokentypen zugeordnet und in einer Form gespeichert, welche die sechste Normalform erfüllt.

Nach Ausführungsformen werden zusätzlich Transaktionszeit und Gültigkeitszeit der Datensätze bitemporal gespeichert. Die Transaktionszeit gibt den Zeitpunkt an, zu dem eine Änderung eines Datenobjekt in der Datenbank erfolgt. Die Gültigkeitszeit gibt einen Zeitpunkt oder Zeitraum an, in dem ein Datenobjekt im modellierten Abbild der realen Welt den beschriebenen Zustand aufweist. Sind sowohl Gültigkeits- als auch Transaktionszeit relevant, spricht man von bitemporal.

Ein Schlüssel-Werte-Datenmodell ermöglicht ein Speichern, Abrufen und Verwalten von assoziativen Datenfeldern. Dabei werden Werte (Value) über einen Schlüssel (Key) eindeutig identifiziert.

In einem dokumentenorientierte Datenmodell, auch bekannt als Dokumentenspeicher (Document Store), bilden Dokumente bzw. Datencontainer die Grundeinheit zur Speicherung der Daten bilden. Ein dokumentenorientiertes Datenmodell

ermöglicht es dokumentenorientierten Informationen, auch bekannt als halbstrukturierte Daten, zu speichern, abzurufen und zu verwalten. Auf einem dokumentenorientierten Datenmodell beruhende Datenbanken gehören zu den NoSQL-Datenbanken und bilden eine Unterklasse der Schlüssel-Wert-Speicher (Key-value Stores). In einem Schlüssel-Wert-Speicher werden die Daten als für die Datenbank inhärent undurchsichtig angesehen, während eine dokumentenorientierte Datenbank auf interne Strukturen in den gespeicherten Dokumenten angewiesen ist, um Metadaten zu extrahieren. Das semistrukturierte Datenmodell ist ein Datenbankmodell, bei dem keine Trennung zwischen den Daten und dem Schema besteht und der Umfang der verwendeten Struktur vom Verwendungszweck der Datenbank abhängt. Jedes Dokument innerhalb des Datenmodells wird über einen eindeutigen Identifikator angesprochen.

Eine Kombination der verschiedenen Datenbankkonzepten ermöglicht es Datensätze als Dokumente bzw. Container zu speichern (document store) und zusätzlich in Form eines Index, z.B. eines Schlüssel-Wert-Speichers, in die 6. Normalform zu überführen. Dieser Schlüssel-Wert-Speicher repräsentiert den gesamten Datenumfang in dem Dokumentenspeicher, während die Originaldatensätze erhalten bleiben.

Nach Ausführungsformen werden Selektionen ausschließlich in dem Schlüssel-Wert-Speicher in der redundanzfreien sechsten Normalform durchgeführt. Erst das Ergebnis wird aus den Containern des Dokumentenspeicher gelesen. Nach Ausführungsformen wird neben Lese- und Schreibrechten in den Datensätzen zusätzlich ein Selektionsrecht auf dem Schlüssel-Wert-Speicher implementiert. Somit kann auch auf dem Index alleine gearbeitet werden, ohne die zugrundeliegenden Daten auslesen zu müssen.

Die vorgeschlagene Multi-Modell-Datenbank stellt somit neben einer schemalosen Datenablage auf Basis eines Dokumentenspeichers eine vollständige Normalisierung des gesamten Datenumfangs in der sechsten Normalform bereit. Ausführungsformen können den Vorteil haben, dass der Index Datenelemente der Datensätze, d.h. Token, als Schlüssel umfasst und jedem dieser Schlüssel jeweils ein oder

mehrere Zeiger als Werte zugeordnet sind, welche anzeigen, in welchen Datensätzen und/oder Feldern der Datensätze der entsprechende Schlüssel, d.h. Token/Datenwert, als Feldwert gespeichert ist.

Dieser Index bildet daher über alle Felder der Datensätze und deren Inhalte, d.h. die Feldwerte, die gesamte Datenbank mit allen von ihr umfassten Datensätze ab, so dass alle Abfragen in dem Index behandelt und die schemalos gespeicherten Daten des dokumentenorientierten Datenmodells nur zur Ausgabe der Suchergebnisse verwendet werden. Die geringe Größe des Index im Vergleich zu den schemalosen Daten ermöglicht schnelle Abfragen in beliebigen Abfragekombinationen.

Unter einem Computer bzw. Computersystem wird hier ein Gerät verstanden, welches mittels programmierbarer Rechenvorschriften Daten verarbeitet. Unter einem Programm bzw. Programminstruktionen wird hier ohne Einschränkung jede Art von Computerprogramm verstanden, welches maschinenlesbare Instruktionen zur Steuerung einer Funktionalität eines Computers umfasst. Ein Computer bzw. Computersystem kann eine Kommunikationsschnittstelle zur Verbindung mit dem Netzwerk umfassen, wobei es sich bei dem Netzwerk um ein privates oder öffentliches Netzwerk handeln kann, insbesondere das Internet oder ein anderes Kommunikationsnetz. Je nach Ausführungsform kann diese Verbindung auch über ein Mobilfunknetz hergestellt werden.

Bei einem Computersystem kann es sich um ein stationäres Computersystem, wie beispielsweise einen Personalcomputer (PC) oder einen in einer Client-Server-Umgebung eingebundenen Client bzw. Server handeln. Ferner kann es sich bei einem Computersystem beispielsweise um ein mobiles Telekommunikationsgerät, insbesondere ein Smartphone, einen tragbaren Computer, wie zum Beispiel einen Laptop PC oder Palmtop-PC, ein Tablet PC, einen Personal Digital Assistant oder dergleichen handeln.

Unter einem Speicher werden hier sowohl flüchtige als auch nicht flüchtige elektronische Speicher bzw. digitale Speichermedien verstanden.

Unter einem nichtflüchtigen Speicher wird hier ein elektronischer Speicher zur dauerhaften Speicherung von Daten verstanden. Ein nichtflüchtiger Speicher kann als nichtänderbarer Speicher konfiguriert sein, der auch als Read-Only Memory (ROM) bezeichnet wird, oder als änderbarer Speicher, der auch als Non-Volatile Memory (NVM) bezeichnet wird. Insbesondere kann es sich hierbei um ein EEPROM, beispielsweise ein Flash-EEPROM, kurz als Flash bezeichnet, handeln. Ein nichtflüchtiger Speicher zeichnet sich dadurch aus, dass die darauf gespeicherten Daten auch nach Abschalten der Energieversorgung erhalten bleiben.

Unter einem flüchtigen elektronischen Speicher wird hier ein Speicher zur vorübergehenden Speicherung von Daten, welcher dadurch gekennzeichnet ist, dass alle Daten nach dem Abschalten der Energieversorgung verloren gehe. Insbesondere kann es sich hierbei um einen flüchtigen Direktzugriffsspeicher, der auch als Random-Access Memory (RAM) bezeichnet wird, oder einen flüchtigen Arbeitsspeicher des Prozessors handeln.

Unter einem Prozessor wird hier und im Folgenden eine Logikschaltung verstanden, die zur Ausführung von Programminstruktionen dient. Die Logikschaltung kann auf einem oder mehreren diskreten Bauelementen implementiert sein, insbesondere auf einem Chip. Insbesondere wird unter einem Prozessor ein Mikroprozessor oder ein Mikroprozessorsystem aus mehreren Prozessorkernen und/oder mehreren Mikroprozessoren verstanden.

Nach Ausführungsformen umfasst das Ergänzen des Index:

- Abgleichen der zusätzlichen Token mit dem Index,
- falls eines der zusätzlichen Token nicht von dem Index umfasst ist, Ergänzen des entsprechenden zusätzlichen Tokens unter seinen Klassenzuordnungen in dem Index und Verknüpfen des entsprechenden zusätzlichen Tokens in dem Index mit dem Zeiger auf den zusätzlichen in dem dokumentenorientierten Datenmodell gespeicherten Datensatz,

- falls eine der Klassenzuordnungen eines von dem Index umfassten zusätzlichen Tokens von dem Index nicht umfasst ist, Ergänzen der entsprechenden Klassenzuordnung mit dem entsprechenden zusätzlichen Token in dem Index und Verknüpfen des entsprechenden zusätzlichen Tokens in dem Index mit dem Zeiger auf den zusätzlichen in dem dokumentenorientierten Datenmodell gespeicherten Datensatz,
- falls eines der zusätzlichen Token mit allen seinen Klassenzuordnungen von dem Index umfasst ist, Verknüpfen des entsprechenden zusätzlichen Tokens in dem Index mit dem Zeiger auf den zusätzlichen in dem dokumentenorientierten Datenmodell gespeicherten Datensatz.

Ausführungsformen können den Vorteil haben, dass Daten aus zusätzlichen Datensätze effizient in die bestehende Datenbank und insbesondere in den Index eingefügt werden können. Die unter Verwendung des zusätzlichen Datensatzes erzeugten Token werden mit dem Index abgeglichen. Alle Token, welche der Index (noch) nicht umfasst, werden in dem Index als zusätzliche Token inklusive ihrer Klassenzuordnungen ergänzt. Ferner werden die zusätzlichen Token jeweils mit dem Zeiger auf den zusätzlichen Datensatz verknüpft.

Für solche Token des zusätzlichen Datensatzes, welche der Index bereits umfasst, für welche aber ein oder mehrere unter Verwendung des zusätzlichen Datensatzes ermittelte Klassenzuordnungen von dem Index noch nicht berücksichtigt werden, werden die entsprechenden Klassenzuordnungen ergänzt. Zusätzlich wird in dem Index für diese Token jeweils der Zeiger auf den zusätzlichen Datensatz ergänzt.

Schließlich wird für solche Token des zusätzlichen Datensatzes, welche der Index bereits mit all ihren für den zusätzlichen Datensatz ermittelten Klassenzuordnungen umfasst, lediglich der Zeiger auf den zusätzlichen Datensatz ergänzt.

Ausführungsformen können den Vorteil haben, dass stets sichergestellt werden kann, dass der Index alle von den Datensätzen der Datenbank umfassten Token aufweist. Zudem umfasst der Index für alle entsprechenden Token alle

aufgefundenen Klassenzuordnungen. Zudem ist jeder der Token des Index mit Zeiger auf alle Datensätze der Datenbank verknüpft, welche das entsprechende Token umfassen.

Beispielsweise wird im Zuge des Vortrainierens des Lernmodus eine initiale Menge von vorbestimmten Trigger-Definitionen festgelegt. Im Zuge eines Erfassens von Daten, werden Datensätze empfangen und in dem dokumentenbasierten Datenmodell gespeichert. Die gespeicherten Datensätze werden tokenisiert und für die resultierenden Token werden Klassen-Zuordnungen unter Verwendung der initial festgelegten Trigger-Definitionen ermittelt und ein initialer Index für die resultierenden Token erzeugt. Nach Ausführungsformen umfasst der initiale Index alle von den Trigger-Definitionen umfassten Trigger als Token. Nach alternativen Ausführungsformen werden durch die Trigger-Definitionen als Trigger festgelegte Token in dem Index nur unter der Voraussetzung ergänzt, dass sie von einem der Datensätze umfasst sind.

Eine Zuordnung eines Token zu einer Klasse unter Verwendung einer vorbestimmten Trigger-Funktion stellt ein durch die entsprechende vorbestimmten Trigger-Funktion gesichertes Faktum dar. Für Token, bei welchen es sich um keinen Trigger handelt und welche von keiner der Trigger-Definitionen erfasst werden, fehlt es an einem solchen Faktenwissen. Die entsprechenden Token werden vielmehr als unbekannte Daten der Auffangklasse zugeordnet. Ausführungsformen können somit den Vorteil haben, dass unter Verwendung von initial festgelegten Trigger-Definitionen neue Daten in bekannte Daten, d.h. Trigger oder unter Verwendung von Trigger-Definitionen klassifizierter Token, und unbekannte Daten eingeteilt werden können, d.h. der Auffangklasse zugeordnete Token.

Nach Ausführungsformen werden die Kombinationen aus zweiten zusätzlichen Token mit ein oder mehreren der identifizierten Trigger, welche eine Klassenzuordnung gemäß einer der Trigger-Definitionen getriggert haben, in dem Index als klassifizierte Kombinationen gekennzeichnet und Klassenzuordnungen werden nur für Kombinationen aus zweiten zusätzlichen Token und ein oder mehreren

identifizierten Triggern ausgeführt, welche nicht als klassifizierte Kombinationen gekennzeichnet sind.

Ausführungsformen können den Vorteil haben, dass für alle Token-Kombinationen, für welche bereits bei einer Klassenzuordnung berücksichtigt wurden bzw. für welche bereits eine Klassenzuordnung ausgeführt wurde, in dem Index jeweils als bereits klassifiziert gekennzeichnet werden. Somit lässt sich vermeiden, dass für Token-Kombinationen, welche das Lernmodul bereits zuvor gesehen und im Zuge der Klassifizierungen vollständig berücksichtigt hat, dieselben Klassifizierungen erneut ausgeführt. Somit kann das System deutlich effizienter ausgestaltet werden. Beispielsweise umfasst der Index neben elementaren Token alle Token-Kombinationen, für welche bereits eine Klassifizierung erfolgt ist, d.h. alle Token-Kombinationen, welche als klassifiziert zu kennzeichnen sind. Nach Ausführungsformen sind die entsprechenden Token-Kombinationen in dem Index jeweils mit einem Flag versehen, welches anzeigt, ob es sich bei der entsprechenden Token-Kombinationen um klassifizierte Token-Kombinationen handelt.

Nach Ausführungsformen erfolgt auf eine Tokenisierung eines zusätzlichen Datensatzes hin zunächst ein Abgleich mit allen als bereits klassifiziert gekennzeichneten Token-Kombinationen. Für diese Token-Kombinationen wird die Klassifizierung nicht wiederholt, vielmehr erfolgt lediglich eine Verknüpfung mit dem Zeiger auf den zusätzlichen Datensatz. Nach Ausführungsformen wird der entsprechende Zeiger auch mit allen von der Token-Kombination umfassten Token in dem Index verknüpft. Nach Ausführungsformen erfolgt der Abgleich zunächst mit den größten, d.h. umfangreichsten, Token-Kombinationen des Index. Für alle bereits als klassifiziert erkannten Token-Kombinationen des zusätzlichen Datensatzes wird lediglich der Zeiger auf den entsprechenden Datensatz in der Datenbank abgespeichert. Nach Ausführungsformen wird der entsprechende Zeiger auch mit allen von der Token-Kombination umfassten Token in dem Index verknüpft. Somit kann das Einarbeiten der Daten des zusätzlichen Datensatzes in den Index deutlich beschleunigt werden. Nach Ausführungsformen erfolgt sukzessive ein Abgleich mit weiteren Token-Kombination, wobei die Größe bzw. der Umfang der verwendeten weiteren Token-

Kombinationen sukzessive abnimmt. Nach Ausführungsformen werden nur solche weiteren Token-Kombinationen mit geringerer Größe bzw. Umfang berücksichtigt, welche nicht als Teil einer größeren bzw. umfangreichern Token-Kombinationen eine Übereinstimmung im Zuge des Abgleichs festgestellt wurde. Ausführungsformen können den Vorteil haben, dass für umfangreiche Token-Kombinationen, welche als bereits klassifiziert erkannt werden, kein zusätzlicher Abgleich für von der entsprechenden Token-Kombination umfasste Unterkombinationen erfolgt. Vielmehr erfolgt ein entsprechender Abgleich lediglich, wenn die entsprechende Unterkombination in dem zusätzlichen Datensatz unabhängige von der entsprechenden umfangreicheren Token-Kombination als eigenständige Token-Kombination umfasst ist.

Nach Ausführungsformen umfasst das Verfahren ferner:

- Identifizieren von ein oder mehreren Trigger-Kombinationen, welche jeweils von zumindest einem der Datensätzen umfasst sind und ein Kombinationskriterium erfüllen,
- für jede der identifizierten Trigger-Kombinationen Kombinieren der Trigger-Definitionen der Trigger der entsprechenden Trigger-Kombinationen zu ein oder mehreren zusätzlichen kombinierten Trigger-Definitionen,
- Ergänzen der Mehrzahl von vorbestimmten Trigger-Definitionen des Lernmoduls durch die ein oder mehreren zusätzlichen kombinierten Trigger-Definitionen.

Ausführungsformen können den Vorteil haben, dass basierend auf den von den initialen Trigger-Definitionen identifizierten Triggern zusätzliche Trigger in Form von Trigger-Kombinationen identifiziert werden können. Basierend auf diesen identifizierten Trigger-Kombinationen können aus den initialen Trigger-Definitionen kombinierte Trigger-Definitionen bestimmt werden, mit denen die Mehrzahl der vorbestimmten Trigger-Definitionen erweitert werden kann.

Nach Ausführungsformen werden mehrere Token-Kombinationen, welche von demselben Datensatz umfasst werden und unter die kombinierte Trigger-Definition

fallen, miteinander kombiniert und die resultierende Kombination in dem Index als eine klassifizierte Kombination gekennzeichnet. Ausführungsformen können den Vorteil haben, dass auch auf Basis kombinierte Trigger-Definitionen Token-Kombinationen im Index als klassifizierte Kombinationen gekennzeichnet und dadurch unnötige Wiederholungen von Klassifizierungen bereits klassifizierter Token-Kombinationen vermieden werden können.

Nach Ausführungsformen umfasst das Kombinationskriterium eine Mindesthäufigkeit für ein Auftreten der entsprechenden Trigger-Kombination in den Datensätzen. Ausführungsformen können den Vorteil haben, dass entsprechende Trigger-Kombinationen nur dann herangezogen werden zum Bilden einer kombinierten Trigger-Definition, wenn die entsprechende Trigger-Kombination in den Datensätzen mit einer Mindesthäufigkeit auftreten. Somit kann verhindert werden, dass zusätzliche kombinierte Trigger-Definition gebildet werden aufgrund eines zufälligen Auftretens von Triggern unterschiedlicher Trigger-Definitionen in ein und demselben Datensatz. Ein solches zufälliges Auftreten ist ab einer bestimmten Größe und/oder Komplexität der Datensätze zu erwarten, ohne dass daraus Rückschlüsse auf einen zugrundeliegenden Zusammenhang zwischen den Triggern geschlossen werden könnte. Treten die entsprechenden Trigger-Kombinationen jedoch häufiger auf, so kann daraus auf einen Zusammenhang geschlossen werden.

Nach Ausführungsformen legt die Mindesthäufigkeit einen absoluten Häufigkeitswert des Auftretens in den Datensätzen fest. Bei der entsprechenden Mindesthäufigkeit kann es sich um einen Mindestwert für das Auftreten der entsprechenden Trigger-Kombination in allen Datensätzen handeln. Das Auftreten der entsprechenden Trigger-Kombination wird über alle Datensätze aufsummiert. Ist die resultierende Summe größer oder gleich dem Mindestwert, so ist dieser erfüllt. Ferner kann es sich dabei der Mindesthäufigkeit um eine Mindestwert für das Auftreten in einem der Datensätze handeln. Das Auftreten der entsprechenden Trigger-Kombination wird für die einzelnen Datensätze jeweils individuell aufsummiert. Erfüllt ein der resultierenden Summen den Mindestwert, so liegt die Mindesthäufigkeit vor. Nach alternativen Ausführungsformen muss der Mindestwert von einer vorbestimmten

Anzahl von Datensätzen oder einem vorbestimmten Prozentsatz der Datensätze erfüllt werden. Bei dem entsprechenden vorbestimmten Prozentsatz handelt es sich entweder um einen Prozentsatz aller Datensätze der Datenbank oder aller Datensätze, welche die entsprechende Trigger-Kombination umfassen. Nach alternativen Ausführungsformen muss der Mindestwert von allen Datensätzen erfüllt werden und/oder von allen Datensätzen, welche die entsprechende Trigger-Kombination umfassen. Ferner kann sich bei der entsprechenden Mindesthäufigkeit um einen Mindestwert für eine Durchschnittshäufigkeit des Auftretens der entsprechenden Trigger-Kombination in allen Datensätzen der Datenbank oder allen Datensätzen, welche die entsprechende Trigger-Kombination umfassen, handeln.

Nach Ausführungsformen legt die Mindesthäufigkeit einen relativen Häufigkeitswert des Auftretens in den Datensätzen fest. Nach Ausführungsformen ist die entsprechenden Mindesthäufigkeit abhängig von der Anzahl der Datensätze und/oder der Anzahl der Token und/oder der Größe der von den Datensätzen umfassten Daten. Beispielsweise wächst der von der Mindesthäufigkeit festgelegte Häufigkeitswert mit der Anzahl der Datensätze und/oder der Anzahl der Token und/oder der Größe der von den Datensätzen umfassten Daten.

Nach Ausführungsformen legt die Mindesthäufigkeit einen relativen Häufigkeitswert des Auftretens in den Datensätzen fest relativ zu Häufigkeiten des Auftretens von ein oder mehreren der von der entsprechenden Trigger-Kombination umfassten Triggern in den Datensätzen. Beispielsweise ist der relative Häufigkeitswert von dem Auftreten des Triggers mit der höchsten Häufigkeit eines Auftretens, des Trigger mit der geringsten Häufigkeit eines Auftretens und/oder einem Durchschnittswert des Auftretens aller Trigger der entsprechenden Trigger-Kombination abhängig. Ausführungsformen können den Vorteil haben, dass bei einer Berücksichtigung eines relativen Häufigkeitswerts die Häufigkeit des Auftretens von ein oder mehreren der von der entsprechenden Trigger-Kombination umfassten Triggern in den Entscheidungsprozess, ob auf Basis der entsprechenden Trigger-Kombination eine zusätzliche kombinierte Trigger-Definition zu ergänzen ist, mit einfließt. Die Häufigkeit des Auftretens der entsprechenden Trigger kann sich wie zuvor im Fall des

absoluten Häufigkeitswerts auf ein Auftreten der entsprechenden Trigger in allen Datensätzen, auf ein durchschnittliches Auftreten in allen Datensätzen, auf ein häufigstes Auftreten in einem der Datensätze und/oder auf ein minimales Auftreten in einem der Datensätze beziehen.

Ausführungsformen können den Vorteil haben, dass der relative Häufigkeitswert umso höher gewählt wird, umso höher die Häufigkeiten des Auftretens des ein oder der mehreren entsprechenden von der Trigger-Kombination umfassten Trigger ist. Somit kann vermieden werden, dass eine Trigger-Definition auf der Basis einer Trigger-Kombination erzeugt wird, deren Auftreten zufällig ist, d.h. deren Trigger zufällig von demselben Datensatz umfasst sind, ohne dass dies einen Zusammenhang der entsprechenden Trigger anzeigt.

Nach Ausführungsformen umfasst das Kombinationskriterium ein oder mehrere Bedingungen an relative Positionen der Trigger der entsprechenden Trigger-Kombination zueinander innerhalb eines der Datensätze. Ausführungsformen können den Vorteil haben, dass für das Kombinationskriterium eine relative Position der Trigger der entsprechenden Trigger-Kombination innerhalb des Datensatzes berücksichtigt wird. Eine entsprechende relative Position von Daten innerhalb von Datensätzen resultiert aus bzw. ist abhängig von Kontextzusammenhängen. Mithin lassen sich entsprechende Kontextzusammenhänge aus der relativen Position ablesen. Abhängig von der Art der von den Datensätzen umfassten Daten kann es sich bei der relativen Position um eine relative Position in einer eindimensionalen, d.h. sequenziellen, Datenstruktur, wie beispielsweise einer Text- oder Sprachdatei, einer zweidimensionalen Datenstruktur, wie beispielsweise einer Bilddatei, oder einer höher dimensionalen, beispielsweise dreidimensionalen oder n-dimensionalen, Datenstruktur handeln.

Nach Ausführungsformen umfassen die Trigger-Definitionen jeweils eine Definition einer Trigger-Struktur, welche für ein oder mehrere von der entsprechenden Trigger-Definition umfasste Trigger und ein oder mehrere gemäß der entsprechenden

Trigger-Definition einer der Klassen zuzuordnende Token relative Positionen zueinander festlegt.

Ausführungsformen können den Vorteil haben, dass eine entsprechende Trigger-Definition anhand ein oder mehrerer Trigger definiert, wie ein oder mehrere Token zu klassifizieren sind in Abhängigkeit von einer relativen Position der entsprechenden Token zu den entsprechenden Triggern. Je nach Art der von den Datensätzen umfassten Datenstrukturen kann es sich bei der entsprechenden relativen Position um eine relative Position in einem eindimensionalen, zweidimensionalen oder höher dimensionalen, beispielsweise dreidimensionalen oder n-dimensionalen, Datenraum handeln.

Nach Ausführungsformen umfassen die Festlegungen der relativen Positionen zumindest eine der folgenden Festlegungen: die ein oder mehreren zuzuordnenden Token sind nach einem von der entsprechenden Trigger-Definition umfassten Trigger angeordnet, die ein oder mehreren zuzuordnenden Token sind vor einem von der entsprechenden Trigger-Definition umfassten Trigger angeordnet, die ein oder mehreren zuzuordnenden Token sind jeweils zwischen von der entsprechenden Trigger-Definition umfassten Triggern angeordnet.

Ein Trigger kann beispielsweise eine Klassifikation vorangehendender Daten triggern, z.B. „[davor1] [Trigger1]“. In diesem Fall triggert das Auftreten des Triggers „Trigger1“ eine Klassifikation der vorangehenden Daten „davor1“. Nach Ausführungsformen ist der Trigger selbst Bestandteil der Klassifikation, d.h. klassifiziert wird die Kombination „[davor1] [Trigger1]“. Nach Ausführungsformen wird der Trigger „Trigger1“, wenn er erkannt wird, als Trigger der entsprechenden Trigger-Definition zugeordnet.

Ein Trigger kann beispielsweise eine Klassifikation nachfolgender Daten triggern, z.B. „[Trigger2] [danach1]“. In diesem Fall triggert das Auftreten des Triggers „Trigger2“ eine Klassifikation der nachfolgenden Daten „danach1“. Nach Ausführungsformen ist der Trigger selbst Bestandteil der Klassifikation, d.h. klassifiziert wird die

Kombination „[Trigger2] [danach1]“. Nach Ausführungsformen wird der Trigger „Trigger2“, wenn er erkannt wird, als Trigger der entsprechenden Trigger-Definition zugeordnet.

Ein Trigger kann beispielsweise eine Klassifikation vorangehender und nachfolgender Daten triggern, z.B. „[davor2] [Trigger3] [danach2]“. In diesem Fall triggert das Auftreten des Triggers „Trigger3“ eine Klassifikation der vorangehendenden Daten „davor2“ sowie der nachfolgenden Daten „danach2“. Nach Ausführungsformen ist der Trigger selbst Bestandteil der Klassifikation, d.h. klassifiziert wird die Kombination „[davor2] [Trigger3] [danach2]“. Nach Ausführungsformen wird der Trigger „Trigger3“, wenn er erkannt wird, als Trigger der entsprechenden Trigger-Definition zugeordnet.

Eine Kombination aus zwei oder mehr Trigger kann beispielsweise eine Klassifikation vorangehender, nachfolgender sowie zwischen den Triggern angeordneter Daten triggern, z.B. „[davor3] [Trigger4] [dazwischen1] [Trigger5] [danach3]“. In diesem Fall triggert das Auftreten der Kombination aus den Triggern „Trigger4“ und „Trigger5“ eine Klassifikation der vorangehendenden Daten „davor3“, der nachfolgenden Daten „danach3“ sowie der dazwischenstehenden Daten „dazwischen 1“. Nach Ausführungsformen sind die Trigger selbst Bestandteil der Klassifikation, d.h. klassifiziert wird die gesamte Kombination „[davor3] [Trigger4] [dazwischen1] [Trigger5] [danach3]“. Nach Ausführungsformen werden die Trigger „Trigger4“ und „Trigger5“, wenn er erkannt wird, als Trigger der entsprechenden Trigger-Definition zugeordnet.

Nach Ausführungsformen kann eine Trigger-Kombination eine beliebige Anzahl an Triggern umfassen, z.B. „[davor4] [Trigger6] [dazwischen2] [Trigger7] [...] [Trigger6+N] [dazwischen2+N] [Trigger6+(N+1)] [danach4]“. In diesem Fall triggert das Auftreten der Kombination aus den Triggern „Trigger6“ bis „Trigger6+(N+1)“ eine Klassifikation der vorangehendenden Daten „davor4“, der nachfolgenden Daten „danach4“ sowie der dazwischenstehenden Daten „dazwischen2“ bis „dazwischen2+N“. Nach Ausführungsformen sind die Trigger selbst Bestandteil der Klassifikation, d.h. klassifiziert wird die gesamte Kombination „[davor4] [Trigger6]

[dazwischen2] [Trigger7] [...] [Trigger6+N] [dazwischen2+N] [Trigger6+(N+1)] [danach4]“. Nach Ausführungsformen werden die Trigger „Trigger6“ bis „Trigger6+(N+1)“, wenn er erkannt wird, als Trigger der entsprechenden Trigger-Definition zugeordnet.

Im Falle einer Textdatei kann eine beispielhafte Trigger-Definition wie folgt aussehen: „[Identität] [Trigger1 = darf über] [Identität] [Trigger2 = und] [Identität]". Bei der Formulierung „darf über" handelt es sich um einen ersten Trigger [Trigger1] und bei der Formulierung „und" um einen zweiten Trigger [Trigger2]. Die Struktur entspricht also einer Struktur der Form [davor] [Trigger1] [dazwischen] [Trigger2] [danach]. In diesem Fall werden vorangehenden Daten [davor] als eine Identität klassifiziert, ebenso werden dazwischenstehende Daten [dazwischen] sowie nachfolgende Daten [danach] jeweils als Identitäten klassifiziert.

Ein weiteres Beispiel ist: „[Trigger = Der Kunde trägt den Schaden,] [Bedingung]". Bei der Formulierung „Der Kunde trägt den Schaden," handelt es sich um einen Trigger [Trigger]. Die Struktur entspricht mithin der Struktur [Trigger] [danach]. In diesem Fall werden die nachfolgenden Daten [danach] als eine Bedingung klassifiziert.

Ein weiteres Beispiel ist: „[Trigger1 = zwischen] [Identität] [Trigger2 = und] [Identität]". Bei der Formulierung „zwischen" handelt es sich um einen ersten Trigger [Trigger1] und bei der Formulierung „und" um einen zweiten Trigger [Trigger2]. Die Struktur entspricht mithin der Form [Trigger1] [dazwischen] [Trigger2] [danach]. Umfasst ein Datensatz beispielsweise die Formulierung: „Die Geschäftsbeziehung zwischen dem Kunden und der Bank [...]", so werden die Token „zwischen“ und „und“ als Trigger identifiziert. Anhand dieser Trigger-Kombination werden die dazwischenstehenden Token „dem Kunden“ als Identität klassifiziert, ebenso wie die nachfolgenden Token „der Bank“. Für die vorangehenden Token „Die Geschäftsbeziehung“ wird durch die Trigger keine Klassifikation getriggert. Mithin werden die vorangehenden Token als unbekannte Daten der Auffangklasse klassifiziert.

Nach Ausführungsformen kann eine Trigger-Definition festlegen, dass ein Token, welches sich innerhalb eines Radius um einen bestimmten Trigger in einem n-dimensionalen Datenraum befindet, einer bestimmten Klasse zuzuordnen ist. Nach Ausführungsformen kann neben dem Abstand des Tokens von dem Trigger zudem maßgeblich für die Klassenzuordnung sein, in welcher Raumrichtung der Token von dem Trigger entsprechend beanstandet ist. Dies kann beispielsweise durch einen Vektor definiert werden, welche die relative Position des Tokens zu dem Trigger definiert. Nach Ausführungsformen kann eine Trigger-Definition festlegen, dass ein Token, welches innerhalb einer Mehrzahl von Radien um jeweils einen Trigger einer Mehrzahl von Trigger angeordnet ist, einer bestimmten Klasse zuzuordnen ist. Hierbei überschneiden sich die von den einzelnen Radien begrenzten n-dimensionalen Bereiche und begrenzen einen n-dimensionalen oder niedriger dimensionalen Schnittbereich in dem n-dimensionalen Datenraum. Ein Token, welches Bestandteil dieses n-dimensionalen oder niedriger dimensionalen Schnittbereich ist, wird beispielsweise einer bestimmten Klasse zugeordnet.

Nach Ausführungsformen ist für die Trigger gemäß den Trigger-Definitionen jeweils ein maximaler Trigger-Abstand festgelegt, welcher einen maximalen Abstand relativ zu dem entsprechenden Trigger definiert, auf welche eine Trigger-Wirkung des Triggers beschränkt ist.

Ausführungsformen können den Vorteil haben, dass es sich bei dem entsprechenden maximalen Abstand um einen Radius um den entsprechenden Trigger in einem n-dimensionalen Datenraum handelt. Im Falle eines eindimensionalen Datenraums beschränkt sich die Trigger-Wirkung auf den entsprechenden maximalen Trigger-Abstand vor und hinter dem entsprechenden Trigger. Im Fall eines zweidimensionalen Datenraums beschränkt sich die Trigger-Wirkung auf eine zweidimensionale Kreisfläche um den entsprechenden Trigger herum. Im Falle eines dreidimensionalen Datenraums beschränkt sich die Trigger-Wirkung auf ein Kugelvolumen um den entsprechenden Trigger herum. Im Falle eines n-dimensionalen Datenraums beschränkt sich die Trigger-Wirkung auf ein Volumen einer n-dimensionalen Kugel um den entsprechenden Trigger herum.

Nach Ausführungsformen kann der maximale Abstand von der Raumrichtung abhängen und in unterschiedlichen Raumrichtungen unterschiedlich groß festgelegt sein.

Nach Ausführungsformen ist der maximale Trigger-Abstand für alle Trigger identisch. Nach Ausführungsformen ist der maximale Trigger-Abstand für eine Teilmenge der Trigger identisch. Nach Ausführungsformen ist der maximale Trigger-Abstand für jeden Trigger jeweils individuell bestimmt. Bei dem entsprechenden maximalen Trigger-Abstand kann es sich je nach Art der Daten um einen Abstand in einer bestimmten Einheit handeln. Beispielsweise handelt es sich bei einer zeitlichen sequenziellen Abfolge um einen zeitlichen Abstand gemessen in einer zeitlichen Einheit, wie etwa Millisekunden, Sekunden oder Minuten. Beispielsweise handelt es sich bei einem eindimensionalen, zweidimensionalen oder dreidimensionalen räumlichen Datenstruktur um einen räumlichen Abstand in einer räumlichen Einheit, wie etwa Millimeter, Zentimeter, Dezimeter oder Meter. Im Falle von Bild- oder Videodaten kann der Abstand beispielsweise auf Pixel oder Voxel beruhen. Somit kann es sich bei einem entsprechenden Abstand beispielsweise um eine Pixelzahl oder eine Voxelzahl handeln.

Nach Ausführungsformen handelt es sich bei dem Abstand um einen logischen Abstand. Dieser kann beispielsweise auf elementaren Datenelementen beruhen, wie beispielsweise Elementarzeichen. Somit kann es sich bei einem entsprechenden Abstand beispielsweise um eine Zeichenzahl handeln. Ferner kann es sich bei dem entsprechenden Abstand um eine Anzahl aus elementaren Datenelementen um zusammengesetzte Elemente handeln, wie beispielsweise einer Wortanzahl. Beispielsweise ist die Anzahl auf eine bestimmte Wortart beschränkt. Ferner kann der Abstand durch logische Elemente in der Datenstruktur begrenzt werden, wie beispielsweise ein Interpunktionszeichen und/oder einen Trigger.

Nach Ausführungsformen umfasst das Verfahren ferner:

- Ergänzen des vortrainierten Lernmoduls um ein oder mehrere zusätzliche Trigger-Definitionen, welche zusätzliche Trigger für ein Ersetzen von Zuordnungen von Token in dem Index zu der Auffangklasse durch Zuordnungen zu ein oder mehreren Klassen einer zweiten Gruppe von Klassen im Zuge eines Reklassifizierens definieren,
- Reklassifizieren von ein oder mehreren der Auffangklasse zugeordneten Token in dem Index, welche die zusätzlichen Trigger-Definitionen als zusätzliche Trigger definierten, wobei das Reklassifizieren durch das Lernmodul ein Ersetzen der Zuordnung zu der Auffangklasse durch eine Zuordnung zu der entsprechen zusätzlichen Trigger-Definition umfasst, welche das entsprechende Token als zusätzlichen Trigger umfasst,
- Verwenden der zusätzlichen Trigger zum Reklassifizieren von ein oder mehreren der Auffangklasse zugeordneten Token in dem Index zu ein oder mehreren Klassen der zweiten Gruppe von Klassen durch das Lernmodul, falls die entsprechenden der Auffangklasse zugeordneten Token von einem der Datensätze in einer Kombination mit ein oder mehreren der zusätzlichen Triggern umfasst sind und die entsprechenden zusätzlichen Trigger gemäß der entsprechenden zusätzlichen Trigger-Definition eine entsprechende Zuordnung zu den ein oder mehreren Klassen der zweiten Gruppe von Klassen triggern.

Ausführungsformen können den Vorteil haben, dass durch das Ergänzen des Lernmoduls um zusätzliche Trigger-Definitionen die Anzahl der Token, welche der Auffangklasse zugeordnet sind, reduziert werden kann. Zusätzliche Trigger-Definitionen können gezielt ergänzt werden, um solche Token zu reklassifizieren, welche der Auffangklasse zugeordnet sind. Mithin kann das Ergänzen zusätzlicher Trigger-Definitionen in Abhängigkeit von den Datensätzen, welche die Datenbank umfasst, und den von diesen umfassten unbekannten Daten erfolgen.

Beispielsweise werden zusätzliche Trigger-Definitionen ergänzt, bis alle Token der Auffangklasse reklassifiziert sind. Nach Ausführungsformen werden entsprechende zusätzliche Trigger-Definitionen nach vordefinierten Intervallen ergänzt. Entsprechende vordefinierte Intervalle sind beispielsweise zeitlich definiert, basierend auf

der Anzahl der von der Auffangklasse umfassten Token, der in der Datenbank gespeicherten Datenmenge und/oder der zu der Datenbank seit der letzten Ergänzung hinzugefügten Datenmenge.

Nach Ausführungsformen umfasst die zweite Gruppe von den Klassen der ersten Gruppe verschiedene Klassen. Ausführungsformen können den Vorteil haben, dass zusätzliche Klassen definiert werden, sodass solche die Token der Auffangklasse klassifiziert werden können, für welche die Meta- bzw. Kontextinformationen entsprechend den Klassen der ersten Gruppe nicht anwendbar sind. Vielmehr können durch die Klassen der zweiten Gruppe zusätzliche Meta- bzw. Kontextinformationen definiert und verwendet werden.

Nach Ausführungsformen sind ein oder mehrere Klassen der zweiten Gruppe jeweils identisch mit einer der Klassen der ersten Gruppe. Ausführungsformen können den Vorteil haben, dass durch die zusätzlichen Trigger-Definitionen Trigger bereitgestellt werden, welche eine Zuordnung der Token der Auffangklasse zu Klassen der ersten Gruppe von Klassen ermöglichen.

Nach Ausführungsformen sind die zu ergänzenden Trigger-Definitionen als Ergänzungen jeweils von einer von dem Lernmodul bereits umfassten Trigger-Definition abhängig. Ausführungsformen können den Vorteil haben, dass ein oder mehrere der ergänzenden Trigger-Definitionen in Form von Ergänzungen zu den bereits umfassten Trigger-Definitionen des Lernmoduls definiert werden. Die entsprechenden ergänzenden Trigger-Definitionen erweitern beispielsweise die Trigger-Wirkung bereits bestehender Trigger-Definitionen. Nach Ausführungsformen bilden die ergänzenden Trigger-Definitionen mit den bereits bestehender Trigger-Definitionen kombinierte Trigger-Definitionen.

Nach Ausführungsformen werden die Ergänzungen einem rekursiven Schema folgend wiederholt ausgeführt, wobei die zu ergänzenden Trigger-Definitionen jeder Rekursionsstufe jeweils Ergänzungen einer Trigger-Definition einer vorangehenden Rekursionsstufe umfassen, sodass die rekursiven Ergänzungen Baumstrukturen

bilden, welche jeweils eine der vorbestimmten Trigger-Definition als Wurzelknoten umfassen.

Ausführungsformen können den Vorteil haben, dass die Trigger-Wirkung der bestehenden Trigger-Definitionen sukzessive durch ein fortschreitendes Rekursionsschema erweitert werden, bis alle Token der Auffangklasse reklassifiziert sind. Das Ergebnis der entsprechenden Ergänzungen der bereits bestehenden Trigger-Funktionen können beispielsweise Baumstrukturen sein, denen folgend eine Klassifikation von Token implementiert werden kann.

Nach Ausführungsformen werden die zu ergänzenden zusätzlichen Trigger-Definitionen von dem Lernmodul empfangen. Ausführungsformen können den Vorteil haben, dass die entsprechenden Trigger-Definitionen beispielsweise von außen, etwa durch einen Administrator, bereitgestellt werden können. Mithin hat der entsprechende Administrator stets die Möglichkeit, die Klassifikation zu steuern, zu korrigieren und zu ergänzen.

Nach Ausführungsformen kann erfolgt optional oder fakultativ ein externes Feinjustieren, beispielsweise durch einen Administrator, erfolgen. Hierbei werden nach Ausführungsformen unter Verwendung zusätzliche Trigger-Definitionen aus der Klasse der unbekannten Daten, d.h. der Auffangklasse, Token extrahiert und bestehenden Klassen zugeordnet und/oder es werden neue Klassen generiert, denen extrahierte Token zugordnet werden. Ein Administrator stellt beispielsweise für die in der Auffangklasse als unbekannte Daten klassifizierten Token anlog zu den initialen bereitgestellten Trigger-Definitionen zusätzliche Trigger-Definitionen bereit, welche auf die Auffangklasse angewandt werden. Nach Ausführungsformen werden die zusätzlichen Trigger gemäß den zusätzlichen Trigger-Definitionen ausschließlich auf die Auffangklasse angewendet sowie auf zukünftig empfangene Daten. Nach Ausführungsformen kann die Anwendung eines zusätzlichen Triggers als IF-Bedingung realisiert werden. Wurde beispielsweise auf einen Datensatz bereits ein anderer Trigger erfolgreich angewendet, z.B. ein Trigger1, und der Datensatz

umfasst zudem als unbekannt klassifizierte Daten, wo wird ein zusätzlicher Trigger, z.B. ein Trigger2, gemäß einer der zusätzlichen Trigger-Definitionen angewendete.

Dieses Feinjustieren kann als eine Rekursion mehrmals wiederholt werden. Beispielsweise wird die Rekursion solange fortgesetzt, bis Die Auffangklasse keine Token mehr umfasst, d.h. keine unbekannten Daten mehr existieren, oder die von der Auffangklasse umfasste Token-Anzahl einen vordefinierten Schwellenwert erreicht und/oder unterschreitet, d.h. eine vordefinierte Maximalzahl. Bei dem entsprechenden Schwellenwert kann es sich um einen absoluten Wert handeln, welcher von der Anzahl der von dem Index umfassten Token und der von der Datenbank umfassten Datenmenge unabhängig ist. Alternativ kann es sich bei dem entsprechenden Schwellenwert um einen relativen Wert handeln, welcher von der Anzahl der von dem Index umfassten Token und/oder der von der Datenbank umfassten Datenmenge abhängig ist

Auf diese Weise können Trigger-Bäume bzw. Entscheidungsbäume hinter den initial definierten Trigger bzw. Trigger-Definitionen entstehen, wobei die Anzahl der Ebenen von der Anzahl der Rekursionen N abhängt, z.B. ist die Anzahl der Ebenen gleich N+1. Beispielsweise bildet jeder initiale Trigger bzw. jede initiale Trigger-Definitionen einen Wurzelpunkt eines entsprechenden Trigger-Baums bzw. Entscheidungsbaum. Unter einem Entscheidungsbäumen werden hier geordnete, gerichtete Bäume verstanden, die der Darstellung von Entscheidungsregeln dienen. Umfasst ein Datensatz einen initialen Trigger, wodurch ein Teil der Token des Datensatzes klassifiziert werden kann, ohne dass dadurch zugleich alle Token des Datensatzes klassifiziert werden können, so wird geprüft, ob der Datensatz zudem einen Trigger der ersten Rekursion umfasst. Falls der Datensatz einen Trigger der ersten Rekursion umfasst, wodurch ein weiterer Teil der Token des Datensatzes klassifiziert werden kann, ohne dass dadurch zugleich alle Daten des Datensatzes klassifiziert werden können, so wird geprüft, ob der Datensatz zudem einen Trigger der zweiten Rekursion umfasst und so fort.

Nach Ausführungsformen werden die zu ergänzenden zusätzlichen Trigger-Definitionen von dem Lernmodul erstellt, welches ein statistisches Modell umfasst, wobei das statistische Modell zu einer statistischen Analyse der von der Auffangklassen umfassten Token und deren Vorkommen in den Datensätze verwendet wird, wobei das Ergebnis der statistischen Analyse zum Erstellen der zu ergänzenden zusätzlichen Trigger-Definitionen verwendet wird.

Ausführungsformen können den Vorteil haben, dass das Lernmodul selbstständig ergänzende zusätzliche Trigger-Definitionen erstellen kann. Beispielsweise erfolgt das zuvor beschriebene optionale oder fakultative Feinjustieren unter Verwendung des statistischen Modells. Durch ein Verwenden eines statistischen Modells kann die zuvor beschriebene menschliche Handlung eines Administrators ersetzt und/oder verbessert werden. Nach Ausführungsformen identifiziert das statistische Modell, z.B. durch Häufigkeitsanalysen und Korrelationsanalysen, Trigger innerhalb der unbekannten Daten, welche dann analog zu dem zuvor beschriebenen Vorgehen auf die als unbekannt klassifizierten Token angewendet werden. Nach Ausführungsformen kann zudem ein rekursives Vorgehen unter Verwendung des statistischen Modells erfolgen.

Nach Ausführungsformen umfasst das Verfahren ferner:

- Empfang einer korrigierten Trigger-Definition zum Ersetzen einer der gespeicherten Trigger-Definitionen des Lernmoduls,
- Ersetzen der entsprechenden gespeicherten Trigger-Definition durch die korrigierte Trigger-Definition,
- Reklassifizieren der unter Verwendung der entsprechenden gespeicherten Trigger-Definition klassifizierten Token, wobei das Reklassifizieren unter Verwendung der korrigierten Trigger-Definition erfolgt.

Nach Ausführungsformen kann beispielweise ein Administrator Fehler in klassifizierten Klassen erkennen und gegebenenfalls korrigieren, etwa indem er eine korrigierten Trigger-Definition, anhand derer ein Reklassifizieren von Token erfolgt. Ausführungsformen können den Vorteil haben, dass eine Korrektur von Trigger-

Definitionen zu jedem Zeitpunkt des Verfahrens ermöglicht wird. Beispielsweise kann eine Überprüfung der Trigger-Definitionen nach dem Training des Lernmoduls erfolgen. Werden Korrektur-Trigger-Definitionen identifiziert, so werden entsprechend korrigierte Trigger-Definitionen bereitgestellt.

Ausführungsformen können den Vorteil haben, dass korrigierte Trigger-Definitionen auch zu einem späteren Zeitpunkt bereitgestellt werden können, wenn Fehlklassifikationen erkannt werden. Ein administratives Eingreifen in den Lern- und Klassifizierungsprozess ist somit zu jedem Zeitpunkt möglich. Dadurch können Fehler des lernenden Systems behoben werden, ohne dass das komplette Modell umgebaut werden muss.

Nach Ausführungsformen verweisen die Zeiger, mit denen die Token in dem Index verknüpft gespeichert sind, jeweils auf ein oder mehrere der Feldwerte in den gespeicherten Datensätze.

Ausführungsformen können den Vorteil haben, dass eine feinere Granularität bei der Bestimmung des Ursprungs von Token in den Datensätzen erzielt werden kann. Eine solche feinere Granularität ermöglicht es zudem relative Beziehungen der Token innerhalb der Datensätze aufzuschlüsseln und bei einer Analyse oder sonstigen Verwendung des Index zu berücksichtigen.

Nach Ausführungsformen umfassen die Feldwerte des zusätzlichen Datensatzes Textdaten, Bilddaten, Audiodaten und/oder Videodaten. Nach Ausführungsformen ist das Verfahren beispielsweise anwendbar zur Signalverarbeitung, wie etwa 1D-Audioerkennung, 2D- und 3D-Bildverarbeitung, oder ND-Dateninput von N Sensoren etc. Ferner ist das Verfahren beispielsweise anwendbar bei einer Analyse von Stream-Daten (Bitstream bzw. Bitstrom). Ein Bitstream, auch als Bitstrom bekannt, bezeichnet hier eine Folge von Bits, die einen Informationsfluss repräsentieren, d.h. ein serielles bzw. sequentielles Signal. Ein Bitstrom ist somit eine Sequenz von Bits von unbestimmter Länge in zeitlicher Abfolge. Ein Bitstrom stellt beispielsweise einen in logische Strukturen gegliedert Datenstromes dar, der sich in grundlegendere

Kleinstrukturen wie Symbole fester Größe, d.h. Bits und Bytes, und weiter in Blöcke und Datenpakete unterschiedlicher Protokolle und Formate gliedern kann.

Nach Ausführungsformen umfasst das Erzeugen der Token ein Anwenden einer Tokenisierungslogik auf die Feldwerte des zusätzlichen Datensatzes, welche einen Volltextindizierer umfasst, der dazu konfiguriert ist, Texte in Wörter zu zerlegen und die Wörter als Token auszugeben. Ausführungsformen können den Vorteil haben, dass eine effektive Tokenisierung von Texten bzw. Textdateien implementiert werden kann. Bei entsprechenden Textdateien kann es sich um beliebige Texte handelnden. Beispielsweise kann es sich bei entsprechenden Textdateien um Messwertdateien oder Algorithmen zum Steuern von Computern und/oder technischen Anlagen handeln. Nach Ausführungsformen umfassen die Feldwerte des zusätzlichen Datensatzes Volltexte, wobei die Volltexte aus Buchstaben eines oder mehrerer Alphabete gebildete Wörter und/oder ein oder mehrere Zahlen umfassen.

Eine Volltextindizierung beinhaltet eine Zerlegung von Texten in einzelne Wörter, wobei dann die einzelnen Wörter eines Textfeldes in einem diesem Feld zugeordneten Index gespeichert werden. Volltextindexierung wird nur unterstützt, wenn das entsprechende Feld zur selektiven Speicherung eines bestimmten Datentyps, z.B. CHAR, VARCHAR oder TEXT, konfiguriert ist. Beispielsweise kann in einem Feld natürlichsprachlicher Text im JSON Format gespeichert sein.

Nach Ausführungsformen umfasst das Erzeugen der Token ein Anwenden einer Tokenisierungslogik auf die Feldwerte des zusätzlichen Datensatzes, welche einen generischen Tokenisierer umfasst, der dazu konfiguriert ist, in den Feldwerten Daten unterschiedlichen Datentyps zu erkennen und aus diesen Token in unterschiedlichen Datentypen zu erzeugen. Ausführungsformen können den Vorteil haben, dass eine effektive Tokenisierung für unterschiedliche Datentypen implementiert werden kann, wie etwa Textdaten, Bilddaten, Audiodaten und/oder Videodaten.

Nach Ausführungsformen umfasst das Verfahren ferner:

- Empfangen einer Suchanfrage, wobei die Suchanfrage einen Suchwert beinhaltet,
- Durchsuchen des Index nach dem Suchwert,
- Identifizieren eines Tokens innerhalb des Index, welcher identisch ist mit dem Suchwert,
- Analysieren von Zeigern, mit denen das identifizierte Token verknüpft ist, um ein oder mehrere der Datensätze zu bestimmen, welche ein oder mehrere Feldwerte beinhalten, aus welchen das indizierte Token erzeugt wurde,
- Zurückgeben der bestimmten Datensätze oder von ein oder mehreren Referenzen auf die bestimmten Datensätze als Antwort auf die Suchanfrage.

Ausführungsformen können den Vorteil haben, dass der Index für effektive Suchen in den Datensätzen verwendet werden kann, obwohl diese in ihrer ursprünglichen Form gespeichert sind. So kann beispielsweise das Lernmodul unter Verwendung entsprechender Suchanfragen Muster und/oder Gesetzmäßigkeiten innerhalb der Datensätze suchen.

Nach Ausführungsformen speichert der Index sämtliche aus den Feldwerten der Datensätze einer Datenbank erzeugte Token so, dass der Index jedes Token nur einmal enthält. Jedes Token beinhaltet Zeiger auf ein oder mehrere der Datensätze, aus deren Feldwerten es erzeugt wurde. Wenn ein erfindungsgemäß erzeugter Index also nach einem bestimmten Suchwert durchsucht wird und als Ergebnis der Suche ein in dem Index gespeichertes Token identifiziert wird, welches identisch ist mit dem Suchwert, so verweist dieses Token mittels Zeigern auf sämtliche Datensätze, die dieses Token zumindest einmal in zumindest einem ihrer Feldwerte enthalten und die bei der Erstellung des Index herangezogen wurden. Die Datensätze, die also einen „Treffer“ im Hinblick auf den Suchwert darstellen, können anhand der Verweise sehr schnell identifiziert und zurückgegeben werden, ohne dass ein sequenzieller Suchlauf über sämtliche Datensätze notwendig wäre.

Nach Ausführungsformen umfasst der Suchwert ferner eine Klassenzuordnung und das Identifizieren des Tokens innerhalb des Index erfordert ferner, dass das

identifizierte Token dieselbe Klassenzuordnung aufweist. Ausführungsformen könne den Vorteil haben, dass Klassenzuordnungen und dadurch mit den Klassenzuordnungen indizierte Meta- bzw. Kontextinformationen in den Suchanfragen berücksichtigt werden können.

Nach Ausführungsformen sind Trigger in dem Index mit einem Flag gekennzeichnet. Nach Ausführungsformen umfasst der Suchwert ferner eine Zuordnung zu einer Trigger-Definition und/oder ein einen Trigger kennzeichnendes Flag und das Identifizieren des Tokens innerhalb des Index erfordert ferner, dass das identifizierte Token derselben Trigger-Definition zugeordnet ist und/oder dasselbe Flag aufweist.

Nach Ausführungsformen werden Token, welche der Auffangklasse zugeordnet sind, von der Suche ausgeschlossen. Ausführungsformen können den Vorteil haben, dass die resultierenden Suchergebnisse ein hohes Maß an Zuverlässigkeit aufweisen, da unbekannte Daten von der Suche ausgeschlossen sind.

Nach Ausführungsformen umfasst das Verfahren ferner das Vortrainieren des Lernmoduls. Das Vortrainieren umfasst:

- Empfangen der Mehrzahl von vorbestimmten Trigger-Definitionen durch das Lernmodul, welche die Trigger für das Zuordnen von Token zu den Klassen der ersten Gruppe von Klassen definieren,
- Speichern der empfangenen Mehrzahl von vorbestimmten Trigger-Definitionen durch das Lernmodul.

Nach Ausführungsformen werden durch die vorbestimmten Trigger-Definitionen initial Trigger definiert, die dazu verwendet werden empfangene Daten zu strukturieren bzw. klassifizieren. Nach Ausführungsformen werden, bevor Daten in die Datenbank geladen werden, die initialen Trigger konkret definiert, d.h. vorbestimmten Trigger-Definitionen vorgegeben. Werden Daten geladen, so ermöglichen diese initialen Trigger eine erste Klassifizierung nach bekannten Klassen sowie unbekannten Daten, welche der Auffangklasse zugeordnet werden.

Nach Ausführungsformen umfasst das Vortrainieren ferner:

- Extrahieren der Trigger aus der gespeicherten Mehrzahl von vorbestimmten Trigger-Definitionen durch das Lernmodul,
- Zuordnen der initialen Token durch das Lernmodul jeweils zu der Trigger-Definition, welche das entsprechende initiale Token als Trigger umfasst,
- Erzeugen des durchsuchbaren Index unter Verwendung der Mehrzahl von initialen Token durch das Multi-Modell-Datenbankmanagementsystem in dem weiteren Datenmodell, wobei der erzeugte Index die initialen Token umfasst, wobei jedes der initialen Token in dem Index jeweils eine Zuordnung aufweist zu der Trigger-Definition, welche das entsprechende initiale Token als Trigger umfasst.

Nach der Definition der Trigger, werden Daten, z.B. Textdaten, Audiodaten, Bilddaten, Videodaten oder N-Dimensionale Daten von N Sensoren, in die Datenbank geladen und die Trigger angewandt, um die Daten automatisch zu klassifizieren. Dadurch entsteht eine Fragmentierung der Daten in Trigger, bekannte Klassen, d.h. durch die Trigger-Definitionen definieret Klassen, und in unbekannte Daten.

Ausführungsformen können den Vorteil haben, dass das Lernmodul auf diese Weise effektiv vortrainiert werden kann auf Basis der vorbestimmten Trigger-Definitionen.

Dieses vorbestimmten Trigger-Definitionen können als Grundlage für ein Erlangen weiterer Trigger-Definitionen dienen, etwa durch ein Kombinieren von Trigger-Definitionen. Es erfolgt beispielsweise eine automatische Lernphase des Datenbanksystems bzw. des Lernmoduls, welche ein Kombinieren von der initialen Triggern umfasst. Somit können die initial geladenen Trigger wie zuvor beschrieben basierend auf den von den Datensätzen umfassten Daten kombiniert und damit die Anzahl an zur Verfügung stehenden Trigger-Definitionen erhöht werden. Zudem kann ein Kennzeichnen von bereits klassifizierten Token-Kombination erfolgen. Dies dient dem Zweck, dass identische Daten die später in die Datenbank geladen werden,

nicht erneut klassifiziert werden müssen, sondern in dem System bereits als „bekannt" markiert sind.

Nach Ausführungsformen umfasst das Erzeugen eines der zusätzlichen Token ein Verwenden eines der Feldwerte des zusätzlichen Datensatzes in seiner Gesamtheit als das entsprechende zusätzliche Token. Es ist durchaus möglich, dass der Index auch Token aus Feldern beinhaltet, auf die keine Tokenisierung angewandt wird bzw. deren Inhalt sich schlichtweg nicht in einzelne Token aufteilen lässt. Nach Ausführungsformen umfasst das Erzeugen eines der zusätzlichen Token ein Aufteilen eines der zusätzlichen Feldwerte des zusätzlichen Datensatzes in eine Mehrzahl von Teilfeldwerten und eine Verwenden eines der Teilfeldwerte als das entsprechende zusätzliche Token. Ausführungsformen können den Vorteil haben, dass die Granularität der verwendeten Daten bzw. der Tokenisierung unabhängig von der Granularität der Felder angepasst werden kann.

Nach Ausführungsformen speichert der Index sämtliche aus den Feldwerten der gespeicherten Datensätze erzeugten Token so, dass der Index jedes Token für jede der Token-Zuordnungen des entsprechenden Tokens genau einmal enthält.

Nach Ausführungsformen ist das weitere Datenmodell so strukturiert, dass die in dem weiteren Datenmodell gespeicherten Token und Token-Zuordnungen des Index die fünfte und/oder sechste Normalform erfüllen. Ausführungsformen können den Vorteil haben, dass Redundanzen vermieden werden können.

Nach Ausführungsformen können die Token, die Klassen-Zuordnungen und die Zuordnung zu den Trigger-Definitionen in Form von Relationen oder äquivalenten Strukturen gespeichert sein. Unter einer Relation wird hier im Sinn der relationalen Datenbanktheorie eine Menge von Tupel. Ein Tupel ist eine Menge von Attributwerten. Ein Attribut bezeichnet einen Datentyp bzw. eine ein oder mehreren Daten zugeordnete Eigenschaft. Dabei bestimmt die Anzahl der Attribute den Grad, die Anzahl der Tupel die Kardinalität einer Relation.

Unter einer Normalisierung, insbesondere unter einer Normalisierung eines relationales Datenmodells, wird eine Aufteilung von Attributen in eine Mehrzahl von Relationen gemäß einer Normalisierungsregeln verstanden, sodass Redundanzen reduziert bzw. minimiert werden. Ein relationales Datenmodell lässt sich beispielsweise in tabellenartigen Datenstrukturen implementieren, in denen die Relationen in Form von Tabellen, die Attribute in Form von Tabellenspalten und die Tupel in Form von Tabellenzeilen realisiert sind.

Datenredundanzen haben die Gefahr, dass es bei Änderungen von Daten, welche mehrfach umfasst sind, zu Inkonsistenzen kommen kann und Anomalien auftreten. Ferner steigt durch Redundanzen unnötiger Weise der Speicherplatzbedarf. Durch eine Normalisierung können solche Redundanzen verringert bzw. minimiert werden. Ein relationales Datenmodell kann beispielsweise in eine Normalform gebracht werden, indem die Relationen des Datenschemas fortschreitend anhand der für die entsprechende Normalform geltenden funktionalen Abhängigkeiten in einfachere Relationen zerlegt.

Es können beispielsweise folgende Normalformen unterschieden werden: 1. Normalform (1NF), 2. Normalform (2NF), 3. Normalform (3NF), Boyce-Codd-Normalform (BCNF), 4. Normalform (4NF), 5. Normalform (5NF), 6. Normalform (6NF). Die Normalisierungskriterien nehmen dabei von Normalform zu Normalform zu und umfassen jeweils die Normalisierungskriterien der vorhergehenden Normalformen, d.h. $1NF \subseteq 2NF \subseteq 3NF \subseteq BCNF \subseteq 4NF \subseteq 5NF \subseteq 6NF$.

Eine Relation ist in der ersten Normalform, falls jedes Attribut der Relation einen atomaren Wertebereich besitzt und die Relation frei von Wiederholungsgruppen ist. Unter atomar wird hier ein Ausschluss von zusammengesetzten, mengenwertigen oder geschachtelten Wertebereichen für die Attribute, d.h. relationenwertigen Attributwertebereichen, verstanden. Eine Freiheit von Wiederholungsgruppen erfordert es, dass Attribute, die gleiche bzw. gleichartige Information enthalten, in unterschiedliche Relationen ausgelagert werden.

Eine Relation ist in der zweiten Normalform, wenn sie die Anforderungen der ersten Normalform erfüllt und kein Nichtprimärattribut funktional von einer echten Teilmenge eines Schlüsselkandidaten abhängt. Ein Nichtprimärattribut ist ein Attribut, welches nicht Teil eines Schlüsselkandidaten ist. Das bedeutet, dass jedes Nichtprimärattribut jeweils von allen ganzen Schlüsseln abhängig und nicht nur von einem Teil eines Schlüssels. Relationen in der ersten Normalform, deren Schlüsselkandidaten nicht zusammengesetzt sind, sondern aus jeweils einem einzelnen Attribut bestehen, erfüllen mithin automatisch die zweite Normalform. Unter einem Schlüsselkandidaten wird hierbei eine minimale Menge von Attributen verstanden, welche die Tupel einer Relation eindeutig identifiziert.

Eine Relation ist in der dritten Normalform, wenn sie die Anforderungen der zweiten Normalform erfüllt und kein Nichtschlüsselattribut von einem Schlüsselkandidaten transitiv abhängt. Ein Attribut ist von einem Schlüsselkandidaten transitiv abhängig, wenn das entsprechende Attribut von dem entsprechenden Schlüsselkandidaten über ein weiteres Attribut abhängig ist.

Eine Relation ist in der Boyce-Codd-Normalform, wenn sie die Anforderungen der dritten Normalform erfüllt und jede Determinante ein Superschlüssel ist. Unter einer Determinante wird hier eine Attributmenge verstanden, von welcher andere Attribute funktional abhängen sind. Eine Determinante beschreibt somit die Abhängigkeit zwischen Attributen einer Relation und legt fest, welche Attributmengen den Wert der übrigen Attribute bestimmen. Ein Superschlüssel ist eine Menge von Attributen in einer Relation, welche die Tupel in dieser Relation eindeutig identifizieren. Mithin umfassen die Attribute dieser Menge bei paarweise ausgewählten Tupeln immer unterschiedliche Werte. Schlüsselkandidat ist mithin eine minimale Teilmenge der Attribute eines Superschlüssels, welche die Identifizierung der Tupel ermöglicht.

Eine Relation ist in der vierten Normalform, wenn sie die Anforderungen der Boyce-Codd-Normalform erfüllt und keine nichttrivialen mehrwertigen Abhängigkeiten umfasst.

Eine Relation ist in der fünften Normalform, wenn sie die Anforderungen der vierten Normalform erfüllt und keine mehrwertigen Abhängigkeiten umfasst, die voneinander abhängig sind. Die fünfte Normalform liegt somit vor, falls jeder nichttriviale Verbund-Abhängigkeit durch die Schlüsselkandidaten impliziert ist. Eine Verbund-Abhängigkeit ist durch die Schlüsselkandidaten der Ausgangsrelation impliziert, wenn jede Relation der Menge von Relationen ein Superschlüssel der Ausgangsrelation ist.

Eine Relation ist in der sechsten Normalform, wenn sie die Anforderungen der fünften Normalform erfüllt und keine nichttrivialen Verbund-Abhängigkeiten umfasst. Eine Relation genügt einer Verbund-Abhängigkeit (join dependency) von einer Mehrzahl von Relationen, falls sich die Relation als Ausgangsrelation verlustlos in die entsprechende Menge von Relationen zerlegen lässt. Die Verbund-Abhängigkeit ist trivial, falls eine der Relationen der Menge von Relationen alle Attribute der Ausgangsrelation aufweist.

Nach Ausführungsformen handelt es sich zumindest bei dem von dem Multi-Modell-Datenbankmanagementsystem zum Speichern der Datensätze verwendeten dokumentenbasierten Datenmodellen um ein NoSQL-Datenmodellen. Nach Ausführungsformen handelt es sich bei dem DBMS um ein NoSQL-DBMS. Dies kann vorteilhaft sein, dass da sich herausgestellt hat, dass insbesondere NoSQL-DBMS, die oftmals eine flexiblere Struktur aufweisen als klassische SQL-basierte DBMSs. Aufgrund der Flexibilität ihrer Struktur eignen in sich NoSQL-DBMSs also ganz besonders für die Verwaltung und Speicherung von Datensätzen, aus welchen ein Index gemäß Ausführungsformen der Erfindung erstellt werden kann.

Nach Ausführungsformen weist der Index die Struktur eines Baums auf, insbesondere eines B^+-Baums. Ausführungsformen können den Vorteil haben, dass eine Baumstruktur insbesondere die Struktur eines B^+-Baums, eine besonders effiziente und schnelle Suche nach den in dem Index gespeicherten Token ermöglicht. Unter einem B^+-Baum wird eine Daten- und/oder Indexstruktur, welche eine Erweiterung eines B-Baumes darstellt. Bei einem B^+-Baum werden die eigentlichen

Datenelemente nur in den Blattknoten gespeichert, während die inneren Knoten lediglich Schlüssel enthalten.

Nach Ausführungsformen umfassen mehrerer der in einem dokumentenorientierten Datenmodell gespeicherten Datensätze jeweils eine unterschiedliche Anzahl an Felder. Ausführungsformen können den Vorteil haben, dass Datensätze unterschiedlicher Größen und Strukturierung bzw. Granularität verarbeitet werden können.

Nach Ausführungsformen weisen die Felder jeweils ein gemeinsames, generisches Datenformat auf. Ausführungsformen können den Vorteil haben, dass, da in einem bestimmten Feld eine große Anzahl an unterschiedlichen Datentypen gespeichert werden können. Ein Nutzer bzw. ein Applikationsprogramm, welches Datensätze in der Datenbank speichern will, muss sich also nicht um die Konsistenz und Passung von Datentypen kümmern. Mithin wird kann ein hoher Grad an Flexibilität bezüglich der Struktur und des Umfangs der Datensätze, die von dem Multi-Modell-Datenbankmanagementsystem verwaltet und gespeichert werden können, geboten werden.

Nach Ausführungsformen ist das Lernmodul bzw. das von diesem implementierte maschinelle Lernen konfiguriert für eine Datenextraktion, Konsistenzprüfung, Bilderkennung, Spracherkennung, Sprachsteuerung, Vorrichtungsüberwachung und/oder autonome Vorrichtungssteuerung. Dies kann beispielsweise bereits in der Klassifizierung der Token bestehen, wobei der Auffangklasse als unbekannte Daten zugeordnete Token beispielsweise als ein Hinweis etwa auf eine potentielle Fehlfunktion betrachtet werden. Beispielsweise kann dies auf dem Index mit den Token und deren Meta- bzw. Kontextinformationen beruhen, welche als Grundlage für einen darauf angewandten zusätzlichen Algorithmus zum maschinellen Lernen dienen. Nach Ausführungsformen wird hierzu die Auffangklasse durch ein Ergänzen zusätzlicher Trigger-Definitionen geleert, sodass zu allen Token des Datenbanksystems Meta- bzw. Kontextinformationen bereitgestellt werden.

Eine Datenextraktion kann beispielsweise ein Erkennen und Extrahieren eines Musters in einer Text-, Bild-, Audio- oder Videodatei umfassen. Dieses Muster kann beispielsweise durch eine Trigger-Definition definiert sein oder in den klassifizierten Daten erfasst werden. Ein entsprechendes Muster kann beispielsweise ein vorbestimmtes in Form von Sensorwerten erfasstes Ereignis sein, etwa eine Person in einem Wirkbereich einer Vorrichtung.

Eine Konsistenzprüfung kann beispielsweise eine Konsistenzprüfung in einer Text-, Bild-, Audio- oder Videodatei umfassen. Hierbei wird beispielsweise geprüft, ob die entsprechenden Daten unbekannte und damit inkonsistente Daten umfassen, von den restlichen Daten stark abweichende Daten oder als inkonsistent explizit vordefinierte Daten umfassen. Eine entsprechende Konsistenzprüfung kann beispielsweise zur Fehlerprüfung von Steuerungsalgorithmen von Vorrichtungen dienen, zur Fehlfunktionserkennung unter Verwendung von Messdaten einer Funktion einer Vorrichtung oder zum Erkennen von Fehlern in Textdateien, etwa in Form einer Rechtschreibprüfung.

Eine Bilderkennung kann einem Erkennen von Gegenständen, Ereignissen oder Merkmalen in Bild- oder Videodateien dienen. Beispielsweise werden Kontextinformationen zu dem visuell Dargestellten erfasst und/oder dargestellt. Dies kann beispielsweise eine visuelle Darstellung von Informationen, also die Ergänzung von Bildern oder Videos mit computergenerierten Zusatzinformationen oder virtuellen Objekten mittels Einblendung/Überlagerung, umfassen. Ein solches Verfahren wird allgemeine als erweiterter Realität oder Augmented Reality bezeichnet. Ferner kann eine Bilderkennung auf annotierten Bild- oder Videodateien basieren.

Eine Spracherkennung kann einem Erkennen von Sprache in Audiodateien oder Videodateien, etwa zur Sprachsteuerung oder zum Überführen von Sprache in Textform, dienen.

Eine Mustererkennung in Text-, Bild-, Audio- oder Videodatei kann einer Vorrichtungsüberwachung dienen. Insbesondere können so auftretende oder drohende

Fehlfunktionen erkannt werden. Dies kann der Sicherheit dienen und ermöglicht eine vorausschauende Wartung (Predictiv Maintenance) der entsprechenden Vorrichtung, da potentielle Probleme frühzeitig erkannt werden können. Bei einer entsprechenden Textdatei handelt es sich beispielswiese um einen Datensatz mit Sensormesswerten. Basierend auf einer Vorrichtungsüberwachung kann zudem eine autonome Vorrichtungssteuerung implementiert werden, etwa eine autonome Steuerung von Fahrzeugen, Robotern oder Industrieanlagen.

Unter einer „Vorrichtung“ wird hier allgemein eine technische Vorrichtung verstanden mit Sensoren zur Erfassung von Zustandsdaten der Vorrichtung und einem Vorrichtungscomputersystem zum Protokollieren der erfassten Zustandsdaten. Die Vorrichtung kann auch in dem entsprechenden Computersystem mit Sensorik bestehen. Beispielsweise handelt es sich bei den empfangenen Datensätzen um von einer Vorrichtungscomputersystem unter Verwendung der Sensoren erfasste Datensätze. Computersystem zum maschinellen Lernen Eine Vorrichtung umfasst beispielsweise ein Fahrzeug, eine Anlage, wie etwa eine Produktionsanlage, eine Verarbeitungsanlage, eine Förderanlage, eine Energiegewinnungsanlage, eine Wärmegewinnungsanlage, eine Steuerungsanlage, eine Überwachungsanlage, etc. sein.

Unter einem „Fahrzeug“ wird hier ein mobiles Verkehrsmittel verstanden. Ein solches Verkehrsmittel kann beispielsweise dem Transport von Gütern (Güterverkehr), von Werkzeugen (Maschinen oder Hilfsmittel) oder Personen (Personenverkehr) dienen. Fahrzeuge umfassen insbesondere auch motorisierte Verkehrsmittel. Bei einem Fahrzeug kann es sich beispielsweise um ein Landfahrzeug, ein Wasserfahrzeug und/oder ein Luftfahrzeug handeln. Ein Landfahrzeug kann beispielsweise sein: ein Automobil, wie etwa ein Personenkraftwagen, Omnibus oder ein Lastkraftwagen, ein motorbetriebenes Zweirad, wie etwa ein Motorrad, Kleinkraftrad, Motorroller oder Motorfahrrad, ein landwirtschaftlicher Traktor, Gabelstapler, Golfmobil, Autokran. Darüber hinaus kann es sich bei einem Landfahrzeug auch um ein Schienen gebundenes Fahrzeug handeln. Wasserfahrzeug kann beispielsweise sein: ein Schiff oder Boot. Ferner kann ein Luftfahrzeug beispielsweise sein: ein Flugzeug

oder Hubschrauber. Unter einem Fahrzeug wird insbesondere auch ein Kraftfahrzeug verstanden.

Nach Ausführungsformen umfasst die Vorrichtung zumindest einen Sensor zum Erfassen von Zustandsdaten der Vorrichtung. Die Zustandsdaten der Vorrichtung werden durch das Vorrichtungscomputersystem von dem zumindest einen Sensor empfangen. Nach Ausführungsformen umfasst die Vorrichtung eine Mehrzahl von Sensoren zum Erfassen von Zustandsdaten der Vorrichtung. Ausführungsformen können den Vorteil haben, dass die vorrichtungseigene Sensorik genutzt werden kann, um den Zustand der Vorrichtung zu erfassen. Der Zustand der Vorrichtung kann beispielsweise beschrieben werden durch Angaben zu Kenngrößen des aktuellen Leistungsvermögens der Vorrichtung, wie etwa Kilometerstand bei einem Fahrzeug, Verbrauchswerte, Leistungswerte, Fehlermeldungen, Ergebnisse vordefinierter Prüfprotokolle und/oder Kennungen von Komponenten der Vorrichtung.

Kenngrößen des aktuellen Leistungsvermögens eines Fahrzeugs können zum Beispiel Drehzahl, Geschwindigkeit, Kraftstoffverbrauch, Abgaswerte, Getriebegang sein.

Unter einem „Sensor“ wird hier ein Element zum Erfassen von Messdaten verstanden. Messdaten sind Daten, welche physikalische oder chemische Eigenschaften eines Messobjekts, wie beispielsweise Wärmemenge, Temperatur, Feuchtigkeit, Druck, Durchflussmenge, Schallfeldgrößen, Helligkeit, Beschleunigung, pH-Wert, Ionenstärke, elektrochemisches Potential, und/oder dessen stoffliche Beschaffenheit qualitativ oder quantitativ wiedergeben. Messdaten werden mittels physikalischer oder chemischer Effekte erfasst und in ein elektronisch weiterverarbeitbares elektrisches Signal umgeformt. Ferner können Messdaten Zustände und/oder Zustandsänderung von elektronischen Geräten durch Außeneinflüsse und/oder in Folge einer Benutzung durch einen Nutzer wiedergeben.

Sensoren zum Erfassen von Zustandsdaten in einem Fahrzeug können beispielsweise umfassen: Kurbelwellensensor, Nockenwellensensor, Luftmassenmesser,

Lufttemperatursensor, Kühlwassertemperatursensor, Drosselklappensensor, Klopfsensor, Getriebesensor, Wegstreckensensor, Getriebesensor, Niveausensor, Bremsverschleißsensor, Achslastsensor, Lenkwinkelsensor. Diese Sensoren erfassen und überwachen das Fahrverhalten des Fahrzeugs. Aus Abweichungen von Sollwerten und/oder einem Auftreten von bestimmten Mustern lassen sich Fehlfunktionen erkennen und identifizieren. Teils lassen sich auch konkrete Fehlerursachen, wie ausgefallene Komponenten des Fahrzeugs, identifizieren. Sensoren können zudem auch Kennungen elektronischer Komponenten, die in das Fahrzeug eingebaut sind abfragen, um deren Identität zu prüfen.

Ausführungsformen umfassen ein Computersystem zum maschinellen Lernen, wobei das Computersystem ein oder mehrere Prozessoren, eine Datenbank, welche von ein oder mehrere Datenspeichermedien bereitgestellt wird, ein Multi-Modell-Datenbankmanagementsystem, welches die Datenbank verwaltet und dazu konfiguriert ist, eine Mehrzahl von Datensätze in einem dokumentenorientierten Datenmodell in den Datenspeichermedien zu speichern, wobei die gespeicherten Datensätze jeweils ein oder mehreren Feldwerte umfassen, wobei die einzelnen Feldwerte der gespeicherten Datensätze jeweils in einem Feld gespeichert werden, wobei die Feldwerte der gespeicherten Datensätze jeweils ein oder mehreren Feldtypen einer Mehrzahl unterschiedlicher Feldtypen zugeordnet sind, ein vortrainiertes Lernmodul zum maschinellen Lernen und eine Programmlogik umfasst.

Die Datenbank umfasst ferner einen durchsuchbaren Index, welcher in einem weiteren Datenmodell gespeichert ist, wobei der Index eine Mehrzahl von aus den Feldwerten der gespeicherten Datensätze erzeugten Token umfasst, wobei Token in dem Index jeweils mit einem oder mehreren Zeigern auf ein oder mehrere der in dem dokumentenorientierten Datenmodell gespeicherten Datensätze verknüpft ist, aus deren Feldwerten das entsprechende Token erzeugt wurde.

Erste Token in dem Index, welche von einer der Trigger-Definitionen als Trigger umfasst sind, sind jeweils der entsprechen Trigger-Definition zugeordnet, wobei zweite Token in dem Index jeweils ein oder mehreren Klassen der ersten Gruppe von

Klassen zugeordnet sind und wobei die verbleibenden Token in dem Index zum Kennzeichnen der entsprechenden verbleibenden Token als unbekannte Daten einer Auffangklasse zugeordnet sind, wobei die Zuordnung zu der Auffangklasse eine Zuordnung zu einer der Trigger-Definitionen ebenso wie eine Zuordnung zu einer der Klassen der ersten Gruppe von Klassen ausschließt.

Die Programmlogik ist zum Ausführen eines Verfahrens zum maschinellen Lernen konfiguriert ist. Das Verfahren umfasst:

- Empfangen eines zusätzlichen Datensatzes,
- Speichern des zusätzlichen Datensatzes, welcher ein oder mehrere zusätzliche Feldwerte umfasst, durch das Multi-Modell-Datenbankmanagementsystem in dem dokumentenorientierten Datenmodell der Datenbank,
- Erzeugen von ein oder mehreren zusätzlichen Token aus den zusätzlichen Feldwerten,
- falls eines oder mehrere erste zusätzliche Token von einer der Trigger-Definitionen als Trigger umfasst sind, Identifizieren des entsprechenden Tokens als Trigger durch das Lernmodul,
- Verwenden der identifizierten Trigger zum Zuordnen von ein oder mehreren zweiten zusätzlichen Token zu ein oder mehreren Klassen der ersten Gruppe von Klassen durch das Lernmodul, falls die entsprechenden zweiten zusätzlichen Token von dem zusätzliche Datensatz in einer Kombination mit ein oder mehreren der identifizierten Trigger gemäß einer der Trigger-Definitionen umfasst sind, wobei die entsprechenden Trigger gemäß der entsprechenden Trigger-Definition eine entsprechende Klassenzuordnung triggern,
- Zuordnen der verbleibenden zusätzlichen Token, für welche keine Zuordnung zu einer der Trigger-Definitionen und keine Klassenzuordnung aufgrund einer der Trigger-Definitionen erfolgt ist, zu der Auffangklasse,
- Ergänzen des Index durch das Multi-Modell-Datenbankmanagementsystem unter Verwendung der zusätzlichen Token, der Klassenzuordnungen der zusätzlichen Token und eines Zeigers auf den zusätzlichen in dem dokumentenorientierten Datenmodell gespeicherten Datensatz.

Nach Ausführungsformen ist das Computersystem dazu konfiguriert eine oder mehrere der zuvor genannten Ausführungsformen des Verfahrens zum maschinellen Lernen auszuführen.

Ausführungsformen können den Vorteil haben, dass sie ein selbstlernendes System zu schaffen, welches auf allen Daten der Datenbank arbeitet, keinen Zufall in dem Entscheidungs- bzw. Klassifizierungsprozess verwendet und initiale festgelegte Trigger verwendet um empfangene Daten zu klassifizieren. Nach Ausführungsformen erlaubt das System ferner jederzeit, d.h. auch nach einer initialen Anlernphase, externe Eingriffe in die Entscheidungsfindungsprozesse.

Das Computersystem stellt ein System zum maschinellen Lernen auf Basis einer Datenbank bereit, welches unter Verwendung initial festgelegte Trigger-Definitionen beliebige Daten in bekannte Klassen und in Unbekanntes unterteilt. Durch die Zuordnung zu bekannten Klassen werden Meta- bzw. Kontextinformationen in den Datensätzen identifiziert. Der bereitgestellte Index ermöglicht effizient Suchverfahren und/oder maschinelle Lernverfahren auf den von den Datensätzen umfassten Daten laufen zu lassen. Dies kann ohne expliziten Zugriff auf die Datensätze erfolgen, d.h. ausschließlich auf dem Index, oder mit einem expliziten Zugriff auf relevante Datensätze anhand von Zeigern, welche mit in dem Index identifizierten Token verknüpft sind.

Im Weiteren werden Ausführungsformen der Erfindung mit Bezugnahme auf die Zeichnungen näher erläutert. Es zeigen:

Figur 1 ein schematisches Blockdiagramm einer Ausführungsform eines exemplarischen Computersystems,

Figur 2 ein schematisches Blockdiagramme einer exemplarischen Datenverarbeitung durch das Multi-Modell-Datenbankmanagementsystem,

Figur 3 ein schematisches Blockdiagramme einer exemplarischen Datenverarbeitung durch das Multi-Modell-Datenbankmanagementsystem,

Figur 4 ein schematisches Blockdiagramm einer Ausführungsform eines exemplarischen Computersystems,

Figur 5 ein Flussdiagramm einer Ausführungsform eines exemplarischen Verfahrens,

Figur 6 ein Flussdiagramm einer Ausführungsform eines exemplarischen Verfahrens,

Figur 7 ein Flussdiagramm einer Ausführungsform eines exemplarischen Verfahrens,

Figur 8 ein Flussdiagramm einer Ausführungsform eines exemplarischen Verfahrens und

Figur 9 ein Flussdiagramm einer Ausführungsform eines exemplarischen Verfahrens.

Elemente der nachfolgenden Ausführungsformen, die einander entsprechen, werden mit denselben Bezugszeichen gekennzeichnet.

Figur 1 zeigt ein Blockdiagramm einer Ausführungsform eines exemplarischen Computersystems 100 zum maschinellen Lernen. Das Computersystems 100 umfasst zumindest eine Datenbank 104 und ein Multi-Modell-Datenbankmanagementsystem (MM-DBMS) 118. Das MM-DBMS 118 verwaltet die, ggf. strukturierte, Speicherung der Daten in der zumindest einen Datenbank 104 und kontrolliert alle lesenden und schreibenden Zugriffe auf die Datenbank 104. Die MM-DBMS 118 unterstützt zumindest zwei Datenmodelle 106, 110, in welchen die Daten in der Datenbank 104 gespeichert werden. Dabei legt das Datenbankmodell fest, in welcher Form die entsprechenden Daten organisiert, gespeichert und bearbeitet werden. Bei einem oder beiden Datenmodellen 106, 110 handelt es sich um NoSQL Datenmodelle. In dieser Hinsicht handelt es sich bei dem MM-DBMS 118 um ein NoSQL DBMS. Das erste Datenmodell 106 ist ein dokumentenbasiertes Datenmodell, in welchem eine Mehrzahl von Datensätzen DS1, DS2, DS3 gespeichert sind. Jeder Datensatz DS1, DS2, DS3 wird in einem Dokument bzw. einem Datencontainer gespeichert. Den Datensätzen DS1, DS2, DS3 selbst wird beim Speichern von dem dokumentenbasiertes Datenmodell 106 keine spezifische Struktur vorgegeben.

Mithin können die Datensätzen DS1, DS2, DS3 mit der internen Struktur gespeichert werden, mit welcher die Datensätzen DS1, DS2, DS3 von der Datenbank 104 empfangen werden. Insofern handelt es sich bei den in dem dokumentenbasiertes Datenmodell 106 gespeicherten Datensätzen DS1, DS2, DS3 um Rohdaten. Die Datensätze DS1, DS2, DS3 können beispielsweise Textdaten, Bilddaten, Audiodaten und/oder Videodaten umfassen. Die Datensätze DS1, DS2, DS3 umfassen jeweils zumindest ein Feld F1, ..., F8, mit Feldwerten. Die Datensätze DS1, DS2, DS3 können bereits eine innere Struktur mit einer Mehrzahl von Feldern F1, ..., F8 aufweisen, wenn sie gespeichert werden. Dann umfassen die entsprechenden Datensätze DS1, DS2, DS3 jeweils eine Mehrzahl von Feld F1, ..., F8. Weisen die Datensätze DS1, DS2, DS3 selbst bei ihrem Empfang keine Felder auf, so umfassen sie in gespeicherter Form jeweils beispielsweise genau ein Feld, welches den gesamten Datenumfang des entsprechenden Datensatzes DS1, DS2, DS3 umfasst. Die Felder F1, ..., F8 umfassen jeweils ein oder mehrere Feldwerte. Jeder der Feldwerte eines Datensatzes DS1, DS2, DS3 ist in einem entsprechenden Feld, einer Art Datencontainer, gespeichert. Jedes Feld F1, ..., F8 kann einem Feldtyp zugewiesen sein. Dabei können die Felder F1, ..., F8 unterschiedlichen oder alle demselben Feldtyp zugewiesen sein. Die Zusammensetzung der Feldwerte der einzelnen Datensätze DS1, DS2, DS3 kann sich dabei im Hinblick auf deren Feldtypen unterscheiden. Es ist auch möglich das einzelne Datensätze gar keine Felder eines bestimmten Feldtyps beinhalten. In anderen Ausführungsformen (hier nicht gezeigt) können auch mandatorische Feldtypen definiert sein, d.h. dass jedes Dokument ein Feld für jeden mandatorischen Feldtyp umfasst und optional ein oder mehrere weitere Felder für optionale Feldtypen umfasst. Die Daten der Datensätze werden dann in Feldern des für sie vorgesehenen Feldtyps gespeichert, d.h. z.B. Textdaten in einem oder mehreren Textfeldern, Bilddaten in einem oder mehreren Bildfeldern, Audiodaten in einem oder mehreren Audiofeldern und/oder Videodaten in einem oder mehreren Videofeldern.

Ferner umfasst das Computersystem 100 ein Lernmodul zum Verarbeiten der in der Datenbank 104 gespeicherten Daten. Das Lernmodul 120 umfasst beispielsweise zumindest einen Tokenisierer 120 zum Erzeugen von Token 109, Trigger-

Definitionen 123, welche Trigger für eine Klassifizierung von Token 109 definieren, und/oder einen Klassifizierer 124, welcher die Token 109 unter Verwendung der Trigger-Definitionen 123 klassifiziert. Nach Ausführungsformen umfasst das Lernmodul 120 ferner ein statistisches Modell 125. Das statistisches Modell 125 kann dazu konfiguriert sein, Trigger-Kombinationen zu erfassen und kombinierte Trigger-Definitionen zu erstellen, zusätzliche Trigger-Definitionen zu erstellen und/oder korrigierte Trigger-Definition zu erstellen. Nach alternativen Ausführungsformen kann auch das MM-DBMS 118 den Tokenisierer 122 umfassen und/oder auf einen von dem Lernmodul 120 bereitgestellten Tokenisierer 122 zugreifen. Nach alternativen Ausführungsformen kann können die Trigger-Definitionen 123 auch in der Datenbank 104 gespeichert sein.

Das MM-DBMS 118 und/oder das Lernmodul 120 verfügen dabei über eine eingebaute Programlogik, die zur Generierung eines Index 112 konfiguriert ist. Der entsprechende Index 112 wird dabei in einem weiteren Datenmodell 110 bereitgestellt, in welchem die vollständigen Daten der Datensätze DS1, DS2, DS3 in umstrukturierter, redundanzfreier Form gespeichert sind. Zum Erzeugen des Index 112 wird auf den Tokenisierer 122 zugegriffen, welcher dazu konfiguriert ist, die Feldwerte der Feldern F1, ..., F8 der in dem dokumentenbasierten Datenmodell 106 gespeicherten Datensätze 106 zu tokenisieren. Dabei können die resultierenden Token 109 auch mit einem Feldwert eines Feldes bzw. eines Datensatzes identisch sein, falls keine weitere Zerlegung in Token 109 möglich oder sinnvoll ist. Die Tokenisierung kann nach Ausführungsformen auch stufenweise erfolgen, so dass eine immer feinere Zerlegung erfolgt. Mithin umfasst der resultierende Index 112 in diesem Fall Token 109, welche aus anderen Token 109 zusammengesetzt sind.

Vorzugsweise werden sämtliche oder zumindest die meisten Feldwerte sämtlicher Datensätze DS1, DS2, DS3 der Datenbank 104 tokenisiert, sodass eine umfangreiche Menge an Token 109 entsteht. In Abhängigkeit von der Art der Daten in den einzelnen Feldwerten können die Token 109 eine Mischung aus Zahlen, Buchstabenwörtern, Bildern oder Bildsegmenten, Audiodateien oder Audioelementen oder sonstigen Datenstrukturen, insbesondere Sensordaten von ein oder mehreren

Sensoren, umfassen. Jedes der erzeugten Token 109 wird in dem Index 112 mit einem Zeiger verknüpft gespeichert, wobei der Zeiger auf den Datensatz bzw. das Feld verweist, aus dem das Token 109 entstammt.

Dabei werde Token 109 in dem Index 112, welche von einer der Trigger-Definitionen 123 als Trigger umfasst sind, jeweils der entsprechen Trigger-Definition 123 zugeordnet. Ferner werden Token 109 in dem Index 112, welche von einem der Datensätze DS1, DS2, DS3 in einer Kombination mit ein oder mehreren der identifizierten Trigger gemäß einer der Trigger-Definitionen 123 umfasst sind, jeweils ein oder mehreren Klassen zugeordnet. Die entsprechenden Klassenzuordnungen stellen dabei Meta- bzw. Kontextinformationen für die entsprechenden Token 109 bereit. Schließlich werden sind die verbliebenen Token 109 in dem Index 112, welche sich unter Verwendung der Trigger-Definitionen 123 weder als Trigger identifizieren, noch einer Klasse zuordnen als, zum Kennzeichnen als unbekannte Daten einer Auffangklasse zugeordne. Dabei schließt eine Zuordnung zu der Auffangklasse eine Zuordnung zu einer der Trigger-Definitionen 123 ebenso wie eine Zuordnung zu einer der Klassen gemäß den Trigger-Definitionen 123 aus. Die zuvor beschriebenen Zuordnungen erfolgen beispielsweise unter Verwendung des Klassifizierers 124 des Lernmoduls 120.

Im Zuge der Erzeugung des Index 112 wird aus der Menge an Token 109 eine nichtredundante, unique Token-Menge gebildet, in welcher jedes der Token 109 nur ein einziges Mal vorkommt. Auch wenn ein Token 109 mit einem bestimmten Wert und einer bestimmten Klassenzuordnung mehrfach in der Datenbank 104 bzw. in dem Datenmodell 106 vorkommt, wird es beispielsweise nur ein einziges Mal mit dieser Klassenzuordnung in der nichtredundante Token-Menge und in dem Index 112 gespeichert. Vorzugsweise erfolgt die Speicherung aller Token 109 der nichtredundanten Tokenmenge in dem Index 112 so, dass die Token 109 nach einem Sortierkriterium sortiert werden und in sortierter Form in der Indexstruktur gespeichert werden. Die Sortierung kann beispielsweise anhand des Alphabets für alphanumerische Daten oder sonstiger, an die Daten angepasste Sortierkriterien erfolgen. Da die Token 109 in dem Index 112 vorzugsweise in sortierter Form gespeichert sind,

und weiterhin vorzugsweise in einer Baumstruktur gespeichert sind, ist es sehr schnell möglich, ein bestimmtes Token 109 innerhalb des Index 112 zu identifizieren und dann die Verweise dieses identifizierten Tokens 109 auf ein oder mehrere Datensätze DS1, DS2, DS3 zu verwenden, um sehr schnell diejenigen Datensätze zu identifizieren, die ein bestimmtes, gesuchtes Token 109 enthalten. Es ist also nicht erforderlich, alle Datensätze DS1, DS2, DS3 der Datenbank 104 sequenziell zu durchsuchen.

Figur 2 zeigt ein schematisches Blockdiagramme einer exemplarischen Datenverarbeitung durch das Multi-Modell-Datenbankmanagementsystem und das Lernmodul. Es wird eine vorbestimmte Trigger-Definition 123 der Form „[Vorname] [Nachname] [Trigger1 = wohnt in der] [Straße] [Trigger2 = in] [Stadt]“ bereitgestellt. Diese Trigger-Definition 123 definiert zwei Trigger, d.h. einen ersten Trigger „wohnt in der“ und einen zweiten Trigger „in“. Zudem definiert die Trigger-Definition, dass es sich bei einem dem ersten Trigger unmittelbar vorangehenden Token um einen Nachnamen handelt, während es sich bei einem dem Nachnamen unmittelbar vorangehenden Token um einen Vornamen handelt. Ferner definiert Trigger-Definition, dass es sich bei einem zwischen den beiden Triggern angeordneten Token um eine Straße handelt und dass es sich bei einem dem zweiten Trigger unmittelbar nachfolgenden Token um eine Stadt handelt.

In einem dokumentenbasierten Datenmodell 106 einer Datenbank sind zwei Dokumente 108 gespeichert. Jedes Dokument 108 umfasst jeweils einen Datensatz DS1, DS2. Beispielsweise handelt es sich bei den Datensätzen DS1, DS2 jeweils um eine Textdatei. Der erste Datensatz DS1 umfasst beispielsweise den Satz: „Mustervorname_1 Musternachname_1 wohnt in der Musterstr._1 in Musterstadt_1“. Dieser Satz wird mittels Tokenisierers in Token 109 zerlegt: „Mustervorname_1“, „Musternachname_1“, „wohnt in der“, „Musterstr._1“, „in“, „Musterstadt_1“.

Die beiden Token „wohnt in der“ und „in“ werden als Trigger gemäß der Trigger-Definition 123 identifiziert. Unter Verwendung der identifizierten Trigger sowie der Trigger-Definition 123 werden die verbleibenden Token 109 jeweils den von der Trigger-

Definition definierten Klassen 111 zugeordnet. So wird das Token „Mustervorname_1“ der Klasse „Vorname“, das Token „Musternachname_1“ der Klasse „Nachname“, das Token „Musterstr._1“ der Klasse „Straße“ und das Token „Musterstadt_1“ der Klasse „Stadt“ zugeordnet. Die als Trigger identifizierten Token werden ebenso wie die anhand dieser Trigger klassifizierten Token in einem Index in einem zweiten Datenmodell 110 gespeichert. Dabei werden die Trigger in Form einer Trigger-Zuordnung 117 jeweils der Trigger-Definition 123 zugeordnet. Die verbleibenden Token 109 werden jeweils in Form einer Klassen-Zuordnung 113 einer der von der Trigger-Definition 123 definierten Klassen zugeordnet gespeichert. Zudem werden alle Trigger und klassifizierten Token in dem zweiten Datenmodell 110 mit einem Zeiger 115 auf ihren Speicherort in dem ersten Datenmodell, d.h. DS1, verknüpft.

Empfängt die Datenbank einen zweiten Datensatz DS2 der Form: „Mustervorname_2 Musternachname_2 wohnt in der Musterstr._2 in Musterstadt_2“, so wird dieser Satz mittels Tokenisierers in Token zerlegt: „Mustervorname_2“, „Musternachname_2“, „wohnt in der“, „Musterstr._2“, „in“, „Musterstadt_2“ und in redundanzfreier Form ebenfalls in dem zweiten Datenmodell 110 gespeichert.

Die beiden Token „wohnt in der“ und „in“ werden als Trigger gemäß der Trigger-Definition 123 identifiziert. Da diese beiden Trigger der Trigger-Definition 123 von dem Index bereits umfasst sind, werden diese nicht nochmals in dem zweiten Datenmodell 110 gespeichert. Es wird lediglich ein Zeiger auf den zweiten Datensatz DS2 ergänzt. Unter Verwendung der identifizierten Trigger sowie der Trigger-Definition 123 werden die verbleibenden Token 109 des Datensatzes DS2 jeweils den von der Trigger-Definition definierten Klassen 111 zugeordnet. So wird das Token „Mustervorname_2“ der Klasse „Vorname“, das Token „Musternachname_2“ der Klasse „Nachname“, das Token „Musterstr._2“ der Klasse „Straße“ und das Token „Musterstadt_2“ der Klasse „Stadt“ zugeordnet. Die klassifizierten Token 109 des Datensatzes DS2 werden jeweils in Form einer Klassen-Zuordnung 113 einer der von der Trigger-Definition 123 definierten Klassen zugeordnet gespeichert und mit einem Zeiger 115 auf ihren Speicherort in dem ersten Datenmodell, d.h. DS2, verknüpft.

Mithin werden alle Token des zweiten Datensatzes DS2 ebenfalls in redundanzfreier Form jeweils mit ihren Klassen-Zuordnungen in dem zweiten Datenmodell 110 verknüpft mit einem Zeiger auf ihren Speicherort in dem ersten Datenmodell gespeichert.

Figur 3 zeigt ein schematisches Blockdiagramme einer exemplarischen Datenverarbeitung durch das Multi-Modell-Datenbankmanagementsystem und das Lernmodul. Es wird eine vorbestimmte Trigger-Definition 123 der Form „[Trigger1 = +, Radius = NP] [Trigger1 = x, Radius = NP] [Klasse]“ bereitgestellt. Diese Trigger-Definition 123 dient dazu aus einer Bilddatei erzeugte Token zu klassifizieren, wobei die Bilddatei in Token in Form von Pixelgruppen zerlegt wird. Die Trigger-Definition 123 definiert zwei Trigger, d.h. einen ersten Trigger in Form einer Pixelgruppe mit dem Inhalt „+“ und einen zweiten Trigger in Form einer Pixelgruppe mit dem Inhalt „x“. Zudem definiert die Trigger-Definition, dass es sich bei einer Pixelgruppe, welche innerhalb eines ersten Radius von N Pixeln um den ersten Trigger und zugleich innerhalb eines zweiten Radius von N Pixeln um den zweiten Trigger angeordnet ist, um eine ein Token der der Klasse „Klasse“ handelt.

In einem dokumentenbasierten Datenmodell 106 einer Datenbank ist ein Dokumente 108 gespeichert. Dieses Dokument 108 umfasst einen Datensatz DS. Beispielsweise handelt es sich bei dem Datensatz DS um eine zweidimensionale Bilddatei. Diese Bilddatei wird mittels Tokenisierers in Token zerlegt, wobei es sich bei den Token jeweils um Pixelgruppen 150 handelt. Beispielsweise wird der in Z mal Z gleichgroße Pixelgruppen zerlegt. Die Token umfassen beispielsweise ein erstes Token in Form einer Pixelgruppe mit dem Inhalt „x“, ein zweites Token in Form einer Pixelgruppe mit dem Inhalt „+“, ein drittes Token in Form einer Pixelgruppe mit dem Inhalt „#“ und ein viertes Token in Form einer Pixelgruppe mit dem Inhalt „-“.

Die beiden Token „+“ und x“ werden als Trigger 121 gemäß der Trigger-Definition 123 identifiziert. Unter Verwendung der identifizierten Trigger sowie der Trigger-Definition 123 wird das dritte Token „#“ der von der Trigger-Definition definierten

Klasse 111 zugeordnet, da es in der zweidimensionalen Bilddatei innerhalb eines ersten Radius 152 von N Pixeln um den ersten Trigger „+“ und zugleich innerhalb eines zweiten Radius 154 von N Pixeln um den zweiten Trigger „x“ angeordnet ist. Da das vierte Token „-“ nicht unter die Trigger-Definition 123 fällt, wird es als unbekanntes Datum der Auffangklasse zugeordnet.

Die als Trigger 121 identifizierten Token „+“ und „x“ werden ebenso wie das anhand dieser Trigger klassifizierte Token „#“ und das der Auffangklasse zugeordnete Token „-“ in einem Index in einem zweiten Datenmodell 110 gespeichert. Dabei werden die Trigger „+“ und „x“ in Form einer Trigger-Zuordnung 117 jeweils der Trigger-Definition 123 zugeordnet. Das Token „#“ wird in Form einer Klassen-Zuordnung 113 der von der Trigger-Definition 123 definierten Klassen zugeordnet gespeichert. Das Token „-“ wird in Form einer Zuordnung 119 der Auffangklassen zugeordnet gespeichert. Zudem werden alle Trigger und klassifizierten Token in dem zweiten Datenmodell 110 mit einem Zeiger 115 auf ihren Speicherort in dem ersten Datenmodell, d.h. DS, verknüpft.

Figur 4 zeigt ein schematisches Blockdiagramm einer Ausführungsform eines exemplarischer Computersystems 110. Das Computersystem 100 umfasst einen Prozessor 114, welcher Programminstruktionen 116, wodurch das Computersystem zum Ausführen des zuvor beschriebenen Verfahrens zum maschinellen Lernen veranlasst wird. Im Zuge des Ausführens des Verfahrens führt der Prozessor 114 zudem eine Multi-Modell-Datenbankmanagementsystem 118 und ein Lernmodul 120 zum maschinellen Lernen mit einem Tokenisierer 122 und einem Klassifizierer 124 aus. Zudem umfasst das Lernmodul 120 Trigger-Definitionen 123. Nach Ausführungsformen umfasst das Lernmodul 120 außerdem ein statistisches Modell 125. Ferner umfasst das Computersystems 110 in einem Speicher 102 eine Datenbank 104, welche von dem Multi-Modell-Datenbankmanagementsystem 118 verwaltet wird. Die Datenbank umfasst ein erstes Datenmodell 106, z.B. ein dokumentenorientiertes Datenmodell, in welchem Datensätze 108 gespeichert werden. Ferner umfasst die Datenbank ein zweites Datenmodell 110 mit einem Index 112 aller in den Datensätzen 108 gespeicherten Daten.

Figur 5 zeigt ein Flussdiagramm einer Ausführungsform eines exemplarischen Verfahrens zum maschinellen Lernen. In Block 200 wird ein vortrainiertes Lernmoduls zum maschinellen Lernen bereitgestellt, welches eine Mehrzahl von vorbestimmten Trigger-Definitionen umfasst. Diese vorbestimmten Trigger-Definitionen definieren Trigger für ein Zuordnen von Token zu Klassen einer Gruppe von Klassen. In Block 202 wird eine Datenbank bereitgestellt. Die Datenbank wird von einem Multi-Modell-Datenbankmanagementsystem verwaltet und umfasst eine Mehrzahl von Datensätze, welche in einem dokumentenorientierten Datenmodell gespeichert sind. Diese gespeicherten Datensätze umfassen jeweils ein oder mehreren Felder mit Feldwerten. Zudem umfasst die bereitgestellte Datenbank einen durchsuchbaren Index aller von den gespeicherten Datensätzen umfassten Daten. Dieser Index wird redundanzfrei in einem weiteren von dem Multi-Modell-Datenbankmanagementsystem verwalteten Datenmodell gespeichert. Der Index umfasst eine Mehrzahl von aus den Feldwerten der gespeicherten Datensätze erzeugten Token die in dem Index jeweils mit einem oder mehreren Zeigern auf ein oder mehrere der in dem dokumentenorientierten Datenmodell gespeicherten Datensätze und/oder Felder verknüpft sind, aus deren Feldwerten das entsprechende Token erzeugt wurde.

Erste Token in dem Index, welche von einer der Trigger-Definitionen als Trigger umfasst sind, sind jeweils der entsprechen Trigger-Definition zugeordnet. Zweite Token in dem Index sind jeweils ein oder mehreren Klassen der Gruppe von Klassen zugeordnet. Die verbleibenden Token in dem Index sind schließlich zum Kennzeichnen der entsprechenden verbleibenden Token als unbekannte Daten einer Auffangklasse zugeordnet. Dabei schließt die Zuordnung zu der Auffangklasse eine Zuordnung zu einer der Trigger-Definitionen ebenso wie eine Zuordnung zu einer der Klassen der ersten Gruppe von Klassen aus.

In Block 204 wird ein zusätzlicher Datensatz empfangen und in Block 206 durch das Multi-Modell-Datenbankmanagementsystem in dem dokumentenorientierten Datenmodell der Datenbank gespeichert. Dabei erfolgt die Speicherung in einem Dokument bzw. Datencontainer. In Block 208 werden ein oder mehrere zusätzliche

Token aus zusätzlichen Feldwerten erzeugt, welche der zusätzliche Datensatz umfasst. In Block 210 werden ein oder mehrere erste zusätzliche Token als Trigger identifiziert, falls diese von einer der Trigger-Definitionen als Trigger umfasst sind. In Block 212 werden die verbleibenden zusätzlichen Token klassifiziert. Die in Block 210 identifizierten Trigger werden zum Zuordnen von ein oder mehreren zweiten zusätzlichen Token zu ein oder mehreren Klassen der Gruppe von Klassen verwendet, falls die entsprechenden zweiten zusätzlichen Token von dem zusätzliche Datensatz in einer Kombination mit ein oder mehreren der identifizierten Trigger gemäß einer der Trigger-Definitionen umfasst sind und die entsprechenden Trigger gemäß der entsprechenden Trigger-Definition eine entsprechende Klassenzuordnung triggern. Die verbleibenden zusätzlichen Token, für welche keine Zuordnung zu einer der Trigger-Definitionen und keine Klassenzuordnung aufgrund einer der Trigger-Definitionen erfolgt ist, werden im Zuge des Klassifizierens in Block 212 der Auffangklasse zugeordnet. Schließlich wird in Block 214 der Index durch das Multi-Modell-Datenbankmanagementsystem unter Verwendung der unter Verwendung der zusätzlichen Token aus Block 208, der Klassenzuordnungen der zusätzlichen Token aus Block 212 und eines Zeigers auf den zusätzlichen in dem dokumentenorientierten Datenmodell gespeicherten Datensatz ergänzt. Falls Zeiger einzelne Felder des zusätzlichen Datensatzes anzeigen, wird bei einer Mehrzahl von Feldern eine Mehrzahl von Zeigern verwendet.

Dabei kann das Ergänzen in Block 214 ein Abgleichen der zusätzlichen Token mit dem Index umfassen. Falls eines der zusätzlichen Token nicht von dem Index umfasst ist, wird das entsprechende zusätzliche Token unter seinen Klassenzuordnungen in dem Index ergänzt und mit dem Zeiger auf den zusätzlichen in dem dokumentenorientierten Datenmodell gespeicherten Datensatz verknüpft. Falls eine der Klassenzuordnungen eines von dem Index umfassten zusätzlichen Tokens von dem Index nicht umfasst ist, wird die entsprechende Klassenzuordnung mit dem entsprechenden zusätzlichen Token in dem Index ergänzt und das entsprechende zusätzliche Token in dem Index mit dem Zeiger auf den zusätzlichen in dem dokumentenorientierten Datenmodell gespeicherten Datensatz verknüpft. Falls eines der zusätzlichen Token mit allen seinen Klassenzuordnungen von dem Index umfasst ist, wird

das entsprechende zusätzliche Token in dem Index mit dem Zeiger auf den zusätzlichen in dem dokumentenorientierten Datenmodell gespeicherten Datensatz verknüpft.

Zudem kann das Ergänzen in Block 214 ein Kennzeichnen von Kombinationen aus zweiten zusätzlichen Token mit ein oder mehreren der identifizierten Trigger, welche eine Klassenzuordnung gemäß einer der Trigger-Definitionen getriggert haben, in dem Index als klassifizierte Kombinationen umfassen. Klassenzuordnungen werden nur für Kombinationen aus zweiten zusätzlichen Token und ein oder mehreren identifizierten Triggern ausgeführt, welche nicht als klassifizierte Kombinationen gekennzeichnet sind. Somit kann durch das Kennzeichnen vermieden werden, dass für bereits bekannte und klassifizierte Kombinationen bei einem wiederholten Auftreten in verschiedenen Datensätzen jeweils das Klassifizieren wiederholt wird. Vielmehr erfolgt vor einem Klassifizieren ein Abgleich von Token-Kombinationen mit dem Index. Umfasst der Index die entsprechende Token-Kombination bereits und ist diese als klassifiziert gekennzeichnet, so erfolgt keine erneute Klassifikation für diese Token-Kombination. Es werden lediglich die entsprechende Token-Kombination und/oder die von der entsprechenden Token-Kombination umfassten Teilkombinationen und Einzeltoken in dem Index mit dem Zeiger auf den zusätzlichen in dem dokumentenorientierten Datenmodell gespeicherten Datensatz verknüpft.

Figur 6 zeigt ein Flussdiagramm einer Ausführungsform eines exemplarischen Verfahrens zum Erzeugen kombinierter Trigger-Definitionen. In Block 300 werden ein oder mehr Trigger-Kombinationen durch das Lernmodul identifiziert, welche jeweils von zumindest einem der Datensätzen umfasst sind und ein Kombinationskriterium erfüllen. In Block 302 werden für jede der in Block 300 identifizierten Trigger-Kombinationen die Trigger-Definitionen der Trigger der entsprechenden Trigger-Kombinationen zu ein oder mehreren zusätzlichen kombinierten Trigger-Definitionen kombiniert. In Block 304 wird die Mehrzahl von vorbestimmten Trigger-Definitionen des Lernmoduls durch die ein oder mehreren zusätzlichen kombinierten Trigger-Definitionen ergänzt.

Figur 7 zeigt ein Flussdiagramm einer Ausführungsform eines exemplarischen Verfahrens zum Ergänzen zusätzlicher Trigger-Definitionen. In Block 400 wird das vortrainierte Lernmoduls um ein oder mehrere zusätzliche Trigger-Definitionen ergänzt. Die zusätzlichen Trigger-Definitionen definieren zusätzliche Trigger für ein Ersetzen von Zuordnungen von Token in dem Index zu der Auffangklasse durch Zuordnungen zu ein oder mehreren Klassen einer weiteren Gruppe von Klassen im Zuge eines Reklassifizierens. Die zusätzlichen Trigger-Definitionen können beispielsweise von dem Lernmodul empfangen werden. Beispielsweise werden die entsprechenden zusätzlichen Trigger-Definitionen von einem Administrator bereitgestellt. Nach alternativen Ausführungsformen werden die zu ergänzenden zusätzlichen Trigger-Definitionen von dem Lernmodul erstellt. Hierzu umfasst das Lernmodul ein statistisches Modell, welches zu einer statistischen Analyse der von der Auffangklassen umfassten Token und deren Vorkommen in den Datensätze verwendet wird. Das Ergebnis der statistischen Analyse wird zum Erstellen der zu ergänzenden zusätzlichen Trigger-Definitionen verwendet.

In Block 402 werden ein oder mehreren der Auffangklasse zugeordneten Token in dem Index reklassifiziert, welche die zusätzlichen Trigger-Definitionen als zusätzliche Trigger definierten. Das Reklassifizieren durch das Lernmodul umfasst ein Ersetzen der Zuordnung zu der Auffangklasse durch eine Zuordnung zu der entsprechen zusätzlichen Trigger-Definition, welche das entsprechende Token als zusätzlichen Trigger umfasst. In Block 404 werden die zusätzliche Trigger zum Reklassifizieren von ein oder mehreren der Auffangklasse zugeordneten Token in dem Index zu ein oder mehreren Klassen der weiteren Gruppe von Klassen durch das Lernmodul verwendet, falls die entsprechenden der Auffangklasse zugeordneten Token von einem der Datensätze in einer Kombination mit ein oder mehreren der zusätzlichen Triggern umfasst sind und die entsprechenden zusätzlichen Trigger gemäß der entsprechenden zusätzlichen Trigger-Definition eine entsprechende Zuordnung zu den ein oder mehreren Klassen der weiteren Gruppe von Klassen triggern.

Beispielsweise kann das Verfahrens zum Ergänzen zusätzlicher Trigger-Definitionen einem rekursiven Schema folgend wiederholt ausgeführt werden. Die zu

ergänzenden Trigger-Definitionen jeder Rekursionsstufe umfassen jeweils Ergänzungen einer Trigger-Definition einer vorangehenden Rekursionsstufe, sodass die rekursiven Ergänzungen Baumstrukturen bilden, welche jeweils eine der vorbestimmten Trigger-Definition als Wurzelknoten umfassen.

Figur 8 zeigt ein Flussdiagramm einer Ausführungsform eines exemplarischen Verfahrens zum Korrigieren von Trigger-Definitionen in Block. In Block 500 wird eine korrigierte Trigger-Definition zum Ersetzen einer der gespeicherten Trigger-Definitionen des Lernmoduls empfangen. Diese korrigierte Trigger-Definition wird beispielsweise von einem Administrator bereitgestellt. Nach alternativen Ausführungsformen wird die korrigierte Trigger-Definition von dem Lernmodul unter Verwendung eines statistischen Modells erstellt. In Block 502 wird die entsprechende gespeicherte Trigger-Definition durch die korrigierte Trigger-Definition ersetzt. In Block 504 erfolgt ein Reklassifizieren der unter Verwendung der entsprechenden gespeicherten Trigger-Definition klassifizierten Token, wobei das Reklassifizieren unter Verwendung der korrigierten Trigger-Definition erfolgt.

Figur 9 zeigt schließlich ein Flussdiagramm einer Ausführungsform eines exemplarischen Verfahrens zum Ausführen einer Suche auf der Datenbank. In Block 600 wird eine Suchanfrage empfangen, die einen Suchwert beinhaltet. Block 602 wird der Index nach dem Suchwert durchsucht und in Block 604 wird ein Token innerhalb des Index identifiziert, welcher identisch mit dem Suchwert ist. Nach Ausführungsformen kann der Suchwert neben einem Tokenwert auch eine Klassenzuordnung spezifizieren. In diesem Fall erfordert das Identifizieren des Tokens innerhalb des Index ferner, dass das identifizierte Token diein der Suchanfrage spezifizierte Klassenzuordnung aufweist. Nach Ausführungsformen sind Token, welche der Auffangklasse zugeordnet sind, von der Suche ausgeschlossen. In Block 606 werden Zeiger analysiert, mit denen das identifizierte Token verknüpft ist. Dadurch werden ein oder mehrere der Datensätze bestimmt, welche ein oder mehrere Feldwerte beinhalten, aus welchen das indizierte Token erzeugt wurde. In Block 608 werden die bestimmten Datensätze oder von ein oder mehreren Referenzen auf die bestimmten Datensätze als Antwort auf die Suchanfrage zurückgegeben.

Bezugszeichenliste

- -

100	Computersystem
102	Speicher
104	Datenbank
106	erstes Datenmodell
108	Datensätze
109	Token
110	zweites Datenmodell
111	Klasse
112	Index
113	Klassen-Zuordnung
114	Prozessor
115	Zeiger
116	Programminstruktionen
117	Trigger-Zuordnung
118	MM-DBMS
119	Zuordnung zu Auffangklasse
120	Lernmodul
121	Trigger
122	Tokenisierer
123	Trigger-Definitionen
124	Klassifizierer
125	statistische Modell
150	Pixelgruppe
152	Radius
154	Radius
F1, ..., F8	Felder
DS1, ..., DS3	Datensätze

Patentansprüche

- -

1. Computerimplementiertes Verfahren zum maschinellen Lernen, wobei das Verfahren umfasst:

- Bereitstellen eines vortrainierten Lernmoduls (120) zum maschinellen Lernen, wobei das vortrainierte Lernmodul eine Mehrzahl von vorbestimmten Trigger-Definitionen (123) umfasst, welche Trigger (121) für ein Zuordnen von Token (109) zu Klassen (111) einer ersten Gruppe von Klassen definieren,
- Bereitstellen einer Datenbank (104), welche von einem Multi-Modell-Datenbankmanagementsystem (118) verwaltet wird, wobei die Datenbank eine Mehrzahl von Datensätze (108; DS1, DS2, DS3) umfasst, welche in einem dokumentenorientierten Datenmodell (106) gespeichert sind, wobei die gespeicherten Datensätze jeweils ein oder mehreren Feldwerte umfassen, wobei die einzelnen Feldwerte der gespeicherten Datensätze jeweils in einem Feld (F1, ..., F8) gespeichert sind,

  wobei die Datenbank ferner einen durchsuchbaren Index (112) umfasst, welcher in einem weiteren Datenmodell (110) gespeichert ist, wobei der Index eine Mehrzahl von aus den Feldwerten der gespeicherten Datensätze erzeugten Token umfasst, wobei Token in dem Index jeweils mit einem oder mehreren Zeigern (115) auf ein oder mehrere der in dem dokumentenorientierten Datenmodell gespeicherten Datensätze verknüpft ist, aus deren Feldwerten das entsprechende Token erzeugt wurde,

  wobei erste Token in dem Index, welche von einer der Trigger-Definitionen als Trigger umfasst sind, jeweils der entsprechen Trigger-Definition zugeordnet sind, wobei zweite Token in dem Index jeweils ein oder mehreren Klassen der ersten Gruppe von Klassen zugeordnet sind und wobei die verbleibenden Token in dem Index zum Kennzeichnen der entsprechenden verbleibenden Token als unbekannte Daten einer Auffangklasse zugeordnet sind, wobei die Zuordnung zu der Auffangklasse eine Zuordnung zu einer der Trigger-Definitionen ebenso wie eine Zuordnung zu einer der Klassen der ersten Gruppe von Klassen ausschließt,

- Empfangen eines zusätzlichen Datensatzes,
- Speichern des zusätzlichen Datensatzes, welcher ein oder mehrere zusätzliche Feldwerte umfasst, durch das Multi-Modell-Datenbankmanagementsystem in dem dokumentenorientierten Datenmodell der Datenbank,
- Erzeugen von ein oder mehreren zusätzlichen Token aus den zusätzlichen Feldwerten,
- falls eines oder mehrere erste zusätzliche Token von einer der Trigger-Definitionen als Trigger umfasst sind, Identifizieren des entsprechenden Tokens als Trigger durch das Lernmodul,
- Verwenden der identifizierten Trigger zum Zuordnen von ein oder mehreren zweiten zusätzlichen Token zu ein oder mehreren Klassen der ersten Gruppe von Klassen durch das Lernmodul, falls die entsprechenden zweiten zusätzlichen Token von dem zusätzliche Datensatz in einer Kombination mit ein oder mehreren der identifizierten Trigger gemäß einer der Trigger-Definitionen umfasst sind, wobei die entsprechenden Trigger gemäß der entsprechenden Trigger-Definition eine entsprechende Klassenzuordnung triggern,
- Zuordnen der verbleibenden zusätzlichen Token, für welche keine Zuordnung zu einer der Trigger-Definitionen und keine Klassenzuordnung aufgrund einer der Trigger-Definitionen erfolgt ist, zu der Auffangklasse,
- Ergänzen des Index durch das Multi-Modell-Datenbankmanagementsystem unter Verwendung der zusätzlichen Token, der Klassenzuordnungen der zusätzlichen Token und eines Zeigers auf den zusätzlichen in dem dokumentenorientierten Datenmodell gespeicherten Datensatz.

2. Computerimplementiertes Verfahren nach Anspruch 1, wobei das Ergänzen des Index umfasst:

- Abgleichen der zusätzlichen Token mit dem Index,
- falls eines der zusätzlichen Token nicht von dem Index umfasst ist, Ergänzen des entsprechenden zusätzlichen Tokens unter seinen Klassenzuordnungen in dem Index und Verknüpfen des entsprechenden zusätzlichen Tokens in dem Index mit dem Zeiger auf den zusätzlichen in dem dokumentenorientierten Datenmodell gespeicherten Datensatz,

- falls eine der Klassenzuordnungen eines von dem Index umfassten zusätzlichen Tokens von dem Index nicht umfasst ist, Ergänzen der entsprechenden Klassenzuordnung mit dem entsprechenden zusätzlichen Token in dem Index und Verknüpfen des entsprechenden zusätzlichen Tokens in dem Index mit dem Zeiger auf den zusätzlichen in dem dokumentenorientierten Datenmodell gespeicherten Datensatz,
- falls eines der zusätzlichen Token mit allen seinen Klassenzuordnungen von dem Index umfasst ist, Verknüpfen des entsprechenden zusätzlichen Tokens in dem Index mit dem Zeiger auf den zusätzlichen in dem dokumentenorientierten Datenmodell gespeicherten Datensatz.

3. Computerimplementiertes Verfahren nach einem der vorangehenden Ansprüche, wobei die Kombinationen aus zweiten zusätzlichen Token mit ein oder mehreren der identifizierten Trigger, welche eine Klassenzuordnung gemäß einer der Trigger-Definitionen getriggert haben, in dem Index als klassifizierte Kombinationen gekennzeichnet werden und wobei Klassenzuordnungen nur für Kombinationen aus zweiten zusätzlichen Token und ein oder mehreren identifizierten Triggern ausgeführt werden, welche nicht als klassifizierte Kombinationen gekennzeichnet sind.

4. Computerimplementiertes Verfahren nach einem der vorangehenden Ansprüche, wobei das Verfahren ferner umfasst:

- Identifizieren von ein oder mehr Trigger-Kombinationen, welche jeweils von zumindest einem der Datensätzen umfasst sind und ein Kombinationskriterium erfüllen,
- für jede der identifizierten Trigger-Kombinationen Kombinieren der Trigger-Definitionen der Trigger der entsprechenden Trigger-Kombinationen zu ein oder mehreren zusätzlichen kombinierten Trigger-Definitionen,
- Ergänzen der Mehrzahl von vorbestimmten Trigger-Definitionen des Lernmoduls durch die ein oder mehreren zusätzlichen kombinierten Trigger-Definitionen.

5. Computerimplementiertes Verfahren nach Anspruch 4, wobei das Kombinationskriterium eine Mindesthäufigkeit umfasst für ein Auftreten der entsprechenden Trigger-Kombination in den Datensätzen.

6. Computerimplementiertes Verfahren nach Anspruch 5, wobei die Mindesthäufigkeit einen absoluten Häufigkeitswert des Auftretens in den Datensätzen festlegt und/oder wobei die Mindesthäufigkeit einen relativen Häufigkeitswert des Auftretens in den Datensätzen festlegt relativ zu Häufigkeiten des Auftretens von ein oder mehreren der von der entsprechenden Trigger-Kombination umfassten Triggern in den Datensätzen.

7. Computerimplementiertes Verfahren nach einem der Ansprüche 4 bis 6, wobei das Kombinationskriterium ein oder mehrere Bedingungen an relative Positionen der Trigger der entsprechenden Trigger-Kombination zueinander innerhalb eines der Datensätze umfasst.

8. Computerimplementiertes Verfahren nach einem der vorangehenden Ansprüche, wobei die Trigger-Definitionen jeweils eine Definition einer Trigger-Struktur umfassen, welche für ein oder mehrere von der entsprechenden Trigger-Definition umfasste Trigger und ein oder mehrere gemäß der entsprechenden Trigger-Definition einer der Klassen zuzuordnende Token relative Positionen zueinander festlegt.

9. Computerimplementiertes Verfahren nach Anspruch 8, wobei die Festlegungen der relativen Positionen zumindest eine der folgenden Festlegungen umfassen: die ein oder mehreren zuzuordnenden Token sind nach einem von der entsprechenden Trigger-Definition umfassten Trigger angeordnet, die ein oder mehreren zuzuordnenden Token sind vor einem von der entsprechenden Trigger-Definition umfassten Trigger angeordnet, die ein oder mehreren zuzuordnenden Token sind jeweils zwischen von der entsprechenden Trigger-Definition umfassten Triggern angeordnet.

10. Computerimplementiertes Verfahren nach einem der Ansprüche 8 oder 9, wobei für die Trigger gemäß den Trigger-Definitionen jeweils ein maximaler Trigger-Abstand (152, 154) festgelegt ist, welcher einen maximalen Abstand relativ zu dem entsprechenden Trigger definiert, auf welche eine Trigger-Wirkung des Triggers beschränkt ist.

11. Computerimplementiertes Verfahren nach einem der vorangehenden Ansprüche, wobei das Verfahren ferner umfasst:

- Ergänzen des vortrainierten Lernmoduls um ein oder mehrere zusätzliche Trigger-Definitionen, welche zusätzliche Trigger für ein Ersetzen von Zuordnungen von Token in dem Index zu der Auffangklasse durch Zuordnungen zu ein oder mehreren Klassen einer zweiten Gruppe von Klassen im Zuge eines Reklassifizierens definieren,
- Reklassifizieren von ein oder mehreren der Auffangklasse zugeordneten Token in dem Index, welche die zusätzlichen Trigger-Definitionen als zusätzliche Trigger definierten, wobei das Reklassifizieren durch das Lernmodul ein Ersetzen der Zuordnung zu der Auffangklasse durch eine Zuordnung zu der entsprechen zusätzlichen Trigger-Definition umfasst, welche das entsprechende Token als zusätzlichen Trigger umfasst,
- Verwenden der zusätzlichen Trigger zum Reklassifizieren von ein oder mehreren der Auffangklasse zugeordneten Token in dem Index zu ein oder mehreren Klassen der zweiten Gruppe von Klassen durch das Lernmodul, falls die entsprechenden der Auffangklasse zugeordneten Token von einem der Datensätze in einer Kombination mit ein oder mehreren der zusätzlichen Triggern umfasst sind und die entsprechenden zusätzlichen Trigger gemäß der entsprechenden zusätzlichen Trigger-Definition eine entsprechende Zuordnung zu den ein oder mehreren Klassen der zweiten Gruppe von Klassen triggern.

12. Computerimplementiertes Verfahren nach Anspruch 11, wobei die zweite Gruppe von den Klassen der ersten Gruppe verschiedene Klassen umfasst oder wobei ein oder mehrere Klassen der zweiten Gruppe jeweils identisch sind mit einer der Klassen der ersten Gruppe.

13. Computerimplementiertes Verfahren nach einem der Ansprüche 11 bis 12, wobei die zu ergänzenden Trigger-Definitionen als Ergänzungen jeweils von einer von dem Lernmodul bereits umfassten Trigger-Definition abhängig sind.

14. Computerimplementiertes Verfahren nach Anspruch 13, wobei die Ergänzungen einem rekursiven Schema folgend wiederholt ausgeführt werden, wobei die zu ergänzenden Trigger-Definitionen jeder Rekursionsstufe jeweils Ergänzungen einer Trigger-Definition einer vorangehenden Rekursionsstufe umfassen, sodass die rekursiven Ergänzungen Baumstrukturen bilden, welche jeweils eine der vorbestimmten Trigger-Definition als Wurzelknoten umfassen.

15. Computerimplementiertes Verfahren nach einem der Ansprüche 11 bis 14, wobei die zu ergänzenden zusätzlichen Trigger-Definitionen von dem Lernmodul empfangen werden.

16. Computerimplementiertes Verfahren nach einem der Ansprüche 11 bis 14, wobei die zu ergänzenden zusätzlichen Trigger-Definitionen von dem Lernmodul erstellt werden, welches ein statistisches Modell umfasst, wobei das statistische Modell zu einer statistischen Analyse der von der Auffangklassen umfassten Token und deren Vorkommen in den Datensätze verwendet wird, wobei das Ergebnis der statistischen Analyse zum Erstellen der zu ergänzenden zusätzlichen Trigger-Definitionen verwendet wird.

17. Computerimplementiertes Verfahren nach einem der vorangehenden Ansprüche, wobei das Verfahren ferner umfasst:

- Empfang einer korrigierten Trigger-Definition zum Ersetzen einer der gespeicherten Trigger-Definitionen des Lernmoduls,
- Ersetzen der entsprechenden gespeicherten Trigger-Definition durch die korrigierte Trigger-Definition,

- Reklassifizieren der unter Verwendung der entsprechenden gespeicherten Trigger-Definition klassifizierten Token, wobei das Reklassifizieren unter Verwendung der korrigierten Trigger-Definition erfolgt.

18. Computerimplementiertes Verfahren nach einem der vorangehenden Ansprüche, wobei die Zeiger, mit denen die Token in dem Index verknüpft gespeichert sind, jeweils auf ein oder mehrere der Feldwerte in den gespeicherten Datensätze verweisen.

19. Computerimplementiertes Verfahren nach einem der vorangehenden Ansprüche, wobei die Feldwerte des zusätzlichen Datensatzes Textdaten, Bilddaten, Audiodaten und/oder Videodaten umfassen.

20. Computerimplementiertes Verfahren nach einem der vorangehenden Ansprüche, wobei das Erzeugen der Token ein Anwenden einer Tokenisierungslogik (122) auf die Feldwerte des zusätzlichen Datensatzes umfasst, welche einen Volltextindizierer umfasst, der dazu konfiguriert ist, Texte in Wörter zu zerlegen und die Wörter als Token auszugeben.

21. Computerimplementiertes Verfahren nach einem der Ansprüche 1 bis 19, wobei das Erzeugen der Token ein Anwenden einer Tokenisierungslogik (122) auf die Feldwerte des zusätzlichen Datensatzes umfasst, welche einen generischen Tokenisierer umfasst, der dazu konfiguriert ist, in den Feldwerten Daten unterschiedlichen Datentyps zu erkennen und aus diesen Token in unterschiedlichen Datentypen zu erzeugen.

22. Computerimplementiertes Verfahren nach einem der vorangehenden Ansprüche, wobei das Verfahren ferner umfasst:

- Empfangen einer Suchanfrage, wobei die Suchanfrage einen Suchwert beinhaltet,
- Durchsuchen des Index nach dem Suchwert, Identifizieren eines Tokens innerhalb des Index, welcher identisch ist mit dem Suchwert,

- Analysieren von Zeigern, mit denen das identifizierte Token verknüpft ist, um ein oder mehrere der Datensätze zu bestimmen, welche ein oder mehrere Feldwerte beinhalten, aus welchen das indizierte Token erzeugt wurde,
- Zurückgeben der bestimmten Datensätze oder von ein oder mehreren Referenzen auf die bestimmten Datensätze als Antwort auf die Suchanfrage.

23. Computerimplementiertes Verfahren nach Anspruch 22, wobei der Suchwert ferner eine Klassenzuordnung umfasst und das Identifizieren des Tokens innerhalb des Index ferner erfordert, dass das identifizierte Token dieselbe Klassenzuordnung aufweist.

24. Computerimplementiertes Verfahren nach einem der Ansprüche 22 bis 23, wobei Token, welche der Auffangklasse zugeordnet sind, von der Suche ausgeschlossen werden.

25. Computerimplementiertes Verfahren nach einem der vorangehenden Ansprüche, wobei das Verfahren ferner das Vortrainieren des Lernmoduls umfasst, wobei das Vortrainieren umfasst:

- Empfangen der Mehrzahl von vorbestimmten Trigger-Definitionen durch das Lernmodul, welche die Trigger für das Zuordnen von Token zu den Klassen der ersten Gruppe von Klassen definieren,
- Speichern der empfangenen Mehrzahl von vorbestimmten Trigger-Definitionen durch das Lernmodul.

26. Computerimplementiertes Verfahren nach Anspruch 25, wobei das Vortrainieren ferner umfasst:

- Extrahieren der Trigger aus der gespeicherten Mehrzahl von vorbestimmten Trigger-Definitionen durch das Lernmodul,
- Zuordnen der initialen Token durch das Lernmodul jeweils zu der Trigger-Definition, welche das entsprechende initiale Token als Trigger umfasst,
- Erzeugen des durchsuchbaren Index unter Verwendung der Mehrzahl von initialen Token durch das Multi-Modell-Datenbankmanagementsystem in dem

weiteren Datenmodell, wobei der erzeugte Index die initialen Token umfasst, wobei jedes der initialen Token in dem Index jeweils eine Zuordnung aufweist zu der Trigger-Definition, welche das entsprechende initiale Token als Trigger umfasst.

27. Computerimplementiertes Verfahren nach einem der vorangehenden Ansprüche, wobei der Index sämtliche aus den Feldwerten der gespeicherten Datensätze erzeugten Token so speichert, dass der Index die Token für jede der Klassenzuordnungen des entsprechenden Tokens genau einmal enthält.

28. Computerimplementiertes Verfahren nach einem der vorangehenden Ansprüche, wobei das weitere Datenmodell so strukturiert ist, dass die in dem weiteren Datenmodell gespeicherten Token und Token-Zuordnungen des Index die fünfte und/oder sechste Normalform erfüllen.

29. Computerimplementiertes Verfahren nach einem der vorangehenden Ansprüche, wobei es sich zumindest bei dem von dem Multi-Modell-Datenbankmanagementsystem zum Speichern der Datensätze verwendeten dokumentenbasierten Datenmodellen um ein NoSQL-Datenmodellen handelt.

30. Computerimplementiertes Verfahren nach einem der vorangehenden Ansprüche, wobei das maschinelle Lernen konfiguriert ist für eine Datenextraktion, Konsistenzprüfung, Bilderkennung, Spracherkennung, Sprachsteuerung, Vorrichtungsüberwachung und/oder autonome Vorrichtungssteuerung.

31. Computersystem (100) zum maschinellen Lernen, wobei das Computersystem ein oder mehrere Prozessoren (114) eine Datenbank (104), welche von ein oder mehrere Datenspeichermedien (102) bereitgestellt wird, ein Multi-Modell-Datenbankmanagementsystem (118), welches die Datenbank verwaltet und dazu konfiguriert ist, eine Mehrzahl von Datensätze (108; DS1, DS2, DS3) in einem dokumentenorientierten Datenmodell (106) in den Datenspeichermedien zu speichern, wobei die gespeicherten Datensätze jeweils ein oder mehreren Feldwerte umfassen,

wobei die einzelnen Feldwerte der gespeicherten Datensätze jeweils in einem Feld (F1, ..., F8) gespeichert werden, wobei die Feldwerte der gespeicherten Datensätze jeweils ein oder mehreren Feldtypen einer Mehrzahl unterschiedlicher Feldtypen zugeordnet sind, ein vortrainiertes Lernmodul (120) zum maschinellen Lernen und eine Programmlogik (116) umfasst,

wobei das vortrainierte Lernmodul eine Mehrzahl von vorbestimmten Trigger-Definitionen (123) umfasst, welche Trigger (121) für ein Zuordnen von Token (109) zu Klassen (111) einer ersten Gruppe von Klassen definieren,

wobei die Datenbank ferner einen durchsuchbaren Index (112) umfasst, welcher in einem weiteren Datenmodell (110) gespeichert ist, wobei der Index eine Mehrzahl von aus den Feldwerten der gespeicherten Datensätze erzeugten Token umfasst, wobei Token in dem Index jeweils mit einem oder mehreren Zeigern (115) auf ein oder mehrere der in dem dokumentenorientierten Datenmodell gespeicherten Datensätze verknüpft ist, aus deren Feldwerten das entsprechende Token erzeugt wurde,

wobei erste Token in dem Index, welche von einer der Trigger-Definitionen als Trigger umfasst sind, jeweils der entsprechen Trigger-Definition zugeordnet sind, wobei zweite Token in dem Index jeweils ein oder mehreren Klassen der ersten Gruppe von Klassen zugeordnet sind und wobei die verbleibenden Token in dem Index zum Kennzeichnen der entsprechenden verbleibenden Token als unbekannte Daten einer Auffangklasse zugeordnet sind, wobei die Zuordnung zu der Auffangklasse eine Zuordnung zu einer der Trigger-Definitionen ebenso wie eine Zuordnung zu einer der Klassen der ersten Gruppe von Klassen ausschließt,

wobei die Programmlogik zum Ausführen eines Verfahrens zum maschinellen Lernen konfiguriert ist, wobei das Verfahren umfasst:

- Empfangen eines zusätzlichen Datensatzes,
- Speichern des zusätzlichen Datensatzes, welcher ein oder mehrere zusätzliche Feldwerte umfasst, durch das Multi-Modell-Datenbankmanagementsystem in dem dokumentenorientierten Datenmodell der Datenbank,
- Erzeugen von ein oder mehreren zusätzlichen Token aus den zusätzlichen Feldwerten,

- falls eines oder mehrere erste zusätzliche Token von einer der Trigger-Definitionen als Trigger umfasst sind, Identifizieren des entsprechenden Tokens als Trigger durch das Lernmodul,
- Verwenden der identifizieren Trigger zum Zuordnen von ein oder mehreren zweiten zusätzlichen Token zu ein oder mehreren Klassen der ersten Gruppe von Klassen durch das Lernmodul, falls die entsprechenden zweiten zusätzlichen Token von dem zusätzliche Datensatz in einer Kombination mit ein oder mehreren der identifizierten Trigger gemäß einer der Trigger-Definitionen umfasst sind, wobei die entsprechenden Trigger gemäß der entsprechenden Trigger-Definition eine entsprechende Klassenzuordnung triggern,
- Zuordnen der verbleibenden zusätzlichen Token, für welche keine Zuordnung zu einer der Trigger-Definitionen und keine Klassenzuordnung aufgrund einer der Trigger-Definitionen erfolgt ist, zu der Auffangklasse,
- Ergänzen des Index durch das Multi-Modell-Datenbankmanagementsystem unter Verwendung der zusätzlichen Token, der Klassenzuordnungen der zusätzlichen Token und eines Zeigers auf den zusätzlichen in dem dokumentenorientierten Datenmodell gespeicherten Datensatz.

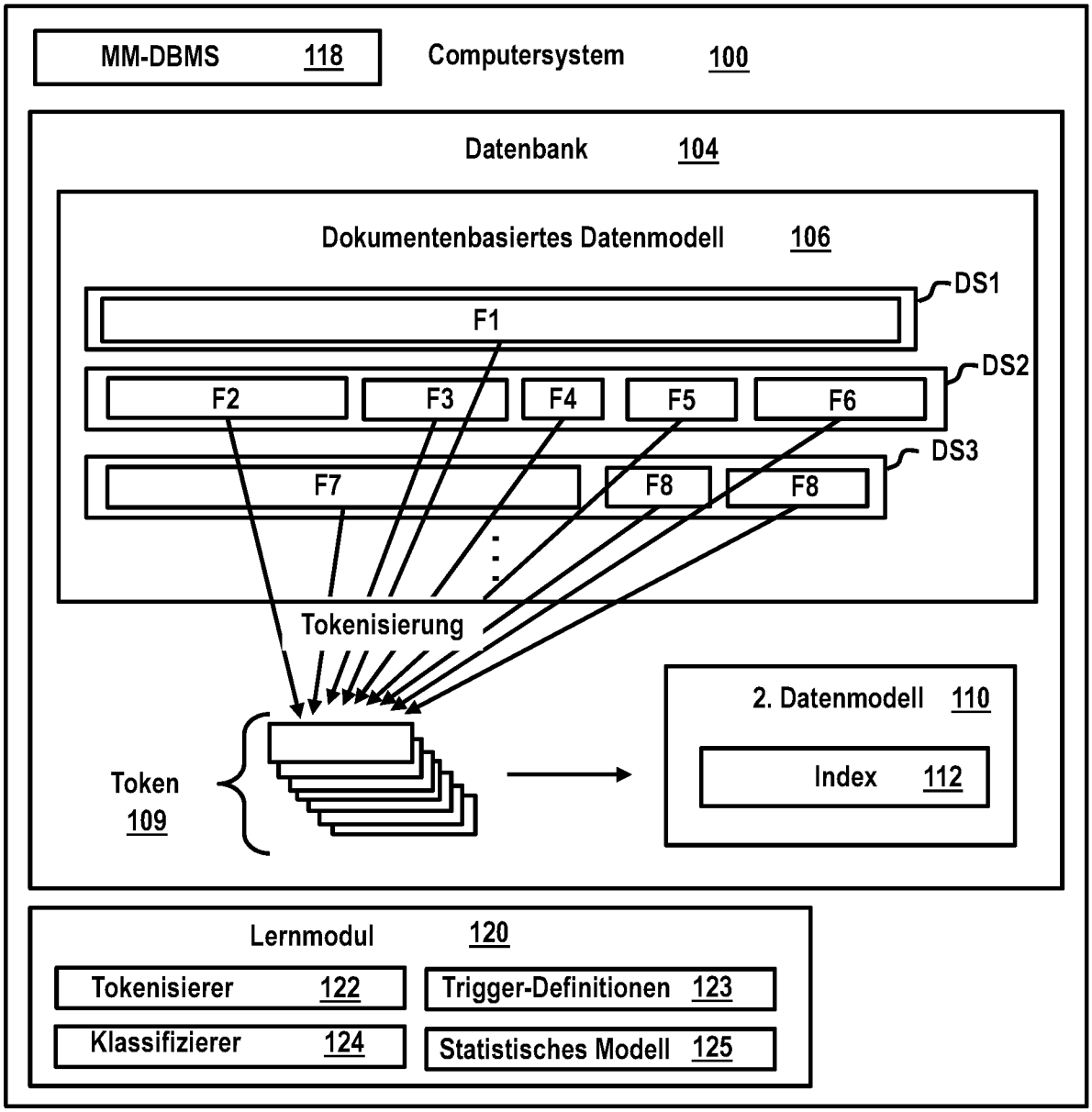
MM-DBMS 118
Computersystem 100
Datenbank 104
Dokumentenbasiertes Datenmodell 106
DS1
F1
DS2
F2
F3
F4
F5
F6
DS3
F7
F8
F8
Tokenisierung
2. Datenmodell 110
Index 112
Token
109
Lernmodul 120
Tokenisierer 122
Trigger-Definitionen 123
Klassifizierer 124
Statistisches Modell 125

Fig. 1

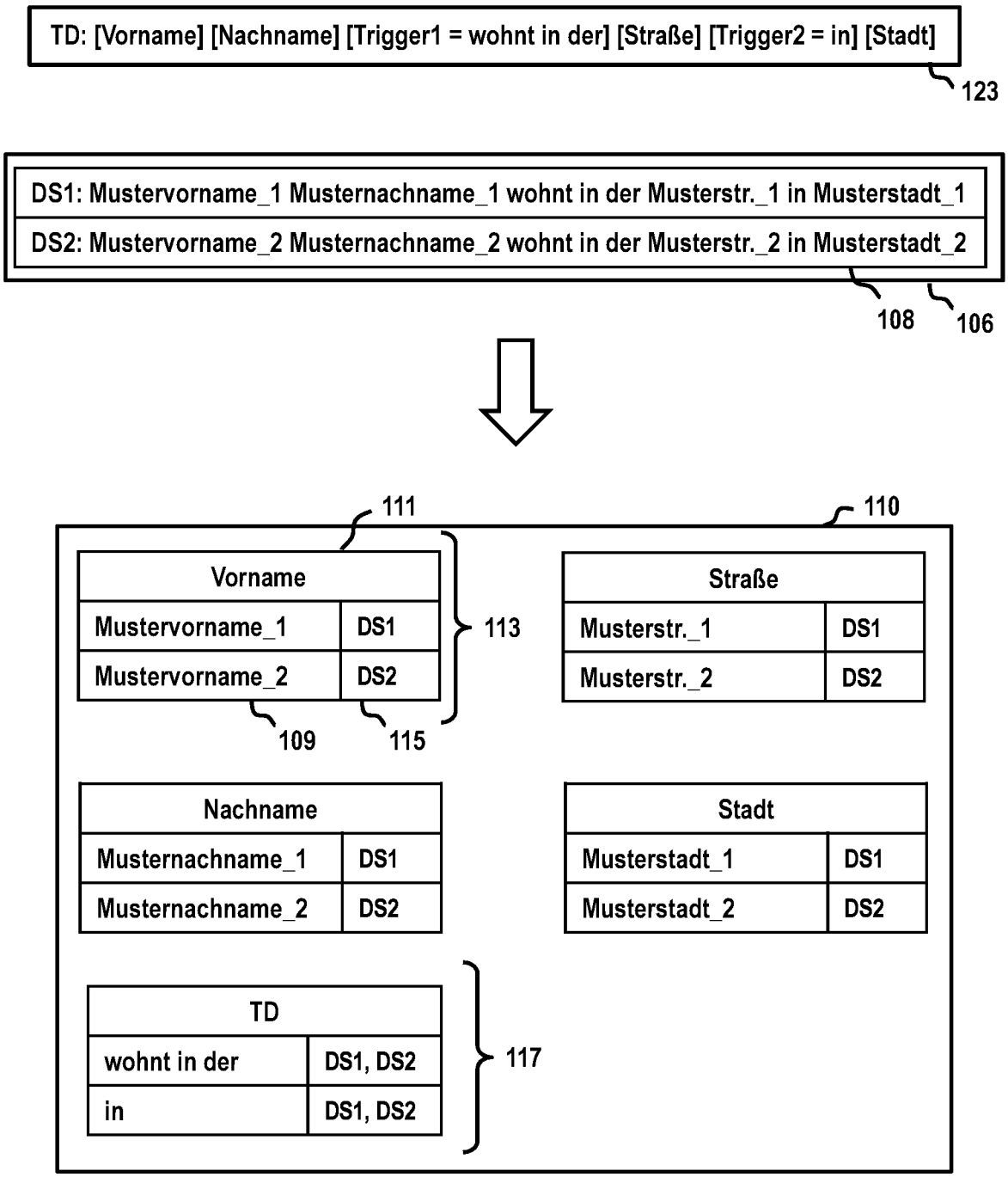
TD: [Vorname] [Nachname] [Trigger1 = wohnt in der] [Straße] [Trigger2 = in] [Stadt]
123
DS1: Mustervorname_1 Musternachname_1 wohnt in der Musterstr._1 in Musterstadt_1
DS2: Mustervorname_2 Musternachname_2 wohnt in der Musterstr._2 in Musterstadt_2
108
106
111
110
Vorname
Mustervorname_1 DS1
Mustervorname_2 DS2
113
109
115
Straße
Musterstr._1 DS1
Musterstr._2 DS2
Nachname
Musternachname_1 DS1
Musternachname_2 DS2
Stadt
Musterstadt_1 DS1
Musterstadt_2 DS2
TD
wohnt in der DS1, DS2
in DS1, DS2
117

Fig. 2

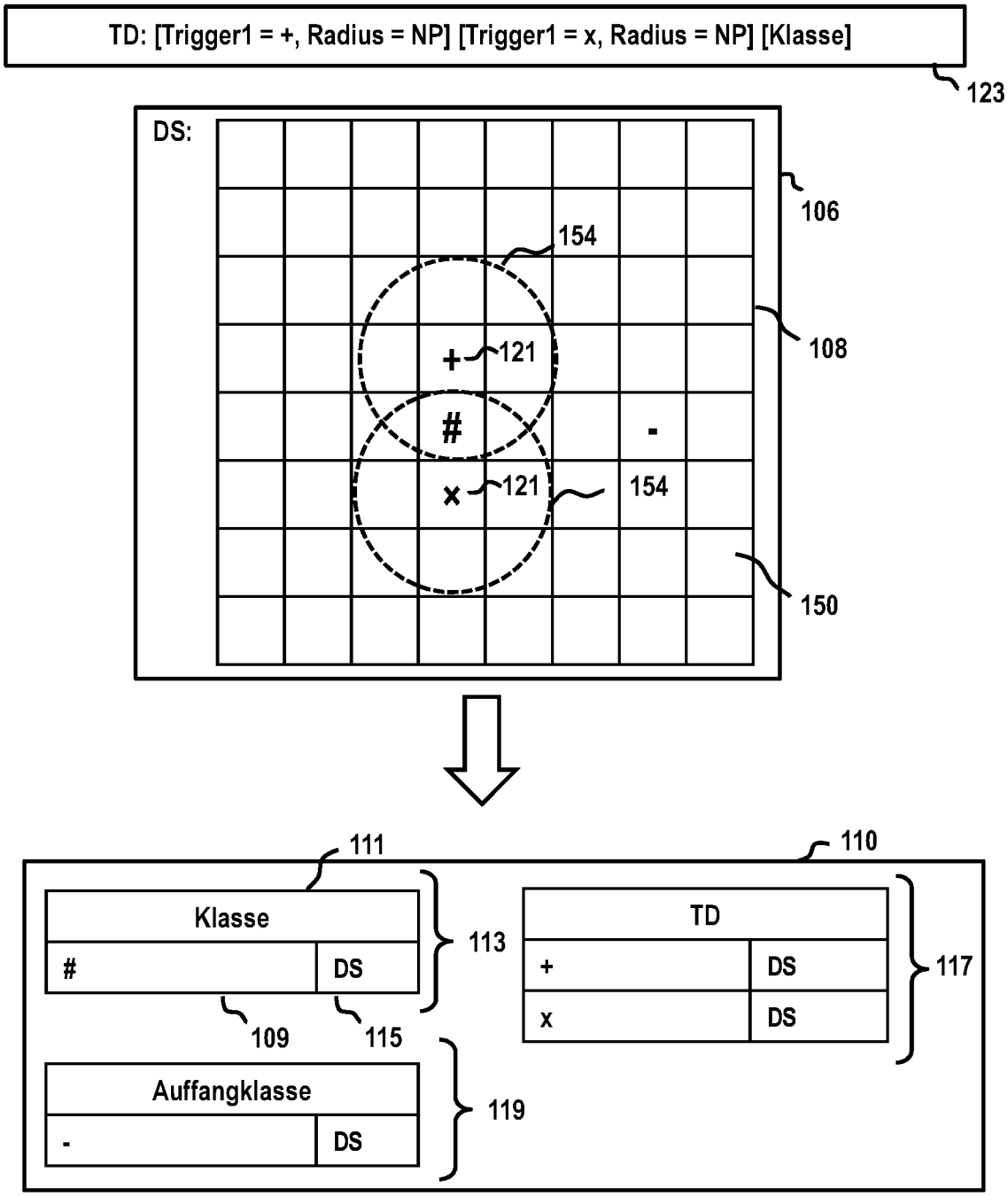
TD: [Trigger1 = +, Radius = NP] [Trigger1 = x, Radius = NP] [Klasse]
123
DS:
106
154
108
+
121
#
-
x
121
154
150
111
110
Klasse
#
DS
113
109
115
TD
+
DS
x
DS
117
Auffangklasse
-
DS
119

Fig. 3

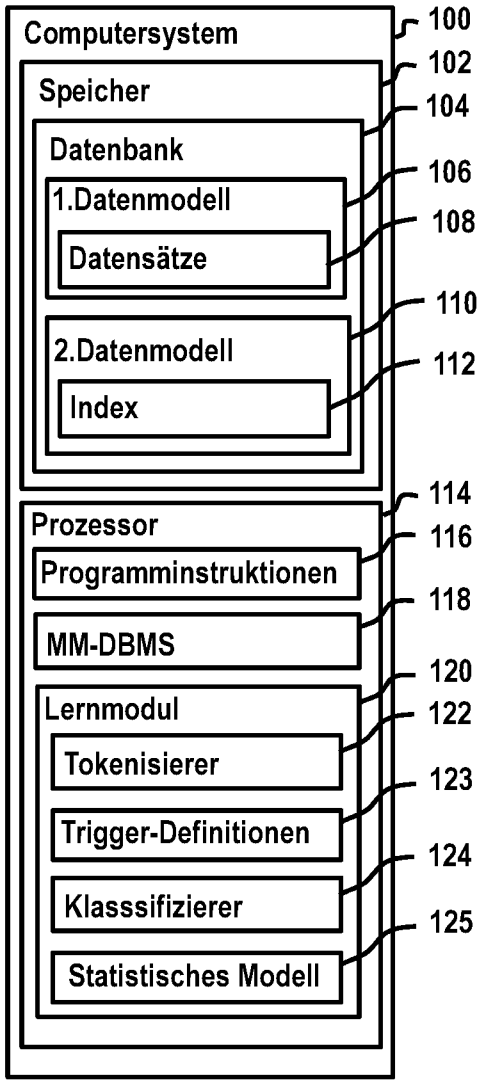
Computersystem
100
Speicher
102
Datenbank
104
1.Datenmodell
106
Datensätze
108
2.Datenmodell
110
Index
112
Prozessor
114
Programminstruktionen
116
MM-DBMS
118
Lernmodul
120
Tokenisierer
122
Trigger-Definitionen
123
Klasssifizierer
124
Statistisches Modell
125

Fig. 4

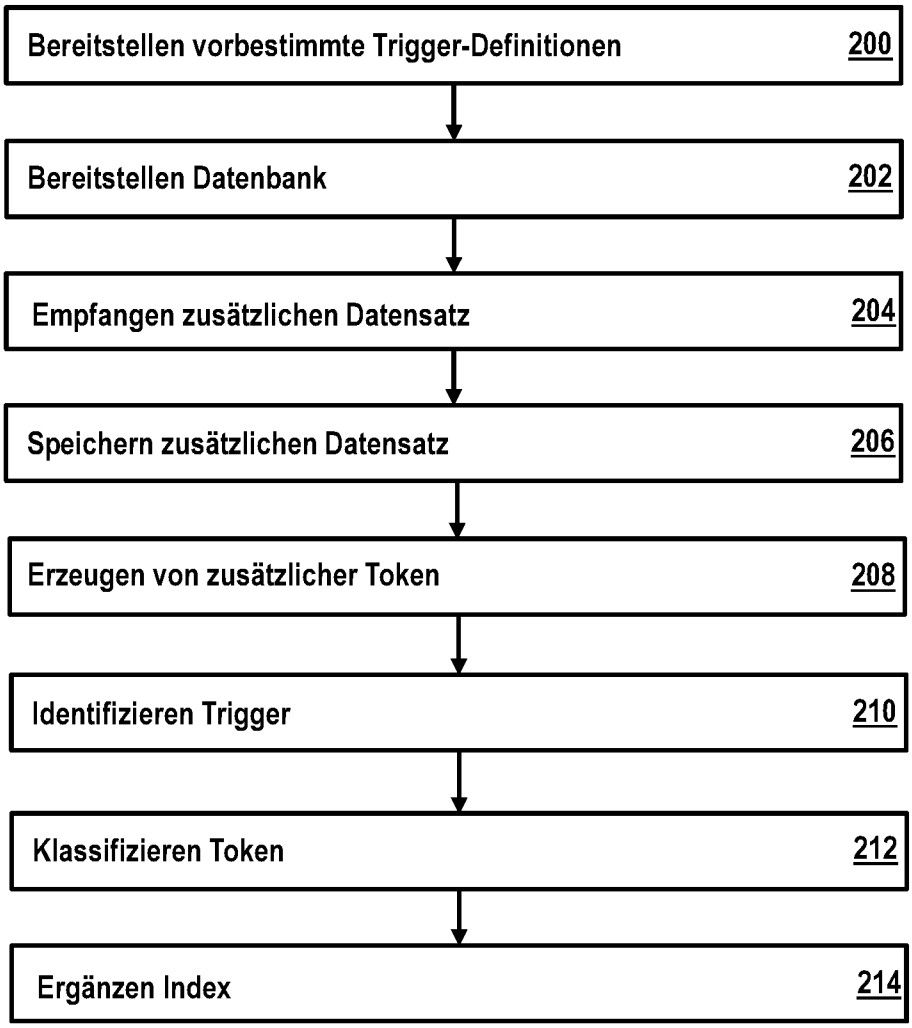
Bereitstellen vorbestimmte Trigger-Definitionen 200
Bereitstellen Datenbank 202
Empfangen zusätzlichen Datensatz 204
Speichern zusätzlichen Datensatz 206
Erzeugen von zusätzlicher Token 208
Identifizieren Trigger 210
Klassifizieren Token 212
Ergänzen Index 214

Fig. 5

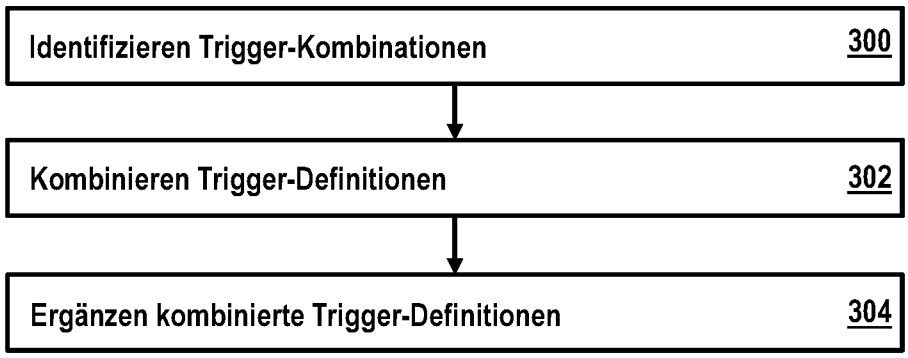
Identifizieren Trigger-Kombinationen
300
Kombinieren Trigger-Definitionen
302
Ergänzen kombinierte Trigger-Definitionen
304

Fig. 6

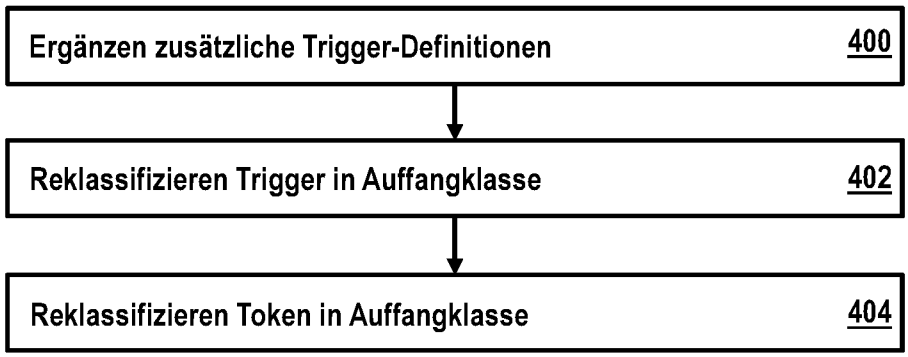
Ergänzen zusätzliche Trigger-Definitionen
400
Reklassifizieren Trigger in Auffangklasse
402
Reklassifizieren Token in Auffangklasse
404

Fig. 7

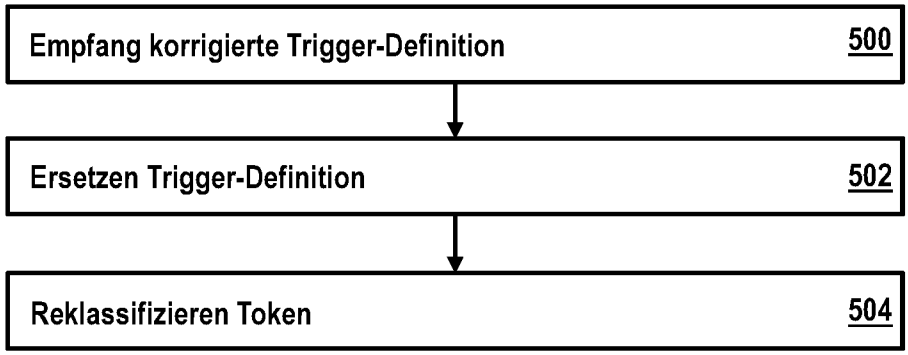
Empfang korrigierte Trigger-Definition
500
Ersetzen Trigger-Definition
502
Reklassifizieren Token
504

Fig. 8

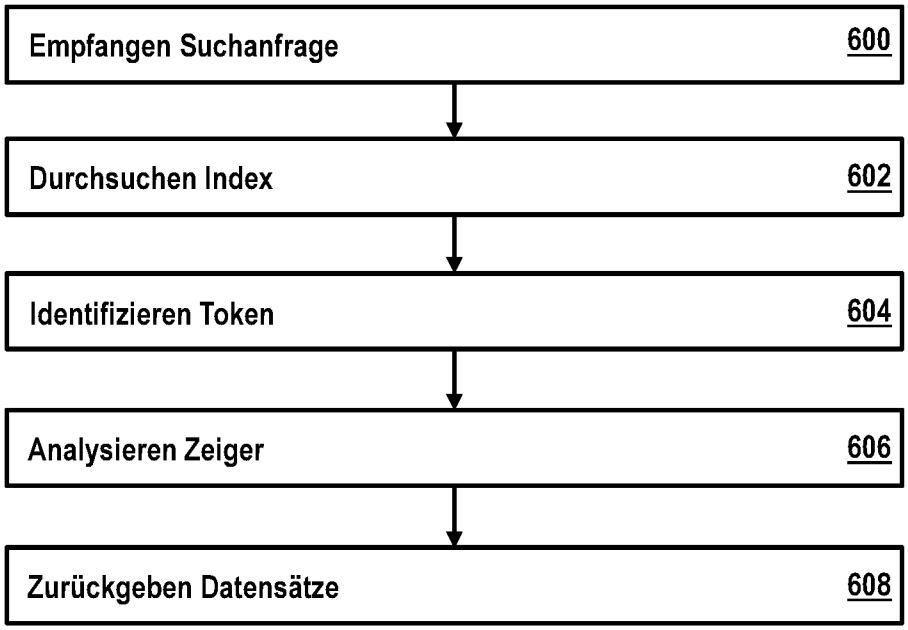
Empfangen Suchanfrage
600
Durchsuchen Index
602
Identifizieren Token
604
Analysieren Zeiger
606
Zurückgeben Datensätze
608

Fig. 9

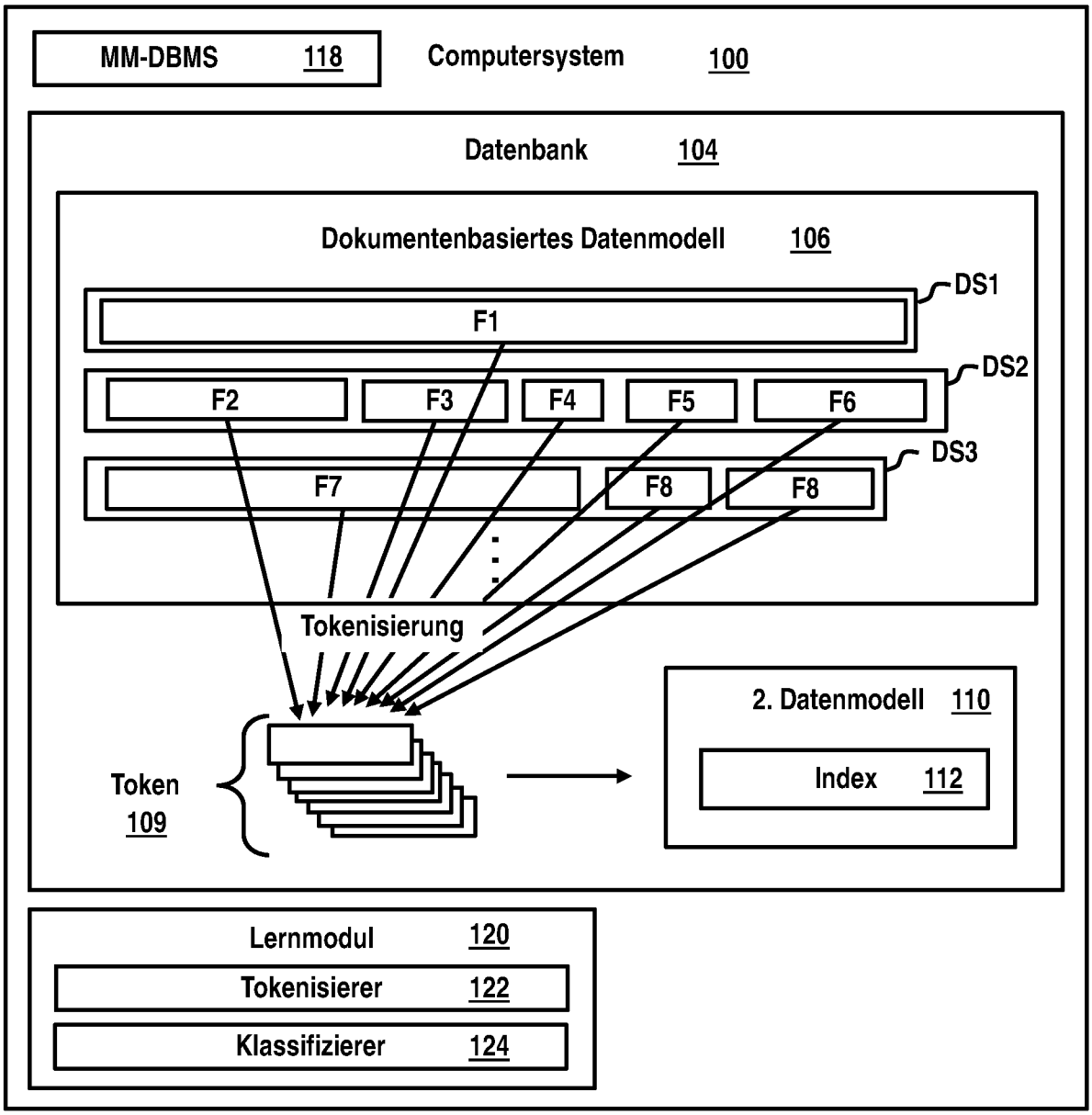
MM-DBMS 118
Computersystem 100
Datenbank 104
Dokumentenbasiertes Datenmodell 106
DS1
F1
DS2
F2
F3
F4
F5
F6
DS3
F7
F8
F8
Tokenisierung
2. Datenmodell 110
Index 112
Token
109
Lernmodul 120
Tokenisierer 122
Klassifizierer 124

Fig. 1

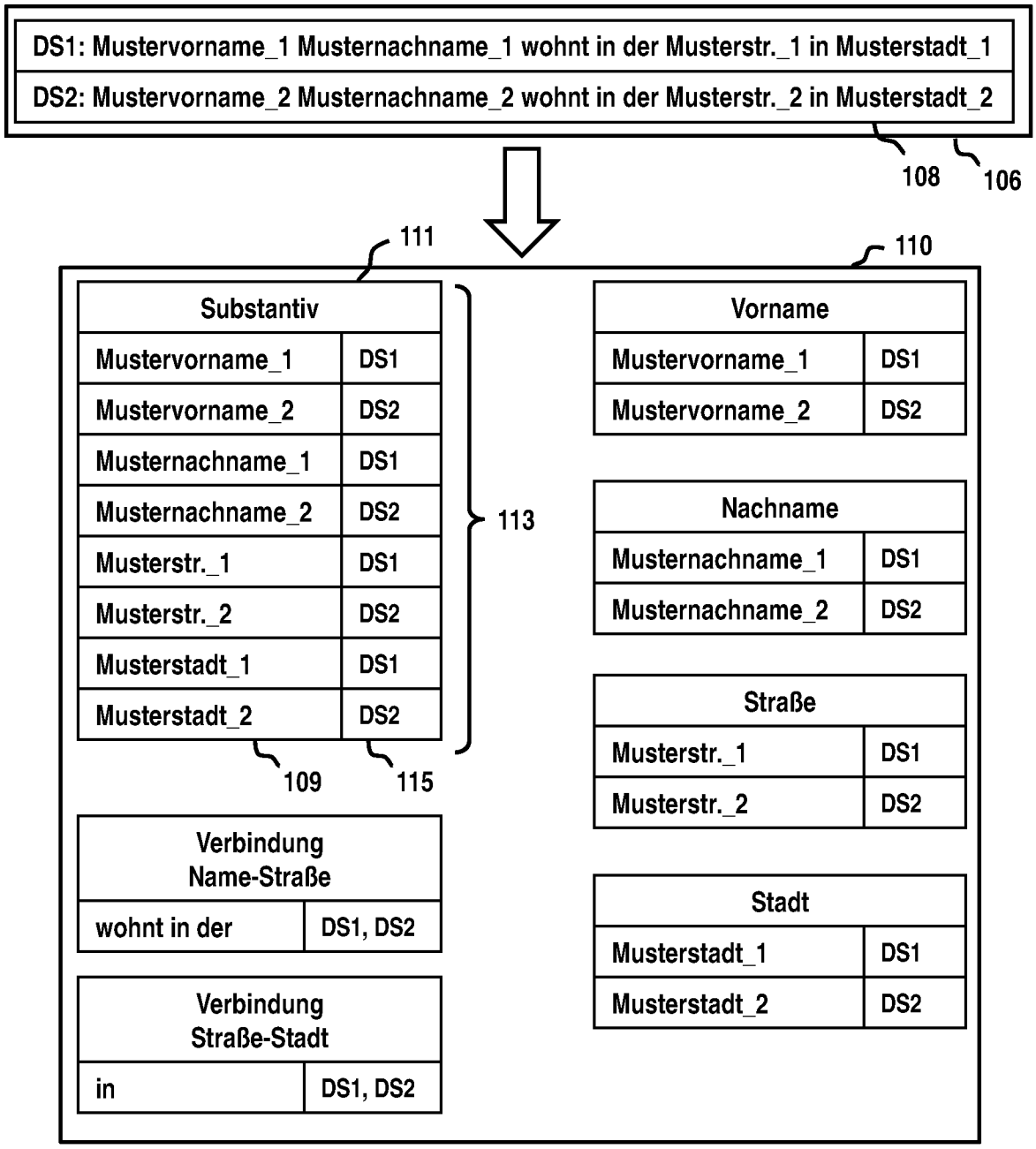
DS1: Mustervorname_1 Musternachname_1 wohnt in der Musterstr._1 in Musterstadt_1
DS2: Mustervorname_2 Musternachname_2 wohnt in der Musterstr._2 in Musterstadt_2
108
106
111
110
Substantiv
Mustervorname_1 DS1
Mustervorname_2 DS2
Musternachname_1 DS1
Musternachname_2 DS2
Musterstr._1 DS1
Musterstr._2 DS2
Musterstadt_1 DS1
Musterstadt_2 DS2
113
109
115
Vorname
Mustervorname_1 DS1
Mustervorname_2 DS2
Nachname
Musternachname_1 DS1
Musternachname_2 DS2
Straße
Musterstr._1 DS1
Musterstr._2 DS2
Stadt
Musterstadt_1 DS1
Musterstadt_2 DS2
Verbindung Name-Straße
wohnt in der DS1, DS2
Verbindung Straße-Stadt
in DS1, DS2

Fig. 2

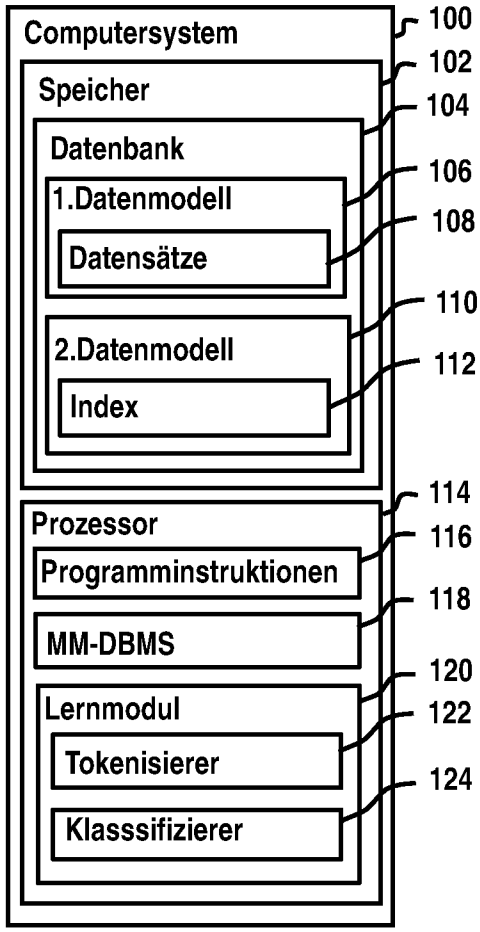
Computersystem
100
Speicher
102
Datenbank
104
1.Datenmodell
106
Datensätze
108
2.Datenmodell
110
Index
112
Prozessor
114
Programminstruktionen
116
MM-DBMS
118
Lernmodul
120
Tokenisierer
122
Klasssifizierer
124

Fig. 3

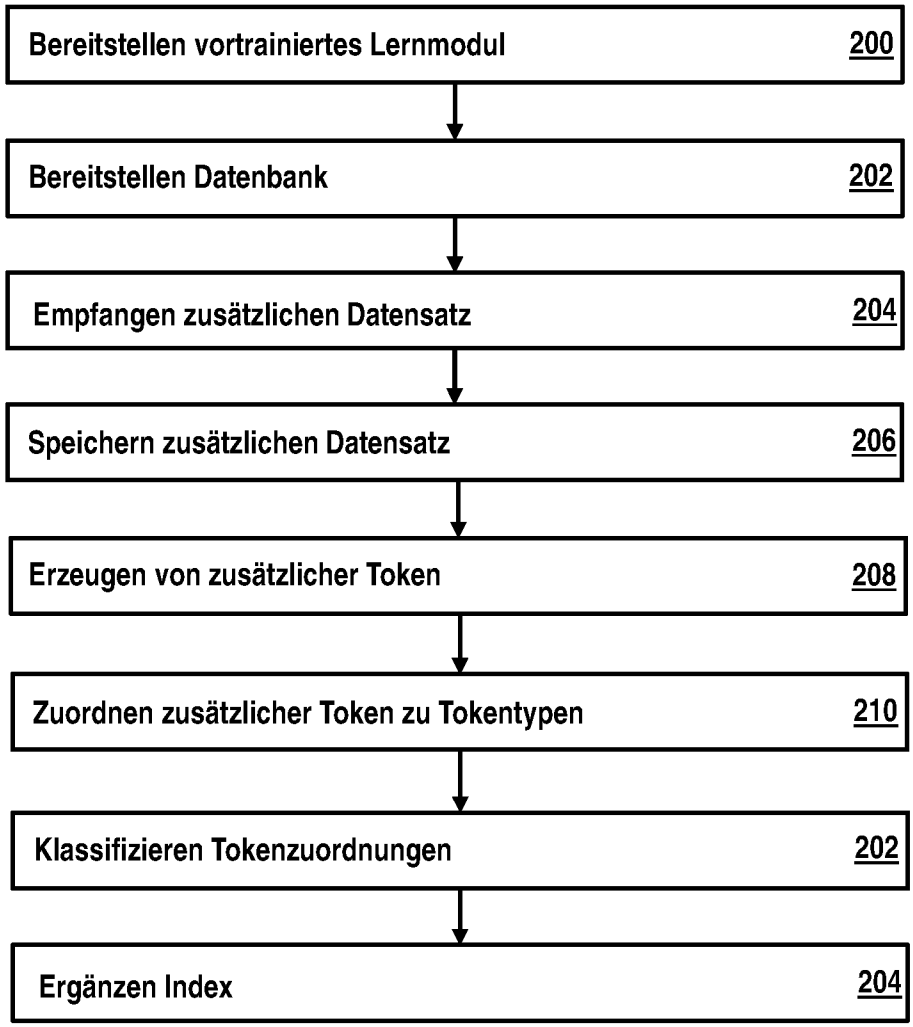
Bereitstellen vortrainiertes Lernmodul 200
Bereitstellen Datenbank 202
Empfangen zusätzlichen Datensatz 204
Speichern zusätzlichen Datensatz 206
Erzeugen von zusätzlicher Token 208
Zuordnen zusätzlicher Token zu Tokentypen 210
Klassifizieren Tokenzuordnungen 202
Ergänzen Index 204

Fig. 4

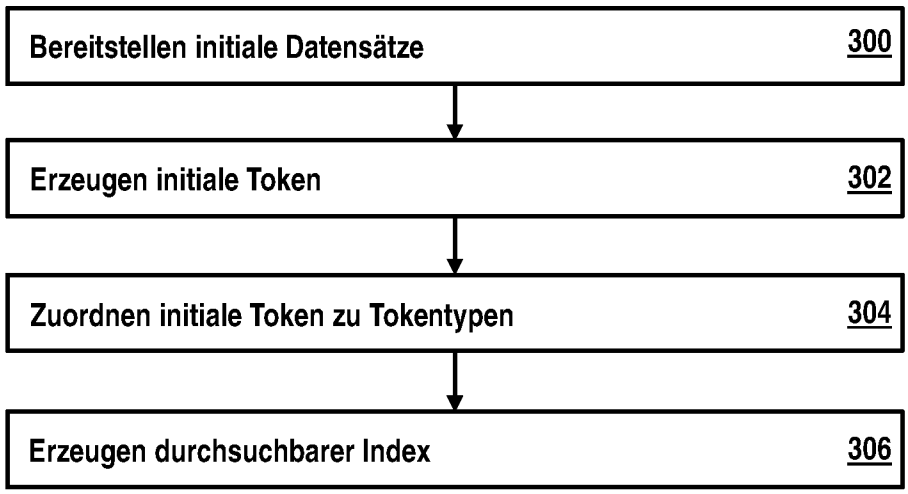
Bereitstellen initiale Datensätze
300
Erzeugen initiale Token
302
Zuordnen initiale Token zu Tokentypen
304
Erzeugen durchsuchbarer Index
306

Fig. 5

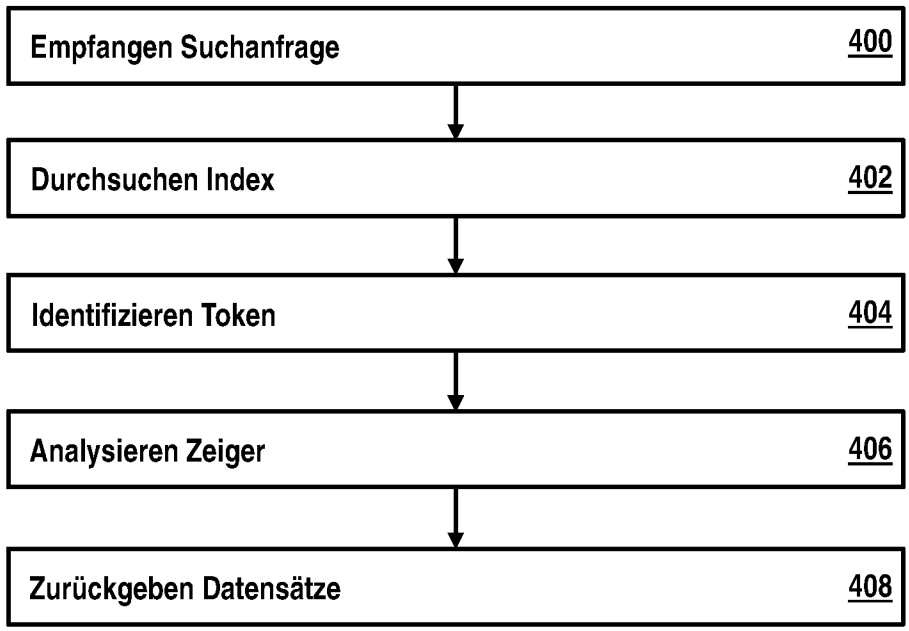
Empfangen Suchanfrage
400
Durchsuchen Index
402
Identifizieren Token
404
Analysieren Zeiger
406
Zurückgeben Datensätze
408

Fig. 6

INTERNATIONAL SEARCH REPORT

International application No.
PCT/EP2020/059042

A. CLASSIFICATION OF SUBJECT MATTER

G06F 16/31(2019.01)i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

EPO-Internal

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	DE 102016226338 A1 (BUNDESDRUCKEREI GMBH [DE]) 05 July 2018 (2018-07-05) paragraph [0009] - paragraph [0017] paragraph [0036] - paragraph [0046] paragraph [0052] paragraph [0054] paragraph [0080] - paragraph [0083]	1-31
X	DE 19627472 A1 (SER SYSTEME AG [DE]) 15 January 1998 (1998-01-15) column 17 - column 18; claims 3-6; figure 11	1-31
X	DE 102010043265 A1 (SYMANTEC CORP [US]) 12 May 2011 (2011-05-12) paragraph [0003] - paragraph [0007] paragraph [0036] - paragraph [0039]	1-31

☐ Further documents are listed in the continuation of Box C. ☑ See patent family annex.

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
08 June 2020	**26 June 2020**

Name and mailing address of the ISA/EP	Authorized officer
European Patent Office **p.b. 5818, Patentlaan 2, 2280 HV Rijswijk** **Netherlands** Telephone No. **(+31-70)340-2040** Facsimile No. **(+31-70)340-3016**	**Frank, Korbinian** Telephone No.

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/EP2020/059042

Patent document cited in search report	Publication date (day/month/year)	Patent family member(s)	Publication date (day/month/year)
DE 102016226338 A1	05 July 2018	DE 102016226338 A1	05 July 2018
		EP 3563261 A1	06 November 2019
		WO 2018122269 A1	05 July 2018
DE 19627472 A1	15 January 1998	AU 3538697 A	02 February 1998
		DE 19627472 A1	15 January 1998
		EP 0910829 A1	28 April 1999
		US 2002133476 A1	19 September 2002
		WO 9801808 A1	15 January 1998
DE 102010043265 A1	12 May 2011	CN 102054022 A	11 May 2011
		DE 102010043265 A1	12 May 2011
		GB 2475151 A	11 May 2011
		JP 5586425 B2	10 September 2014
		JP 2011100457 A	19 May 2011
		US 2011113466 A1	12 May 2011

INTERNATIONALER RECHERCHENBERICHT

Internationales Aktenzeichen
PCT/EP2020/059042

A. KLASSIFIZIERUNG DES ANMELDUNGSGEGENSTANDES
INV. G06F16/31
ADD.

Nach der Internationalen Patentklassifikation (IPC) oder nach der nationalen Klassifikation und der IPC

B. RECHERCHIERTE GEBIETE

Recherchierter Mindestprüfstoff (Klassifikationssystem und Klassifikationssymbole)
G06F

Recherchierte, aber nicht zum Mindestprüfstoff gehörende Veröffentlichungen, soweit diese unter die recherchierten Gebiete fallen

Während der internationalen Recherche konsultierte elektronische Datenbank (Name der Datenbank und evtl. verwendete Suchbegriffe)
EPO-Internal

C. ALS WESENTLICH ANGESEHENE UNTERLAGEN

Kategorie*	Bezeichnung der Veröffentlichung, soweit erforderlich unter Angabe der in Betracht kommenden Teile	Betr. Anspruch Nr.
X	DE 10 2016 226338 A1 (BUNDESDRUCKEREI GMBH [DE]) 5. Juli 2018 (2018-07-05) Absatz [0009] - Absatz [0017] Absatz [0036] - Absatz [0046] Absatz [0052] Absatz [0054] Absatz [0080] - Absatz [0083] -----	1-31
X	DE 196 27 472 A1 (SER SYSTEME AG [DE]) 15. Januar 1998 (1998-01-15) Spalte 17 - Spalte 18; Ansprüche 3-6; Abbildung 11 -----	1-31
X	DE 10 2010 043265 A1 (SYMANTEC CORP [US]) 12. Mai 2011 (2011-05-12) Absatz [0003] - Absatz [0007] Absatz [0036] - Absatz [0039] -----	1-31

☐ Weitere Veröffentlichungen sind der Fortsetzung von Feld C zu entnehmen ☒ Siehe Anhang Patentfamilie

* Besondere Kategorien von angegebenen Veröffentlichungen :
"A" Veröffentlichung, die den allgemeinen Stand der Technik definiert, aber nicht als besonders bedeutsam anzusehen ist
"E" frühere Anmeldung oder Patent, die bzw. das jedoch erst am oder nach dem internationalen Anmeldedatum veröffentlicht worden ist
"L" Veröffentlichung, die geeignet ist, einen Prioritätsanspruch zweifelhaft erscheinen zu lassen, oder durch die das Veröffentlichungsdatum einer anderen im Recherchenbericht genannten Veröffentlichung belegt werden soll oder die aus einem anderen besonderen Grund angegeben ist (wie ausgeführt)
"O" Veröffentlichung, die sich auf eine mündliche Offenbarung, eine Benutzung, eine Ausstellung oder andere Maßnahmen bezieht
"P" Veröffentlichung, die vor dem internationalen Anmeldedatum, aber nach dem beanspruchten Prioritätsdatum veröffentlicht worden ist
"T" Spätere Veröffentlichung, die nach dem internationalen Anmeldedatum oder dem Prioritätsdatum veröffentlicht worden ist und mit der Anmeldung nicht kollidiert, sondern nur zum Verständnis des der Erfindung zugrundeliegenden Prinzips oder der ihr zugrundeliegenden Theorie angegeben ist
"X" Veröffentlichung von besonderer Bedeutung; die beanspruchte Erfindung kann allein aufgrund dieser Veröffentlichung nicht als neu oder auf erfinderischer Tätigkeit beruhend betrachtet werden
"Y" Veröffentlichung von besonderer Bedeutung; die beanspruchte Erfindung kann nicht als auf erfinderischer Tätigkeit beruhend betrachtet werden, wenn die Veröffentlichung mit einer oder mehreren Veröffentlichungen dieser Kategorie in Verbindung gebracht wird und diese Verbindung für einen Fachmann naheliegend ist
"&" Veröffentlichung, die Mitglied derselben Patentfamilie ist

Datum des Abschlusses der internationalen Recherche	Absendedatum des internationalen Recherchenberichts
8. Juni 2020	26/06/2020

Name und Postanschrift der Internationalen Recherchenbehörde	Bevollmächtigter Bediensteter
Europäisches Patentamt, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Fax: (+31-70) 340-3016	Frank, Korbinian

1

INTERNATIONALER RECHERCHENBERICHT

Angaben zu Veröffentlichungen, die zur selben Patentfamilie gehören

Internationales Aktenzeichen
PCT/EP2020/059042

Im Recherchenbericht angeführtes Patentdokument	Datum der Veröffentlichung	Mitglied(er) der Patentfamilie	Datum der Veröffentlichung
DE 102016226338 A1	05-07-2018	DE 102016226338 A1	05-07-2018
		EP 3563261 A1	06-11-2019
		WO 2018122269 A1	05-07-2018
DE 19627472 A1	15-01-1998	AU 3538697 A	02-02-1998
		DE 19627472 A1	15-01-1998
		EP 0910829 A1	28-04-1999
		US 2002133476 A1	19-09-2002
		WO 9801808 A1	15-01-1998
DE 102010043265 A1	12-05-2011	CN 102054022 A	11-05-2011
		DE 102010043265 A1	12-05-2011
		GB 2475151 A	11-05-2011
		JP 5586425 B2	10-09-2014
		JP 2011100457 A	19-05-2011
		US 2011113466 A1	12-05-2011