Verteilte Vektordarstellung von semantischen Reaktionen für eine künstliche Dialogumgebung

Distributed Vector Representation of Semantic Reactions for an Artifical Dialogue Environment

Nhu Huy Le

BACHELORARBEIT

Nr. 5736012

eingereicht am

Bachelorstudiengang

Informatik

in München

im April 2017

Betreuer:

Dr. Oliver Braun

Erklärung

Ich erkläre eidesstattlich, dass ich die vorliegende Arbeit selbstständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen nicht benutzt und die den benutzten Quellen entnommenen Stellen als solche gekennzeichnet habe. Die Arbeit wurde bisher in gleicher oder ähnlicher Form keiner anderen Prüfungsbehörde vorgelegt.

München, am 7. April 2017

Nhu Huy Le

Inhaltsverzeichnis

Kurzfassung

Künstliche Agenten werden zunehmend in schriftlichen Dialogumgebungen eingesetzt. Ein Problem dabei ist die semantische Erkennung bzw. Unterscheidung der Intention von natürlichen Aussagen. Diese Arbeit untersucht einen abfragenden Ansatz zur frühen Erkennung sowie der effizienten Speicherung und Suche eingebetteter Reaktionen für die Intention vergleichbar mit der maschinellen Sprachübersetzung. Dabei wird schrittweise in die funktionale Entwicklung abfragender Dialogmodelle eingeführt. Schließlich wird mit modernen Bibliotheken und Frameworks eine mobile Umgebung einschließlich eines künstlichen Dialogagenten implementiert. Auf deutschen Texten neuronal trainierte, verteilte Worteinbettungen im Vektorraum (auch *Word2Vec* genannt) und ältere Dialogprotokolle bilden die semantische Grundlage für den Agenten.

Die Einrichtung der Erkennungsroutinen ist aufgrund von verfügbaren Frameworks unkompliziert: Das System kann dabei aufgabenbasiert strukturiert werden, um zukünftige Portierungen der künstlichen Logik im Zuge schneller Entwicklungen zu erleichtern. Mit Reaktionseinbettungen wird die semantische Erkennung intuitiver und persönlicher, dennoch hängt die Erfolgsrate hauptsächlich von der Datenquelle, der Datenmenge und der vorherigen Datenverarbeitung ab. Mit wachsender Datenmenge ist menschliche Emotionalität dabei schneller abzubilden als menschliche Rationalität. Die Entwicklung von Dialogagenten auf der Grundlage neuerer, möglicherweise generativer Modelle sind daher vielversprechend.

Abstract

Artifical agents are increasingly utilised in textual dialogue environments. One issue involved is the semantic recognition or distinction of the intention in natural utterances. This work examines a retrieval-based approach for early recognition as well as efficient storage and search of embedded reactions to intentions comparable with machine translation. A stepwise introduction to the functional development of retrieval-based dialogue models is elaborated. Eventually, with modern libraries and frameworks, a mobile environment including an artifical dialogue agent (conversational agent) is implemented, semantically based on older dialogue records and neural word embeddings (also known as *word2vec*), trained on German corpus.

The setup of the recognition routines is straightforward because of the frameworks available: The system structure may be separated by its tasks to allow easy transfers of the artifical logic due to fast developments. With introduction of reaction embeddings the semantic recognition gets more intuitive and personal, nevertheless the success rate depends mainly on the amount of data and the prior processing of the data. With increasing data size, human emotionality is faster portrayed than human rationality. The development of dialogue agents based on newer, possibly generative models are thus promising.

Kapitel 1

Motivation

1.1 Verbreitung von Dialogagenten

Bemerkenswerte Fortschritte in der Verarbeitung der natürlichen Sprache und damit verbunden die Interaktion mit künstlichen Maschinen eröffnen neue Möglichkeiten in der Automatisierung. So ist zwar die Erkennung von Intention und die Reaktion mit faktischem Wissen alltagstauglich, doch die Logiksysteme für letztere Expertensysteme werden noch mühsam manuell zusammengetragen. Neuere Modelle bzw. Schnittstellen für die Entwicklung solcher erfordern daher die Implementierung möglichst vieler passender Szenarien, unter denen ein Mensch mit dem System kommunizieren könnte[1]. Dabei sind sie für jeden frei zugänglich. Andere wiederum versuchen aus der vernetzten Welt heraus, automatisierte Kommunikation mit der riesigen Datenmenge von Nachrichten sozialer Netzwerke zu generieren und erschaffen dadurch eine Art „globales Bewusstsein“[2]. Diese Möglichkeit ist in der Regel den großen Unternehmen vorenthalten.

Nun bleiben die Erinnerungen unbeachtet, die zu jeder Sekunde über private, meist mobile Nachrichtanwendungen geschaffen werden, aber mit ihrer typischen Persönlichkeit hinter großen Datenbanken vergessen werden; oder die vielen beendeten Dialoge mit Kunden, die ein mittelständisches Unternehmen trotz gleichen Inhaltes immer wiederholen muss. Mit der Verbreitung von Nachrichtenanwendungen steigt daher auch die Attraktivität der intelligenten Wiederbenutzung ihrer Persistenz.

[1] *Pandorabots*, *api.ai*, *wit.ai* etc. stellen teilweise solche Dienste dar.
[2] Microsofts *Xaoice* bzw. *Tay* basieren auf dieser Verarbeitung großer Datenmengen.

1.2 Vorschau

Diese Arbeit zeigt (Kap. 2) die primären Anforderungen, die eine Automatisierung und semantische Verarbeitung von Dialogen bemisst. Dazu definiert sie die Strukturen eines (natürlichen) Dialoges und führt (Kap. 3) in die historische und funktionale Entwicklung der Automatisierung von Dialogen ein. Zusätzlich schlägt die Arbeit (Kap. 4) einen Ansatz vor, um ältere oder aktive Dialoge direkt nutzbar in eine Automatisierung einzubinden. Dabei werden die Möglichkeiten und Einschränkungen im Hinblick auf zukünftige Einsätze und Erweiterungen des Ansatzes erörtert (Kap. 5).

Kapitel 2

Problemstellung

2.1 Anforderungen an einem Dialog

Bei einem Dialog handelt es sich um einen Austausch gegenseitig relevanter, sich ergänzender Informationen, also eine Form von **Kommunikation**. Innerhalb einer künstlichen Dialogumgebung können diese Informationen von zwei verschiedenen (Dialog-)Agenten d_1 und d_2 aus einer Menge von Agenten $D = \{d_1, ..., d_n\}_{n \in \mathbb{N}}$ kommen[1] und setzen dafür Sprachverständnis, Argumentation, Entscheidungsmacht und Spracherstellung voraus. Man spricht im Falle von maschinellen Agenten von künstlicher allgemeiner Intelligenz (engl. „artifical general intellgence“ (AGI)) - die Informationen werden also intern verarbeitet bzw. verknüpft und Reaktionen in einem scheinbar kreativen Prozess generiert. Die heutige Technik ist noch nicht soweit, eine derart nützliche AGI für offene Domänen, also ohne Beschränkung der Informationsthematik, zu entwickeln.

Letztendlich soll ein (künstlicher) Dialog dem Menschen zur Informationsgewinnung dienen. Modellhaft kann man diese Umgebung als Aktion und Reaktion zwischen zwei Agenten darstellen: d_1 ist typischerweise ein Mensch mit einem Bedürfnis, die von ihm kommende Aktion eine Beschreibung eines Mangels oder einer Erwartung. d_2 versucht diese Anfrage mithilfe einer entsprechenden Reaktion zu erfüllen. Es wird hierbei vom ersten sog. *Turn* t_1 aus einer Folge von Turns $(t_n)_{n \in \mathbb{N}}$ gesprochen. Der zweite Turn t_2 wäre eine Reaktion von d_1 in Form einer Evaluation bspw. einer Bestätigung.

d_2 benötigt Wissen, um nützlich zu reagieren. Dieses Wissen kann in Form

[1] Diese Arbeit behandelt nur den Dialog, also der Austausch zwischen zweier Agenten. Inwiefern man die Erkenntnisse einer Dialogumgebung auf Dialogumgebungen mit einer größeren Anzahl an Agenten ($|D| > 2$) skalieren kann (*Multilog*), ist Gegenstand aktueller Forschung[8].

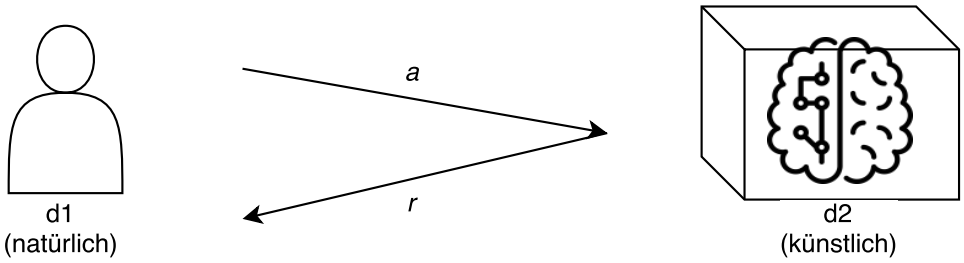

Abbildung 2.1: Atomarer Dialog: Typischer Turn in einer künstlichen Dialogumgebung. Als künstlicher Agent kann d2 hierbei unterschiedlich fortgeschritten sein.

verschiedener Medien dargestellt werden: Die konventionelle ist die unformatierte Schrift. Darüber hinaus werden Informationen formatiert in Tabellen und Listen, als Bilder oder als Weiterleitung auf Webseiten überliefert. Georeferenzen (Ortsbezüge), Bewegtbilder etc. sind weitere Beispiele. Kann dieses Wissen von außerhalb der Umgebung (externes Wissen), also von anderen Menschen (bspw. als Moderator) oder sogar von einer stärkeren KI kommen, so ist d_2 eine Blackbox: Für d_1 ist nicht ersichtlich, wie d_2 zu seiner Reaktion kommt. Internes Wissen wiederum setzt einen Lern-, Trainings- bzw. Einstellungsprozess voraus.

2.2 Dialogelemente

Aktionen und Reaktionen in einem Dialog sind Aussagen $u_{n,n\in\mathbb{N}}$ (engl. *utterances*), also Beschreibungen eines (Welt-)Zustandes und gültig in einem Universum, einer *Domäne*. Vereinfacht wird von einer Frage und der darauf bezogenen Antwort gesprochen. Da Antworten aber verschiedene Formen annehmen können (speziell die rhetorisch *stille* Antwort[2]), ist es passender, von einer abstrahierten Aktion a der Menge $A = \{a_1, a_2, ..., a_n\}_{n\in\mathbb{N}}$ und einer Reaktion r der Menge $R = \{r_1, r_2, ..., r_n\}_{n\in\mathbb{N}}$ zu sprechen. Ein atomarer Dialog ist also ein Turn t_1 zwischen Dialogagent d_1 mit Aktion a_1 und Dialogagent d_2 mit Reaktion r_1. Atomar heißt auch, dass d_2 keine Reaktion von d_1 erwartet, d.h. $r_1 \neq a_2$ und A und R sind disjunkt: es kommt zu keinem t_2.

[2]Die Frage wird zwar verarbeitet, aber auf die zugrundeliegende Erwartung für d_1 nicht sichtbar reagiert

a_n beschreibt also einen Mangel - fehlen darauf folgende r_n, so ist der Turn bzw. der Dialog unvollständig. Daraus folgt ein Grundsatz der nützlichen Dialogumgebung:

> Auf eine Aktion folgt immer eine Reaktion bzw. ein Dialog endet mit einer Reaktion.

Ein Dialog kann auch als Übergänge (t_n) von Zustand u_n zu Zustand u_{n+1} gesehen werden. Diese Ansicht ist wichtig für Automatenmodelle oder probabilistische Modelle im weiteren Kapitel.

Damit entsprechend reagiert werden kann, muss a verstanden werden, d.h. t beinhaltet eine Abbildung von a zu einem r:

$$f_s : A \to R, a \mapsto r$$

In dieser Arbeit wird sie als *semantische Abbildung* f_s bezeichnet. Abhängig vom Modell und der Art seiner Implementierung beinhaltet f_s eine mehr oder weniger anspruchsvolle Verarbeitung von a, um zwischen verschiedenen r diskret und endlich zu unterscheiden. Bei einem atomaren Dialog existiert f_s nur innerhalb eines Turns. Existiert f_s dagegen über mehrere t hinweg oder bildet multivariat mehrere $a_1, a_2, ..., a_n$ auf ein r_n, d.h.

$$f_s(a_1, a_2, ..., a_n) = r_n$$

so kann von einem Dialog mit *erweiterter* Vorgeschichte gesprochen werden und das Problem wird weitaus schwieriger[29, siehe *hierarchischer Kontext*].

Ein Turn t beinhaltet zusätzlich zu dieser Abbildung eine temporale Abhängigkeit zwischen a und r. Sie wird über den Index n definiert und als Zeitstempel, Sortierung etc. implementiert. a_n bzw. r_n mit größerem Index n treten also zeitlich nach $a_{n,*}$ bzw. $r_{n,*}$ mit kleinerem Index n_* auf.

2.2.1 Agenten

In der Praxis unterscheiden sich menschliche und künstliche Agenten u. a. in folgenden Punkten:

- **Die Art und Weise sowie Zeit der Vorbereitung, Einarbeitung, Einstellung, Programmierung etc. vor dem eigentlichen Dialog:** Agenten müssen das notwendige langfristige Wissen für die Verarbeitung der Aktion haben oder zumindest wissen, wo bzw. wie sie es finden können. Das beinhaltet auch, die Sprache zu verstehen, in der die Aktion dargestellt ist. So wird bspw. im Kundendienst das Personal geschult: Ihnen werden Nachschlagewerke beigelegt oder es

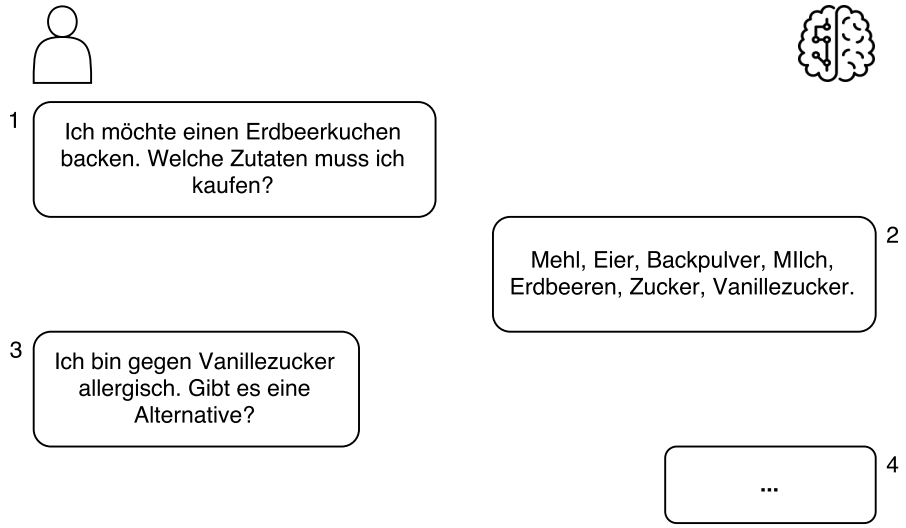

Abbildung 2.2: Erweiterte Vorgeschichte: Der künstliche Agent muss jetzt Bezug auf mehr als nur die unmittelbar letzte Aussage nehmen.

existieren Anlaufstellen für verschiedene Sprachen oder Themen. Maschinen haben den Vorteil, dass sie einmal erworbenes Wissen nicht selbständig vergessen. Die Einstellung sorgt dafür, dass die spätere Aktionsverarbeitung vollständig ist und terminiert. Das betrifft auch den Fall, dass Probleme auftreten, wie fehlende Ressourcen, keine passende Reaktion etc.

- **Die Art und Weise sowie Zeit der Verarbeitung der Aktion bis zu der Ausgabe der Reaktion:** Agenten ordnen jeder Aktion eine Reaktion sinnvoll zu. Das entspricht formal der semantischen Abbildung f_s. Die Verarbeitung schließt eine Überführung der Aktion in eine interne Darstellung ein (sie wird *erkannt* bzw. *interpretiert*). Das können Annotationen bzw. Transformationen, wie syntaktische Stammformreduktion (engl. *stemming*), Rechtschreibkorrektur etc. oder semantische Wortartzuordnung (engl. *part-of-speech tagging*), Oberbegriffzuordnung, Merkmalsvektorisierung etc. sein. Der Agent folgt dabei vorbereiteten, oftmals heuristischen oder stochastischen Regeln.
- **Die Qualität bzw. der Nutzen der Reaktion:** Agenten versuchen das Ziel zu erreichen, sinnvoll zu reagieren. Wie erfolgreich dieses Vorhaben ist, unterliegt nicht nur primär-syntaktischen (stukturellen) Faktoren, sondern auch *sekundär-semantischen* Faktoren, die in der künstlichen Umgebung möglicherweise nur teilweise oder überhaupt nicht beobachtbar bzw. messbar sind. Das können je nach Einsatzgebiet zusätzliche Kontextinformationen wie die (zeitliche) Gültigkeit der Reaktion oder Profil (Kultur, Interessen, Stimmung) der Dialog-

partner sein. Dabei entstehen aufgrund der natürlichen Sprache Mehrdeutigkeiten in der Intention einer Aktion und die Erkennung wird subjektiv abhängig und ontologisch umstritten. Es kann hierbei von einer *Genauigkeit* der Reaktion gesprochen werden, d. h. wie genau erfüllt sie nach eigenem bestem Wissen die Erwartung der Aktion. Die objektive Evaluation dieser Aufgabe ist dennoch Gegenstand aktueller Forschung[17]. Zentrale Teilaufgabe ist es, auch unbekannte Aktionen verarbeiten zu können. Probabilistische Sprachmodelle[29][2] messen dazu die *Perplexität* bzw. die *Kreuzentropie* H_c zwischen der wahren Wahrscheinlichkeitsverteilung $p(a)$ über A und der modellierten Wahrscheinlichkeitsverteilung $q(a)$. Sie sagt aus, wie überrascht über eine unbekannte Aktion a ein Modell ist. Die Kreuzentropie ist definiert als

$$H_c(p, q) = -\sum_x p(a) \log_2 q(a)$$

und die Perplexität als

$$2^{H_c(p,q)}.$$

In der Praxis wird eine Stichprobe N aus p genommen und dadurch die zugrunde liegende Verteilung und damit die Kreuzentropie hypothetisch geschätzt:

$$H_{c*}(p, q) = -\frac{1}{N} \sum_{i=1}^{N} \log_2 q(a_i)$$

Bessere Modelle messen eine niedrigere Kreuzentropie: Sie sind weniger *perplex* über die Aktion und reagieren dadurch vorhersehbarer bzw. *genauer*.

- **Die Anzahl gleichzeitiger Dialoge, die sie führen können (Parallelität):** Maschinen haben den Vorteil, dass sie wesentlich besser skalieren, d.h. statt jeden aktiven Dialog einem Menschen zuzuweisen, behandeln maschinelle Agenten fast beliebig viele Dialoge gleichzeitig. Dagegen sind andere Faktoren in der Verfügbarkeit, den Kosten und der Intelligenz dieser Systeme relevant.

Eine besondere Rolle spielt der moderierende Agent, der zusätzlich zu den aktiven Dialogagenten sichtbar oder unsichtbar in den Dialog greift - er korrigiert bzw. annotiert Aussagen, um den Nutzen des Dialoges für sowohl menschliche als auch künstliche Agenten zu erhöhen. Im Falle von künstlichen Agenten besteht die Möglichkeit, dass der Moderatoragent dadurch die Einstellung der Aktionserkennung während des Dialoges durchführen kann (*on-line*): Der künstliche Agent verändert seinen Aktions- und Reaktionsraum entsprechend den Korrekturen und Hilfestellungen des Moderatoragenten. Im Gegensatz dazu steht die Einstellung vor dem Dialog (*off-line*).

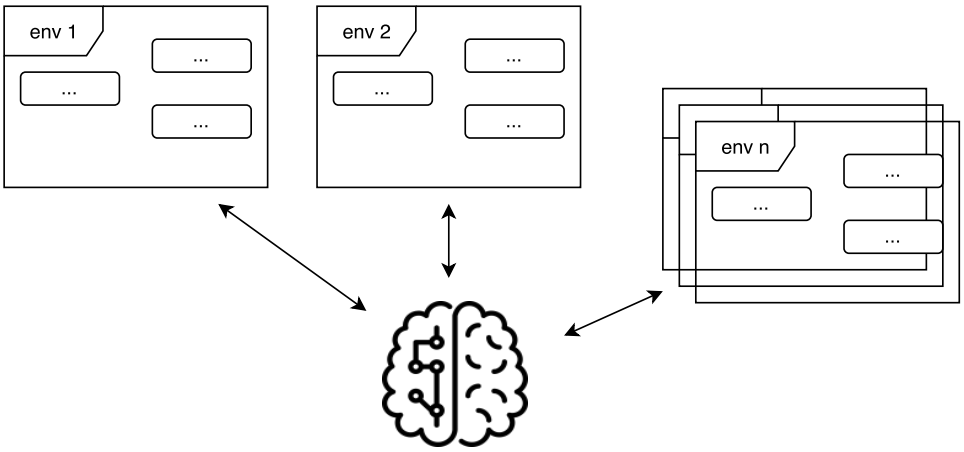

Abbildung 2.3: Parallelisierter Agent: Effektive Führung mehrerer Dialoge gleichzeitig.

Die Funktionsweise und Implementierung der künstlichen Dialogagenten hängen nun von ihrem Modell ab.

2.3 Modelle

Schwache Modelle arbeiten reflexartig. Ihre Verarbeitungsregeln können auf (String-)Vergleiche oder syntaktische Transformationen beschränkt sein und sind nicht erweiterbar. So ist ein Agent auf Basis einer einfachen direkten Tabelle(-nsuche) ein schwaches Modell. Die Reaktionen sind mit keinem oder geringem Aufwand vorhersehbar und entsprechen im Kundendienst einem unausgebildeten Mitarbeiter, der ohne inhaltliches Hinterfragen seine Reaktionen aus einem Regelwerk übernimmt - er besitzt keine eigene Wahl. Probleme treten auf, sobald unbekannte Aktionen auftreten oder Fehler passieren. Verändert werden schwache Modelle anhand der Elemente und Mächtigkeit der Aktions- und Reaktionsmenge A und R bzw. $|A|$ und $|R|$: Es wird die dem Agenten verfügbare Datenmenge vergrößert oder verkleinert.

Starke Modelle arbeiten kognitiv. Sie besitzen Logiken für Sprachverständnis, Argumentation, Entscheidungsmacht und Spracherstellung. Diese Logiken arbeiten geschlossen und sind nur mit relativ viel Aufwand vorhersehbar: Es erscheint der Eindruck von einem eigenständigen *intelligenten* Denkprozess[27]. So können diese Systeme bspw. bekannten Aussagen Wahrheitswerte[3] zuweisen und aus ihnen neue gültige Aussagen schlussfolgern, sofern sie

[3]Das können binäre Werte für *wahr* und *falsch* sein, oder auch numerische Werte ($\mathbb{R}$) zwischen 0 und 1, wie z.B. 0.8943.

entscheidbar sind. Sie besitzen dadurch ein Weltbild mit eigenen Überzeugungen, sog. *belief states*. Diese Überzeugungen werden mithilfe einer **Wissenskomponente** und einer **Inferenzkomponente** begründet. Die Wissenskomponente beinhaltet eine Wissensbank (engl. *knowledge base (KB)*) mit Objekten, Relationen und Prädikaten, mit denen in der Inferenzmaschine theoretisch auf neues Wissen deduziert wird[4]. Dabei wendet sie feste Schlussfolgerungs- und Integritätsregeln an - eine Untermenge der Auswahlregeln für Reaktionen bei Dialogagenten. Die Kompetenz dieser Systeme hängt also von der Entscheidbarkeit neuer Aussagen ab. Wie anfangs erwähnt, ist die heutige Technik aber noch nicht imstande, ein allgemein zufriedenstellendes kognitives System zu entwickeln. Es existieren aber schon seit den 1980er Jahren Expertensysteme[20], die in immer weniger eingegrenzten Arbeitsgebieten Menschen assistieren.

Diese zwei extremen Agentenmodelle verstehen sich nicht als harte Kategorisierung, sondern als fließender Übergang abhängig von der Anzahl und Güte der Auswahlregeln und dadurch von der Verarbeitung der Aktion und der Implementierung der semantischen Abbildung f_s. So wie historisch bei Expertensystemen ein Übergang von klassischen Datenbanken (DB) zu Wissensbanken (KB) stattfand[3], so könnte in Zukunft ein weicher Übergang von schwachen zu starken Modellen stattfinden.

Agentenmodelle können nach der Art und Weise, wie sie Reaktionen bilden, unterschieden werden: Entweder sind sie **abfragend** (wiederholend) oder **generativ**.

2.4 Abfragende Modelle

Abfragende Modelle (engl. *retrieval-based models*) bedienen sich (manuell) eingestellter Reaktionen, d. h. sie antworten mit dem, womit vorher schon geantwortet worden ist - im Falle von Text wiederholen sie also im Wortlaut. Jedes Element r aus R ist nicht aus kleineren Bauteilen zusammengesetzt: Zeichen, Satzteile oder Wortbausteine existieren für den abfragenden Agenten nicht. r ist als solches atomar.

Das direkte Tabellenmodell

Das einfachste abfragende Modell ist das direkte Tabellenmodell. Dabei wird eine $1:1$ injektive Abbildung von einer Aktion auf eine Reaktion ähnlich einer Tabellenstruktur genutzt. Die Auswahl der Reaktion geschieht im naiven Fall mithilfe einer unverarbeiteten Aktion und einem kompromisslosen

[4] [3] spricht auch von *semantischen Abfragen*.

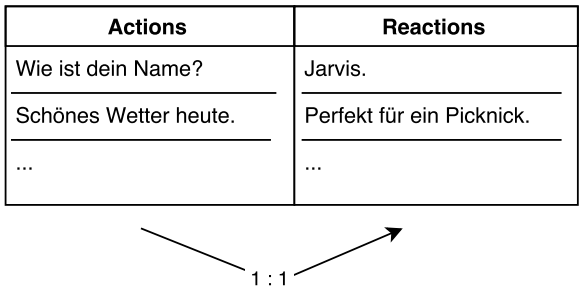

Abbildung 2.4: Das direkte Tabellenmodell: Die semantische Abbildung beschränkt sich exklusiv auf eine Aktion.

Vergleich: Nur wenn die zu vergleichende Aktion mit einer bekannten Aktion in der Tabelle exakt übereinstimmt, wird die Reaktion ausgewählt (*exact match*). Wenn nicht, kann das Modell bspw. eine generische Reaktion für unbekannte Aktionen besitzen. Demnach hängt die Genauigkeit des Modells vom Einfügen genügender (neuer) hart-kodierter Wertepaare (a, r) ab. Sie kann zusätzlich durch entsprechende Metadaten m nach sekundären Auswahlfaktoren spezifiert werden wie bspw. Alter und Geschlecht des Dialogpartners[5], das aber einen beträchtlichen Mehraufwand mit sich zieht: Die Erkennung verkommt zu einer verschachtelten Fallunterscheidung mit Wertetripeln (a, m, r). Die Erkennung entspricht in seiner Reinform einer linearen Suche in A, d. h. sie kann in der Implementierung mit bekannten Mitteln wie Indizierung, Sortierung, Hashing etc. optimiert werden. Das Modell ist impraktikabel bei Dialogen mit Freitext, da es bei jeder unbekannten Aktion scheitert. Der Austausch wirkt starr, das Modell kann aber praktisch im Falle von vordefinierten Dialogen sein (*frequently-asked-questions* oder *multiple-choice*).

Das Pattern-Matching Modell

Der künstliche Agent erhält das Wissen um korrekte Reaktionen auf mehr als eine Aktion, d. h. r ist das Ziel einer $n : 1$ Abbildung einer echten Untermenge U_a von A. Die Erkennung entspricht einer Schablone, die Aktionen rein syntaktisch filtert. Die Genauigkeit wird im Vergleich zu dem direkten Tabellenmodell erhöht: Auf weniger (unbekannte) Aktionen muss generisch verweisend reagiert werden. Die Einstellung bedient sich formalen Sprachen wie *Prolog*[28], *Regex* (reguläre Ausdrücke) oder dem XML-Dialekt *AIML*[6][37] und setzt eine genaue Beschreibung von U_a voraus. So ist es nicht möglich, einzelne Fälle aus der Abbildung auszuschließen, ohne die Fallun-

[5] Denkbar um eine geeignete Anrede zu finden.
[6] Prominentestes Beispiel ist der Chatbot ELIZA.

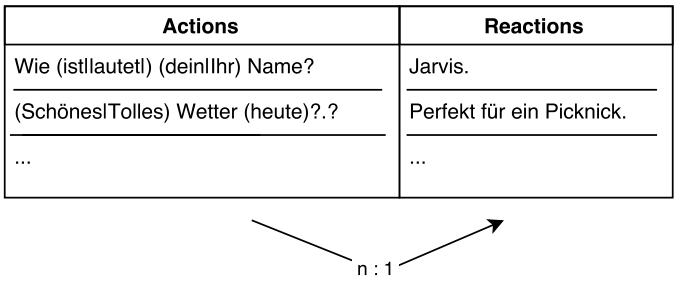

Abbildung 2.5: Das Pattern-Matching-Modell: Jede Einstellung (hier als Zeile) bedient mehrere mögliche Aktionen.

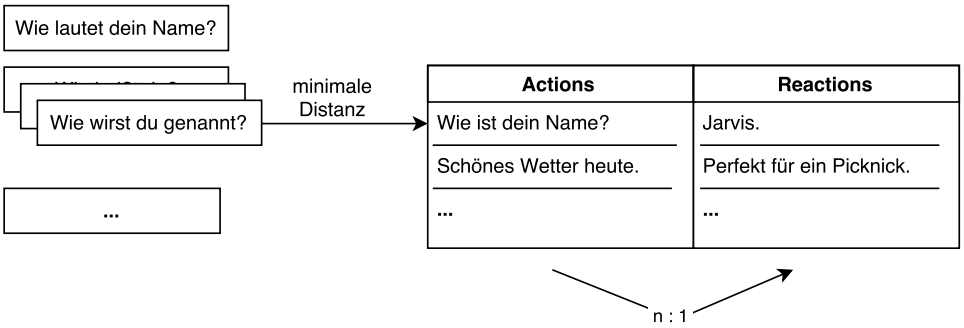

Abbildung 2.6: Wortgruppen.

terscheidung zu teilen oder zu verschachteln. Die Einstellung der Erkennung wird dadurch aufwändig, da sie nur manuell und off-line durchgeführt werden kann.

2.4.1 Wortgruppenmodelle (Ähnlichkeitsmodelle)

Der künstliche Agent erhält das Wissen um das Konzept von Ähnlichkeit(en) zwischen Wörtern, selbst wiederum sinngebend zusammengesetzt aus Zeichen. Das entspricht der Kategorisierung von Wörtern in Gruppen oder der (mehrdimensionalen) Sortierung in Clustern. Wichtig für die Erkennung ist dabei, dass die Ähnlichkeit messbar ist, also es existiert eine Metrik bzw. eine Distanz zwischen einzelnen Wörtern. Durch die Anwendung auf eine (Linear-)Kombination von Wörtern bzw. internen Wortdarstellungen ist es so möglich, Reaktionen auf Untermengen U_a ähnlicher Aktionen abzubilden, um einen größeren Aktionsraum abzudecken, ohne explizit einzeln alle diskreten Werte a angeben zu müssen. Dazu wird immer der bzgl. der definierten Metrik **nächste** Aktionswert erkannt (*fuzzy-matching* mit *nearest neighbor search (NNS)*). In der Praxis besitzt die Erkennung ähnlicher Ak-

tionen eine Toleranz , ab der die zugehörigen Reaktionen dennoch verworfen werden. Ansonsten ist die semantische Abbildung f_a stetig.

2.4.2 Modelle mit syntaktischer Distanz

Modelle mit syntaktischer Distanz (engl. *edit distance*) δ_e messen die **morphologischen** Transformationen M, um von einem Wort w_a zu einem anderen Wort w_b zu kommen. So benötigt etwa w_1 = 'gehe' nur ein zusätzliches Zeichen 'n', um auf w_2 = 'gehen' zu kommen, also $\Delta_M(w_1, w_2) = \{Einfügen('n')\}$. Eine solche Messung ist die **gewichtete Damerau-Levenshtein-Distanz**. Sie ist definiert durch die minimale Anzahl an Transformationen einzelner Zeichen zweier anzugleichender Worte. Dabei erlaubt es nur die Transformationen

$$M_{lev} = \{Einfügen(c),\ Löschen(c),\ Ersetzen(c),\ Tauschen(c)^7\}$$

für einzelne Zeichen c. Jede Transformation wird gewichtet bzw. erhält einen Kostenwert v - die gemessene Distanz δ_e zweier Worte ist dann die Summe der Transformationskosten:

$$\delta_e = V_{lev} = \sum_{m=1}^{|M|} v_m.$$

Die Damerau-Levenshtein-Distanz kann als Erweiterung der **Hamming-Distanz** verstanden werden, bei der die anzugleichenden Worte dieselbe Länge besitzen ($|w_1| = |w_2|$) und nur $M_{ham} = \{Ersetzen(c)\}$ erlaubt ist. Modelle mit syntaktischer Distanz sind zur Erkennung von Aktionen mit Rechtschreibfehlern gut geeignet[19].

2.4.3 Modelle mit semantischer Distanz

Modelle mit semantischer Distanz versuchen die Ähnlichkeit einer größtmöglichen Menge an Merkmalen der **Wortbedeutung** zu messen. Zentral ist dabei die Definition bzw. Extraktion dieser Merkmale und die Erkennung von Synonymen (engl. *synsets*), die größtenteils während der Überführung in die interne Darstellung stattfindet. Das sind (im Gegensatz zu den syntaktischen Modellen) **konzeptionelle** Transformationen bzw. Annotationen K, wie die Vektorisierung im Merkmalsraum, die Zuordnung zu grammatikalischen Einheiten (*POS tagging*) oder die Feststellung benannter Entitäten (engl. *named entities recognition*). In diesem Sinne sind die Merkmale also Metadaten, die in die Erkennung zusätzlich einspielen. Dabei wird zwischen manuell eingetragenen Merkmalen und trainierten bzw. konditionierten Merkmalen unterschieden.

[7] c muss dabei direkt benachbart sein, z. B. 'haus' zu 'huas'.

Das Part-of-Speech Modell

Um die semantische Distanz zu veranschaulichen, kann das Part-of-Speech Modell betrachtet werden. Dabei werden die einzelnen Wörter der Aktion mit ihrer grammatikalischen Wortgruppe, wie Substantiv, Verb, Adjektiv, Adverb, etc. notiert[8] und anhand der benötigten Anzahl an Wortarttransformationen $|K(a_1, a_2)|$ von Aktion a_1 zu Aktion a_2 verglichen. So liegt zwischen den Sätzen

'Wie	ist	dein	Name	?'
(Pronomen,Frage)	(Verb)	**(Pronomen,Besitz)**	**(Nomen)**	(Punktuation)
'Wie	wirst	du	genannt	?'
(Pronomen,Frage)	(Verb)	**(Pronomen,Personal)**	**(Verb)**	(Punktuation)

eine einfache Wortart-Distanz von 2. Dabei ist die einzig erlaubte Transformation

$$K_{pos} = \{ErsetzenDurchGruppe(c)\}.$$

Diese Erkennung ist erst auf Aktionsebene statt auf Wort- oder Zeichenebene sinnvoll, da das Konzept der Grammatik auf semantisch reichere ganze Sätze bildet. k unterscheidet sich deswegen von der morphologischen Transformation der syntaktischen Hamming-Distanz dadurch, dass sie auf ganze sinngebende Wörter angewendet wird und c in diesem Fall eine beliebige Wortgruppe aus

$$C_{pos} = \{c \mid c \textit{ ist grammatikalische Wortgruppe oder leer}\}$$

ist. Wortgruppen können zusätzlich unterschiedlich gewichtet werden, sodass bspw. $k(Substantiv)$ und $k(leer)$ (entspricht dem Löschen des Wortes) stärker in die Erkennung spielen als $k(Punktuation)$ und statt $|K|$ dann die Transformationskosten V_{pos} verglichen werden. Das Modell versucht also das Erkennungsproblem der größten grammatikalischen Überlappung zwischen Aktionen zu lösen - semantisches Konzept bzw. Merkmal ist dabei die zugrundeliegende Grammatik. Die interne Darstellung entspricht Tupel dieser semantischen Wortgruppen. Je nach Kombination und Reihenfolge reagiert der künstliche Agent mit einer mehr oder weniger sinnvollen Antwort, so kann bspw. auf das Tupel $(Substantiv, Verb)$ mit der nominativen Aufforderung

'Erzähl mir mehr über ihr/sein Vorhaben.'

oder auf das Tupel $(Substantiv)$ mit dem Kommentar

[8] Es wird in dieser Arbeit nicht tiefer auf das (automatische) *POS Tagging* eingegangen. Unter dem Stichwort findet sich aber genügend Literatur, um seine Funktionsweise vor Augen zu führen. Andere verwendete grammatikalische Wortgruppen bzw. *Tags* können sein: Konjunktion, Partikel, Artikel, Punktuation, Nummer, Adposition, Pronomen.

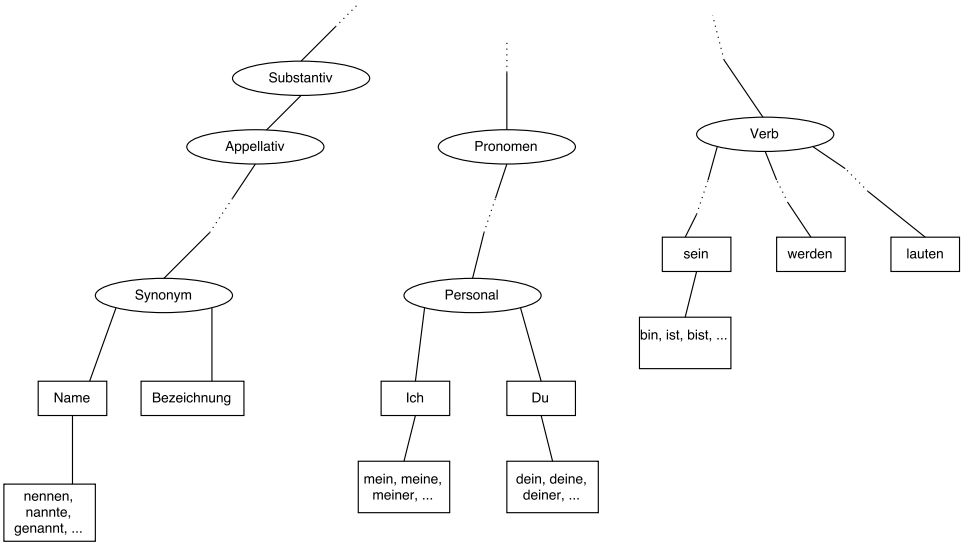

Abbildung 2.7: Klassifikationsbaum für die Distanz zwischen Wörtern. Konjugationen werden hier zu einem einzelnen Blatt zusammengefasst.

'Das ist schön. Was willst du damit tun?'

reagiert werden. Der Austausch wirkt trotz Nichtexistenz einer maximalen Tupellänge aufgrund der geringeren Anzahl an Wortgruppen und der dadurch geringen Anzahl an möglichen Reaktionen sehr oberflächlich, dennoch ist ein sinngebender Ansatz marginal erkennbar. Die semantische Distanz ist also ähnlich der syntaktischen Distanz:

$$\delta_s = V_{pos} = \sum_{k=1}^{|K|} v_k$$

Um das Modell nun zu skalieren, werden die grammatikalischen Einheiten in immer feinere semantische Kategorien aufgeteilt, bis schließlich als direkte Obergruppe nur noch das Synonym ein oder mehrere Wörter einschließt. Es entsteht eine klassifizierende Hierarchie bzw. eine Relation in Form einer klassifizierenden, hierarchischen Ordnung zwischen den Wortgruppen (*Taxonomie*) und die einzige Transformation wird zu

$$K_{pos,*} = \{k = AbstrahierenOderSpezifizieren(c)\}.$$

Die semantische Distanz kann dann auch vereinfacht als kleinste Anzahl durchquerter Knoten oder Kanten dieses semantisch eindimensional geordneten bzw. sortierten *Klassifikationsbaumes* von w_1 zu w_2 verstanden werden.

Das WordNet Modell

Das WordNet[7][23] ist solch ein semantisches Netz in Form einer strukturierten, lexikalischen Datenbank. Sie gruppiert Synonyme der englischen Sprache zu *Synsets* und verbindet sie u. a. mit folgenden, teilweise gerichteten und optional gewichteten Relationen:

- Substantive
  - Synonymie
  - Hyponymie (Unterordnung)
  - Hyperonymie (Überordnung)
  - Meronymie (Teil-aus)
  - Holonymie (Ganzes-aus)
- Verben
  - Hyperonymie (Überordnung)
  - Implikation (Konsequenz)
  - Troponymie (implizierte Unterordnung)

Abgesehen von der Messung der Ähnlichkeit δ_s anhand des kürzesten Pfades als der Anzahl an durchquerten Knoten $length_p$ zwischen w_1 und w_2 geteilt durch die maximale Taxonomietiefe[9] $depth_T$[16], also

$$\delta_{LCH} = -log(length_p/(2 * depth_T)),$$

kann als δ_s im WordNet auch die Pfadtiefe von einer imaginären Wurzel zum nächsten gemeinsamen hypernomischen Knoten w_{lcs} (engl. *Least Common Subsumer* (LCS)) zwischen w_1 und w_2 berechnet werden[39]:

$$\delta_{WUP} = 2 * depth_{lcs}/(depth_{w_1} + depth_{w_2})$$

Beide Messungen[26] liefern dabei einen normalisierten Ähnlichkeitswert zwischen 0 und 1.

Das (vereinfachte) Lesk-Modell

Mithilfe von WordNet, das auch als Wörterbuch dienen kann, d. h. zu jedem Knoten wird eine Definition geliefert, kann das Lesk-Modell implementiert werden. Im vereinfachten Algorithmus[1] wird dazu die Anzahl gleicher

[9] Tiefe ($depth_w$) eines Wortes w heißt die Länge des Pfades von einer Wurzel bis zum Wortknoten w, der die hierarchische Ordnung ausschließlich herabsteigt.

Wörter aus den Definitionen zu w_1 und w_2 gezählt, also basierend auf überlappende Definitionstexte Q_{w_1} und Q_{w_2} mit

$$Q_w = \{q \mid q\ ist\ Wort\ aus\ Definition\ von\ w\}$$

eine semantische Distanz definiert mit

$$\delta_{Lesk} = |Q_{w_1} \cap Q_{w_2}|.$$

Probabilistische Modelle

Bisherige Modelle ignorieren semantisch abhängige Wortsequenzen

$$(w_1, w_2, \ldots, w_n),$$

die in dieser zusammengesetzten Struktur eine andere Bedeutung haben können als jedes einzeln betrachtete Wort und δ_s deshalb nicht optimal gemessen werden kann. Ein Ansatz zur Lösung dieses Problems ist es, die Bedeutung eines Wortes für die umschließende Aussage aus gerade diesem Umfeld zu schließen: Die n Wortnachbarn eines Wortes w definieren die Wortgruppe C_w für w. Dabei sind die anderen Elemente in C_w Wörter w_*, die anstelle von w an derselben Position in der Aussage stehen könnten, ohne die Bedeutung zu sehr zu verändern - sie sind semantisch ähnlich, weil sie in gleichem oder ähnlichem Kontext benutzt werden. Das eigentliche Wort und die manuellen Konzepte bzw. Merkmale (wie bisher Grammatik oder Taxonomie) werden automatisch in einen hochdimensionalen Merkmalsraum erfasst. Da äquivalent auch von der Vorhersage p eines Wortes bzw. $p_1, \ldots, p_n$ ähnlicher Wörter aus dem Kontext gesprochen kann, folgen probabilistische Modelle, die diese Merkmale statistisch trainieren bzw. konditionieren.

Das N-Gramm-Modell

Das N-Gramm-Modell schätzt die konditionalen Wahrscheinlichkeiten p (engl. *likelihoods*) für Wörter W einer Sprache L, dass sie auf eine Folge von Wörtern derselben Sprache auftreten, d. h. aus Sequenzen der Länge n von linksseitigem Subkontext bzw. Historie

$$H_{w_n} \overset{10}{=} w_1 \circ \cdots \circ w_{n-1} = w_1, \ldots, w_{n-1}$$

und dem aufgetretenen Wort w_n selbst (sog. *N-Gramme*[4]) werden Wörter vorhergesagt, die anstelle des aufgetretenen Wortes in L benutzt werden.

[10] $\circ$ bezeichnet die Konkatenation zweier Wörter bzw. Strings.

1-Gramm (Unigramm)

wie	lautet	dein	Name

2-Gramm (Bigramm)

<pre1>[11] wie	wie lautet	lautet dein	dein Name

3-Gramm (Trigramm) mit Vorhersage

<pre2> <pre1> **wie**	<pre1> wie **lautet**	wie lautet **dein**	lautet dein **Name**
(0.3534) wie	(0.3977) heißt	(0.2759) das	(0.1534) einziger
(0.2193) warum	(0.1573) geht	(0.2634) dein	(0.1193) Name
(0.1562) ich	(0.1222) wirst	(0.1642) der	(0.1562) voller
(0.1939) das	(0.1986) hat	(0.1312) nur	(0.0939) Partner

Abbildung 2.8: Beispiel N-Gramme: Aus den Trigrammen werden austauschbare (und somit ähnliche) Wörter vorhergesagt, die zusammen eine semantische Wortgruppe bilden.

Alle folgenden Wörter werden für die Vorhersage also nicht mehr hinzugezogen. Zentral ist die heuristische Markow-Annahme, dass es ausreicht nur eine durch $n-1$ begrenzte Historie zu betrachten, um das Folgewort w_n vorhersagen zu können: Für das Modell sind zwei Historien gleich, wenn sie auf dieselben $n-1$ Wörter enden[4]. Bei $n=5$ ist also

$$p_{n=9}(Bayern|Franz, jagt, im, komplett, verwahrlosten, Taxi, quer, durch)$$
$$= p_{n=5}(Bayern|verwahrlosten, Taxi, quer, durch)$$

und für generelle n

$$p(w_t|w_1^{t-1}) = p(w_t|w_{t-n+1}^{t-1}). \tag{2.1}$$

Dabei ist w_1^{t-1} die Wortfolge von w_1 bis w_{t-1}. Da es oftmals ein untragbarer Aufwand wäre, alle möglichen Wortfolgen einer Sprache zu erfassen, wird p nur aus einem möglichst großen Beispieltext B (eine Stichprobe der Verteilung L) mithilfe der *Maximum-Likelihood-Methode* (ML) statistisch geschätzt bzw. trainiert: Es werden p als ML-Schätzer so angenommen, dass die Verteilung innerhalb des Beispieltextes auch gleichzeitig die wahrscheinlichste Verteilung für die Sprache ist. In einem 1-Gramm-Modell (Unigramm-Modell) ist das für ein Wort w seine Frequenz

$$p_L(w) \approx \frac{C(w)}{B}$$

[11] Für die Vorhersage von Wörtern am Satzanfang wird ein spezieller <preN>-Token eingeführt und trainiert.

in B mit $C(w)$ als Anzahl der Vorkommen von w in B, bei N-Gramm ist es dann

$$p_L(w_n|w_1^{n-1}) \approx \frac{C(w_1^{n-1}w_n)}{\sum_w w_1^{n-1}w}. \tag{2.2}$$

Das N-Gramm Modell unterliegt letztendlich zwei Annahmen: Der Markow'schen Annahme und der ML-Annahme, die beide auf die Repräsentationsdichte von Untermengen vertrauen. Je größer n also ist, desto genauer wird das statistische Modell für die wahre Verteilung L, aber dafür aufwendiger die Beschaffung und Verarbeitung von repräsentativen Trainingsdaten, die groß genug sind. In der Praxis ist der Kompromiss derart effizient und die Anwendung einfach, dass das Modell u. a. in der genetische Sequenzanalyse, der Datenkompression und der Sprachverarbeitung seit langer Zeit erfolgreich genutzt wird.

Das N-Gramm-Modell bildet nun eine heuristische Unterfunktion für die semantische Abbildung in künstlichen Dialogagenten. Die Historie von w definiert eine einfache semantische Ähnlichkeit zwischen w und anderen w_* bezüglich des Kontextes und damit Wortgruppen mit den k dadurch ähnlichsten Wörtern. Eine Möglichkeit wäre, eine unbekannte Aktion a_u mit seinen Wörtern U in überlappende N-Gramme zu unterteilen und für jedes N-Gramm nach (2.2) die konditionalen Wahrscheinlichkeiten

$$P_v = \{p(v_1|H_u^n), \ldots, p(v_n|H_u^n)\}$$

für jedes Wort v aus dem trainierten Vokabular V zu berechnen. Die zu vergleichende Aktion a mit seinen Wörtern[12] W erhält gleichzeitig damit für jedes H_u^n eine Wahrscheinlichkeit q_u:

$$q_u(a|H_u^n) = \begin{cases} 1 & \text{falls } \exists w \in W : u = w \\ \hat{p} = max(P_{w \cap v}) & \text{sonst} \end{cases}.$$

Vereinfacht wird bisher für jedes H_u^n von a_u, gemäß den gewonnenen Erfahrungen des Trainings, eine semantische Entsprechung gesucht und bewertet. Die semantische Distanz δ_s kann dann als Durchschnitt aller dieser q_u verstanden werden:

$$\delta_s(a, a_u) = \frac{\sum_{u=1}^{|U|} q_u}{|U|}.$$

Probleme bereiten dem N-Gramm-Modell grobe Verletzungen der zugrundeliegenden Annahmen sowie schlecht angepasste Parameter oder unbekannte (oder falsch geschriebene) Wörter (engl. *out-of-vocabulary* OOV)[4].

[12] Die Reihenfolge der Aktionswörter U bzw. W wird ignoriert (engl. *bag-of-words* BOW) und idealerweise ist sowohl $U \subseteq V$ als auch $W \subseteq V$, d. h. alle Wörter der beiden Aktionen sind im Vokabular V des künstlichen Agenten enthalten. Ist das nicht der Fall, müssen diese unbekannten Wörter gesondert behandelt werden.

Eine weitere Anwendung von N-Grammen ist die syntaktische Klassifizierung von Text im **Vektorraum-Modell** $\mathbb{R}^{|M|}$ (lokales Modell) - eine reine Indexierung nach den Merkmalen M, die durch diskrete N-Gramme dargestellt werden können. In unserem Anwendungsfall ist jedes mögliche N-Gramm eine Dimension und jedes Wort ein Vektor im Raum, wiederum trainiert anhand eines Beispieltextes. Ein wesentlicher Vorteil dieses Modells ist die linear-algebraische Verarbeitung und Visualisierung syntaktischer Distanzen δ_s. Berechnet wird sie mit der Kosinus-Ähnlichkeit: Der Kosinus des Winkels zwischen zwei Wortvektoren hat bei ähnlicheren Wörtern mit ähnlicher Richtung einen Wert näher an 1 - sie liegen syntaktisch intuitiv näher beieinander. Das ist für den künstlichen Agenten nützlich für die Erkennung im Sinne einer *lexikalischen* Suche, ignoriert aber die bisher erarbeitete semantische Ähnlichkeit bezüglich des Wortkontextes und hinzu kommt der exponentielle Anstieg des Raumvolumens durch jede neue, möglicherweise aussageschwache Dimension[13] (sog. Fluch der Dimensionalität, engl. *curse-of-dimensionality* COD). Der Unterschied zwischen extremen Distanzen wird immer kleiner im Vergleich zur kleinsten Distanz; dadurch wird die Metrik zur Messung von Ähnlichkeiten im Raum immer ungenauer und unbrauchbarer. Kombinieren wir daher die Vektorraumdarstellung und die effektive Kosinus-Ähnlichkeit mit der Merkmalsextraktion zur semantischen Vorhersage und zur Reduzierung der Dimensionszahl, entsteht das verteilte Modell[14].

Das verteilte Modell (globales Modell)

Das verteilte Modell (auch Worteinbettungen, engl. *word embeddings*[21][22] genannt) ist ein Vektorraummodell, das global und stetig abhängige Merkmale extrahiert, d. h. jeder Satz im Training optimiert exponentiell über alle Merkmale und deren gegenseitige Interferenz hinweg[2]. Dadurch ergeben sich im Grunde drei zusammenhängende Unterschiede zur N-Gramm-Extraktion:

Weniger Dimensionen: Da die Merkmale kohärenter sind, braucht es nun weniger Dimensionen um eine semantische Ähnlichkeit aus dem Beispieltext zu modellieren, da relevantere Informationen in die lineare Struktur kombiniert werden. Dadurch sinken die negativen Folgen des COD, weil der Raum

[13] z.B. sind Stopwörter überall häufig vorkommende Worte, deren Informationsgehälter dadurch sinken, die Vektoren bzw. Vektordistanzen aber gleichwertig beeinflussen. Eine Maßnahme gegen diese ungewollte Interferenz ist das Maß **TF-IDF**, der aus der Vorkommenshäufigkeit und zusätzlich der inversen Dokumenthäufigkeit eine Gewichtung der Merkmale und eine Transformation im Raum erzwingt.

[14] Eine andere Möglichkeit wäre das LSA (Latent Semantic Analysis), ein Verfahren zur Reduzierung der Dimensionen und Behandlung von Synonymen durch Singulärwertzerlegung.

vom Training dichter besetzt wird.

Realistischere Metrik: Der Raum ist kompakter und die Abstände aussagekräftiger. Das führt zur besseren Abbildung der Abstände auf die semantische Ähnlichkeit, weil der gesamte Verlauf H_u vom Anfang des Satzes bis zum charakterisierten Wort u die Dimensionen definiert.

Weichere Erkennung: Das Training ist auflösender, die Ergebnisse feiner. Kleine Änderungen in den Merkmalen bewirken nun auch nur kleine Änderungen in der Wahrscheinlichkeit. Dabei fließt nicht nur der wörtliche Verlauf in das Training, sondern auch bereits bekannte, dem Verlauf ähnliche Wörter, d. h.

Franz jagt im komplett verwahrlosten Taxi quer durch Bayern

trainiert gleichzeitig auch mit Sätzen, die möglicherweise nicht im Korpus vorkommen, wie

Bernd jagt im total verwahrlosten Bus quer durch Baden.

Anton jagt im ganz verwahrlosten Auto quer durch Hessen.

...

Um das Modell anhand eines (großen) Beispieltextes zu trainieren, werden die Wahrscheinlichkeiten nicht mehr wie bisher durch das Zählen der Vorkommen berechnet[15], sondern durch eine mehr oder weniger tiefe, gewichtete Verkettung von Schichten neuronaler Aktivierungsfunktionen (Künstliches Neuronales Netz KNN, engl. *artifical neural net*) approximiert. Durch die iterative Anpassung der Gewichte wird das Problem auf eine Optimierung der semantischen Abbildung als Komposition differenzierbarer Funktionen reduziert. Das Ziel dabei ist bekanntermaßen, aus einem großen und endlichen Trainingskorpus eine nützliche Vorhersage aus dem vorangegangenen Verlauf

$$p(w_t|w_1^{t-1}) \overset{KNN}{=} f(w_t, \ldots, w_{t-n+1})$$

zu erhalten[16].

Das Schichtenmodell nach Bengio et al. zeigt zwei Komponenten pro Satz bzw. Trainingsset $w_1 \ldots w_T$ mit Vokabular V:

1. **Die als Unterfunktion angenäherte und weiterverwendbare Einbettung der Wörter im Raum $\mathbb{R}^{|m|}$ als Matrix $C = |V| \times m$ freier Parameter.** Sie wird schrittweise beim Übergang von der

[15] Diese frequenz-basierten Verfahren funktionieren hauptsächlich unter der Annahme, dass Wörter mit ähnlicher Semantik ähnliche Verteilungen in der Sprache besitzen.

[16] Vgl. dazu die N-Gramm Vorhersage (2.1)

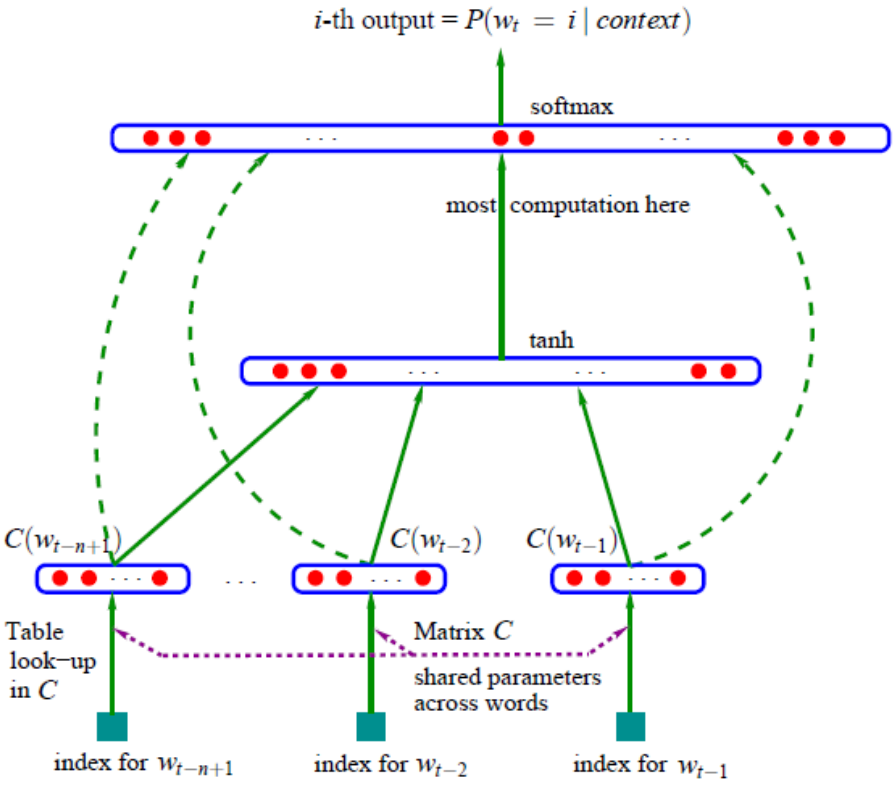

Abbildung 2.9: Neuronales Schichtenmodell nach Bengio et al.[40]

Eingangsschicht zur versteckten (engl. *hidden*) Aktivierungsschicht[17] optimiert als Variable für die Aktivierungsfunktionen. m ist dabei eine vorher festgelegte Konstante für die Anzahl extrahierter Merkmale, bspw. zwischen 50 und 200.

2. **Die Wahrscheinlichkeitsfunktion bzw. Vorhersage p aus einem Subkontext H_t als Summe der künstlichen Neuronen mit skalierter logistischer Aktivierungsfunktion *tanh*[18].** Der direkte Zwischenoutput q der Aktivierungsschicht

$$q = b + Wx + U \cdot \tanh(d + Hx)$$

b: Schwellenwert-Neuronen (Bias) zum Verschieben des Outputs q

d: Schwellenwert-Neuronen zum Verschieben des (versteckten) Aktivierungsoutputs aus *tanh*

[17] Die Aktivierungsschicht kann wiederum aus mehreren zusammenhängenden Teilschichten bestehen. Sie ist der Kern des neuronalen Netzes und beliebig komplex. Die neuronale Architektur bestimmt daher die Topolgie der Schicht.

[18] Tangens Hyperbolicus

W: Gewichte von der Eingangsschicht zu der Ausgangsschicht; überspringt Aktivierung (falls C direkt übergeben wird, ansonsten $W = 0$)
U: Gewichte von der versteckten Schicht zu der Ausgangsschicht
H: Gewichte an der versteckten Schicht
x (Aktivierungsinput): Satzvektor als Konkatenation der jeweiligen Wortvektoren (Merkmalsvektoren) aus den Zeilen der Matrix C

liegt typischerweise logarithmiert[19] vor und wird beim Übergang zur Ausgangsschicht schrittweise maschinell optimiert durch das stochastische Gradientenverfahren (Abstiegsverfahren, engl. *stochastic gradient descent* SGD)

$$\theta \leftarrow +\epsilon \frac{\partial log\ p}{\partial \theta}$$

mit der Lernrate ϵ an den zu optimierenden (freien) Variablen

$$\theta = (b, d, W, U, H, C).$$

Dort findet schließlich eine Softmax-Normalisierung von q zur gesuchten Wahrscheinlichkeit p auf das Intervall $[0, 1]^{|R|}$ statt mit

$$p = \frac{e^{q_t}}{\sum_{i}^{|V|} e^{q_i}}.$$

Sind nur die Worteinbettungen interessant, kann p nach dem vollständigen Training verworfen werden; für die einzelnen Trainingsschritte werden sie jedoch benötigt (überwachtes Lernen) und müssen daher berechnet werden. In den Berechnungen der versteckten Schicht und in der anschließenden Normalisierung liegt auch der Flaschenhals des Modells[22]. Beim Softmax etwa wird zu jedem Schritt durch die aktuellen Log-Wahrscheinlichkeiten **aller** Wörter in V geteilt. Verbessert wird das Modell daher zunächst über Maßnahmen an der Berechnung des Softmax, bspw. kann der Zugriff auf die Werte über eine Baumstruktur geregelt werden (hierarchischer Softmax mit binärem Huffman-Baum), die die Komplexität von O(V) auf O(log V) senken kann. Eine andere Methode wäre Negatives Sampling[22], das statt mit allen Worten nur mit einer wesentlich kleineren Untermenge aus einem Wort des Trainingssatzes und sonst zufällige, nicht enthaltene Worte normalisiert.

Dennoch bleibt die Einschränkung durch das zeitintensive Training bzw. die hohe Rechenlaufzeit in der (ersten) versteckten, auf dem nichtlinearen

[19] Das hat praktische Gründe für die maschinelle Berechnung; so ist die Addition effizienter und bei sehr kleinen Wahrscheinlichkeiten der Logarithmus numerisch stabiler.

Tangens Hyperbolicus basierten Projektion. Daher wird entgegen den Vorzügen einer (tiefen) neuronalen Architektur auf Kosten der Genauigkeit diese Schicht ausgelassen. Es entsteht ein flaches verteiltes Modell (simples Modell), das abhängig von der Effizienz des Softmax mit wesentlich mehr Daten trainiert werden kann[22]. Wie zuvor erwähnt, wird dadurch das Modell auf Worteinbettungen konzentiert und die Vorhersageergebnisse werden obsolet. Dabei kann das simple Modell auch mit echtem Kontext, also zusätzlich zur Historie noch mit n nachfolgenden Wörtern trainiert werden. Trainingsumstand ist die Vorhersage des zentralen Wortes aus dem echten Kontext als Eingang (Stetiges BOW Modell) oder verkehrt herum die Vorhersage des Kontextes aus dem zentralen Wort als Eingang (Stetiges Skip-gram Modell). Ziel bleiben für die künstliche Dialogumgebung die im Unterprozess generierten Worteinbettungen.
Der entgegengesetzte Ansatz wäre eine Vergrößerung der Komplexität mit rekurrenten Architekturen für eine andere Anwendung. Long-Short Term Memory Modelle (LSTM) bspw. ermöglichen das Training mit Gedächtnis, quasi unbegrenztem Kontext und hoher Genauigkeit[34][35]. Dabei besteht die versteckte Schicht aus zeitverzögerten, (teilweise) rückgekoppelten Neuronen. Trotz dieser Optimierungen bleibt das Problem der seltenen oder mehrdeutigen Wörter bestehen.

An den Worteinbettungen[20] bemerkenswert und zunächst überraschend sind die sinngebenden linear-algebraischen Operationen: So besteht die Vektorgleichung

$$v(K\ddot{o}nig) - v(Mann) + v(Frau) = v(K\ddot{o}nigin).$$

Intuitiv würde man daher glauben, dass die Komposition von Wörtern eines Satzes den Sinn des Satzes erhalten könnte. In der Praxis sind einige semantische Satzeigenschaften kaum darstellbar: In der zusammengesetzten Darstellung werden Merkmale wie Wortreihenfolge, Grammatik oder Rhetorik ignoriert. Auch werden alle Wörter inklusive Rauschen gleichgewichtet (Stop-Wörter Problem). Verbessert werden die Operationen mit statistischer Gewichtung wie das bereits bekannte TF-IDF, der Gewichtung nach grammatikalischen Einheiten (engl. *POS Weighting*) oder Parse-Tree-Ordering[32]. Zusätzliche Merkmale des Paragraphen[21], in dem der Satz gefunden wird, können in das Training fließen und erlauben eine Art Meta-Vorhersage: So tauchen ähnliche Sätze zwischen ähnlichen Sätzen auf und es wird trainiert mit dem Ziel, einen Satz zwischen anderen Sätzen vorherzusagen[14][12].

[20] Die Modelle werden daher auch unter *Word2Vec*, übersetzt *Wort-zu-Vektor*, zusammengefasst.

[21] bspw. Thema, Autor, Zeit etc.

Dennoch ist eine schwer-semantische Klassifikation mit vollem Verständnis von Sätzen in dieser Form nicht möglich und es wird sich auf leicht-semantische Aufgaben beschränkt wie die Abbildung der Vektorkompositionen auf einen anderen Raum, bspw. einer anderen Sprache für maschinelle Übersetzungen[41]. Dieser Ansatz spielt eine zentrale Rolle in der Arbeit und wird später wieder aufgegriffen.

2.5 Generative Modelle

Generative Modelle reagieren nicht mit gespeicherten Aussagen, sondern bilden sie anhand trainierter Abfolgen aus kleineren Einzelgliedern. Das können Wörter, aber auch Zeichen sein. Je nach Ansatz entstehen dabei möglicherweise zuvor ungesehene Sätze oder Wörter. Daher wird zusätzlich zur semantischen Genauigkeit für den künstlichen Agenten die wesentlich strengere syntakische Gültigkeit der Reaktionen gefordert, die sich bei abfragenden Modellen in der Verantwortung der Einstellung bzw. des Einstellers befand. Theoretisch sind generative Modelle aber mächtiger: Sie suchen keine Abbildung aus einem bekannten Reaktionsraum, sondern Übergänge zu einem nächsten Teilzustand einer Sequenz auf Wort- oder Zeichenebene und kommen daher einem natürlichen Dialogagenten funktional näher als abfragende Modelle. In der Praxis sind auch rein generative Modelle noch nicht ausreichend, um zufriedenstellende Dialoge führen zu können[33][30].

2.5.1 Probabilistische Modelle

Probabilistische generative Modelle sind im Wesentlichen Markov-Ketten[31][22]: Anhand eines möglichst passenden Startzustandes[23] wird der wahrscheinlichste Übergang in einen nächsten Zustand gesucht. Das kann auch eine zufällige Auswahl zwischen den k wahrscheinlichsten Übergängen sein. Dabei gilt die Markov-Annahme: Die Suche stützt sich zusätzlich zu sekundären Merkmalen nur auf eine begrenzte Anzahl bereits gewählter Zustände. Primäres Kriterium ist wie beim abfragenden N-Gramm Modell die Wahrscheinlichkeitsverteilung des Trainingskorpus mit der Maximum-Likelihood-Annahme. Tatsächlich ist der Trainingsprozess hier derselbe wie beim abfragenden Modell mit gleicher Bezeichnung.

[22] Eine Anwendung mit gleicher Logik ist die Generierung von sinnfreiem Text als Parodie-Generator[11][13] (engl. *dissociated press*).

[23] Das kann wieder ein Wort oder ein Zeichen sein.

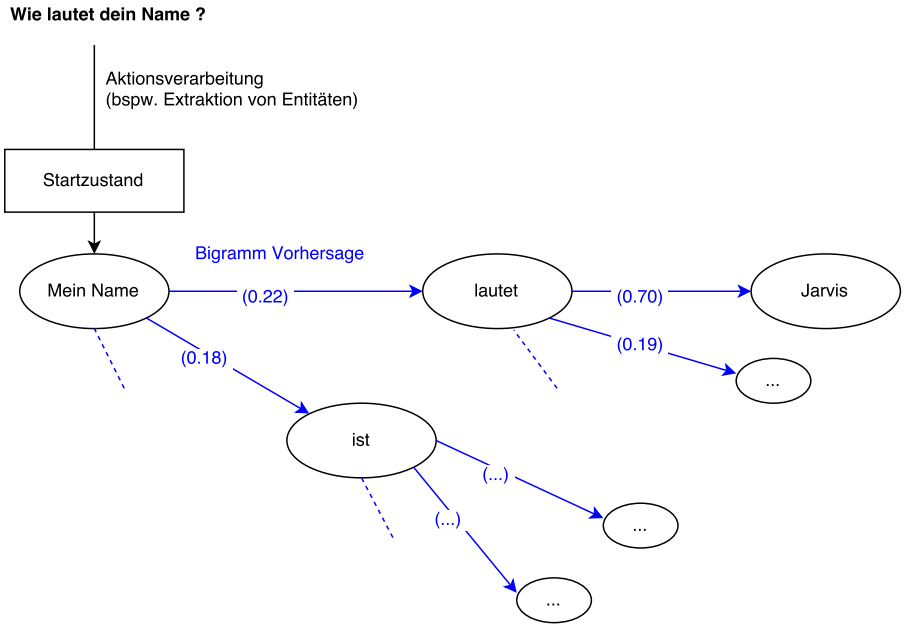

Abbildung 2.10: Markov-Ketten-Modell: Die Verarbeitung der natürlichen Sprache (engl. *natural language processing*, NLP) für den passenden Startzustand kann dabei unterschiedlich anspruchsvoll sein.

2.5.2 Neuronales Modell (HRED)

Neuronale Modelle, vor allem das Sequence-to-Sequence-Modell mit rekurrenter Architektur wie beim hierarchischem rekurrentem Enkoder-Dekoder (HRED)[29], werden in maschinellen Übersetzungen verwendet. Bei Einsatz für künstliche Agenten ist die Eingabe statt einer zu übersetzenden Sprache die Wort- und Satz-Historie[24] als erweiterter Kontext des laufenden Dialoges. Je ein rekurrentes neuronales Netz (RNN) enkodieren dabei die notwendigen Informationen:

1. RNN: Die letzte Aktion, auf die der Agent reagieren soll, wird in die interne neuronale Darstellung überführt. Sie wird dabei bidirektional verarbeitet, also die Wörter sowohl vom Anfang zum Ende als auch vom Ende zum Anfang eingelesen. Die letzte Schicht bildet einen charakterisierenden Satzvektor.
2. RNN: Der erweiterte Kontext, also der bis dahin geführte Dialog zwischen den Agenten, wird mit dem Satzvektor in die interne neuronale Darstellung dieses Netzes überführt. Die letzte Schicht bildet dadurch einen Kontextvektor.

[24] Also auch über Turns hinweg.

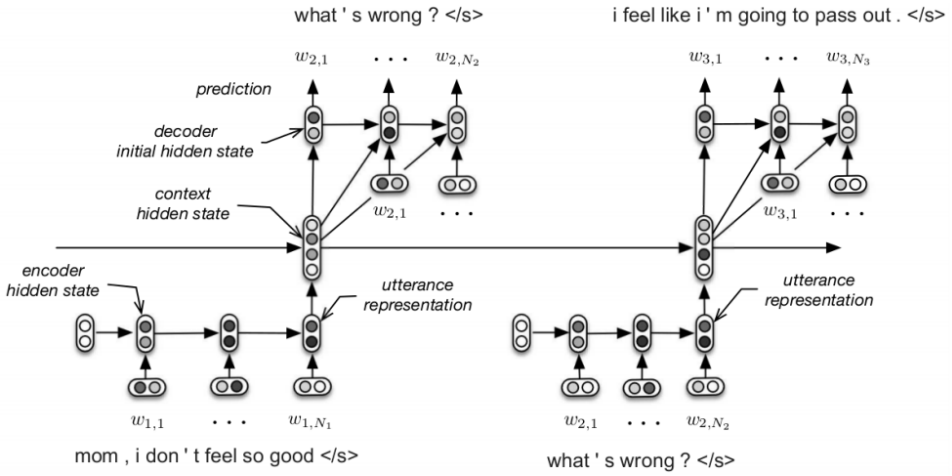

Abbildung 2.11: Modell mit hierarchischem rekurrentem Enkoder-Dekoder (HRED) nach Sordoni et al.[33]

3. RNN: Abschließend wird durch ein letztes rekurrentes Netz der bisher verarbeitete Dialog in eine passende Reaktion Wort für Wort dekodiert.

Die Zwischenergebnisse werden als semantische Zusammenfassung des bisherigen Dialoges aufgefasst. Dabei wird auch die Reihenfolge der Aussagen beachtet. Im Vergleich mit den Anforderungen einer nützlichen Übersetzung ist der künstliche Dialog schwieriger: Es gibt weniger plausible Übersetzungen einer Aussage als plausible Reaktionen auf diese Aussage. Trainiert wird das Modell off-line bspw. anhand von Film- und Theatermanuskripten. Beschleunigt wird die Einstellung dabei mit den bekannten Worteinbettungen, die anderweitig trainiert wurden. Aufgrund einer Überanpassung tendiert das Modell schnell zu generischen, aber plausiblen Reaktionen wie „Ich weiß nicht“. Diesem Mangel an Diversität wird mit einer zufälligen Auswahl zwischen guten Reaktionen statt der direkten Ausgabe der besten Reaktion entgegen gewirkt.

2.6 Zusammenfassung

In dieser Arbeit wird der Fokus bewusst auf abfragende Modelle gelegt. Generische Modelle bieten zurzeit höchstens zu Unterhaltungs- oder Lehrzwecken[25] einen Mehrwert. Dennoch steckt mehr Potential in der Mächtigkeit

[25]Der Gedanke ist, beim Lernen von Fremdsprachen automatisiert möglichst viel Kontakt mit der Sprache zu erhalten[10].

generativer Modelle und es wird davon ausgegangen, dass sie langfristig den schwer individualisierbaren abfragenden Modellen überlegen sind. Bis dahin machen sie eigenständig noch zu viele sowohl semantische als auch grammatikalische Fehler und benötigen weitaus mehr Trainingsdaten. In der Industrie werden hybride Ansätze getestet und es wird weniger von einem Rennen beider Modelle als von einem Konvergieren zueinander gesprochen. Gemeinsam versuchen sie für den künstlichen Agenten das Problem der nützlichen Reaktion zu lösen.

Kapitel 3

Reaktionseinbettungen

In diesem Kapitel wird ein neuer Ansatz für abfragende künstliche Dialogagenten vorgestellt. Dabei werden Worteinbettungen, die durch das flache verteilte Modell[22] trainiert wurden, auf einen separaten Reaktionsraum gleicher Größe $\mathbb{R}^{|m|}$ mit Merkmalen m abgebildet. Der Reaktionsraum ist im Vergleich zu den trainierten Worteinbettungen (Aktionsraum) zunächst leer und wird aus aktiven natürlichen Dialogen on-line mit Reaktionsvektoren befüllt[1]. Dabei übernehmen die Reaktionsvektoren die Merkmalskoordinaten der zugehörigen Aktionsvektoren, die als Satzvektor aus den einzelnen Worteinbettungen zusammengesetzt sind: Vereinfacht zeigt jeder verarbeitete Satz auf eine natürlich aufgetretene Reaktion und jeder Aktionsvektor ist bei Abbildung zugleich auch Reaktionsvektor.

Das Modell setzt ein Pseudotraining voraus: Die Reaktionseinbettungen werden von Hand eingelesen, übernehmen aber Vorteile neuronal trainierter Systeme. So werden mit jedem Speicherungsschritt parallel mehrere semantisch ähnliche Aktionen auf eine natürliche Reaktion abgebildet und der Zugriff erfolgt über linear-algebraische *Nearest-Neighbor*-Operationen. Dabei sind die Reaktionen nicht auf rein textuelle Darstellungen beschränkt, sondern können auch als Verweise, Bilder, Videos, etc.[2] im Reaktionsraum kodiert werden. Aufgrund gleicher Dimensionen fällt zudem eine aufwändige Optimierung einer Transformationsmatrix, aber auch eine mögliche Verfeinerung der Abbildung mithilfe von zusäzlichen Merkmalen zwischen den Räumen weg.

Mit steigender Laufzeit konvergiert der Nutzen [3] gegen einen Sättigungswert

[1] Semantische Abbildungsvorschrift ist somit der Turn von Aktion a auf Reaktion r.

[2] Funktionsaufrufe als automatisierte Reaktion können den Dialog als Benutzeroberfläche für RPCs (Remote Procedure Calls) verwenden.

[3] abgesehen von Schwankungen aufgrund gegensätzlicher Überschreibungen oder starken thematischen Sprüngen

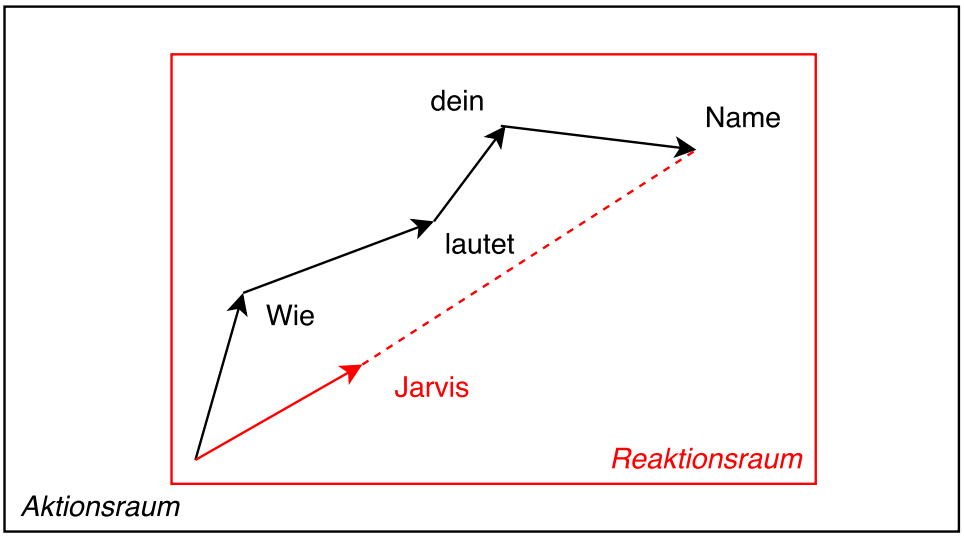

Abbildung 3.1: Aufsummierte Wortvektoren: Ähnliche Wörter erzeugen dabei eine ähnliche Vektorsumme und werden auf die nächste Reaktion abgebildet. Die Reaktion wird dennoch in einem separaten Raum gespeichert und sollte, um verschiedene Satzlängen zu unterstützen, vorher normalisiert werden.

abhängig vom Einsatzgebiet. Denkbar wäre bspw. ein Kundendienst, der einfache, wiederkehrende Anfragen mit reziprokem Aufwand verarbeitet.

3.1 Implementierung

Die Dialogumgebung wird mithilfe eines Netzwerkes in einer Client-Server-Architektur implementiert. Für Steuerungszwecke wird das HTTP-, für den Nachrichtenaustausch das MQTT-Protokoll eingesetzt. Drei eigenständige Server bilden dabei die zentrale Schnittstelle der Kommunikation:

1. Der **Verwaltungsserver** baut auf dem Java Server Framework *Dropwizard*[6] auf und verarbeitet über REST Services Anfragen zu sowohl natürlichen als auch künstlichen Agenten. Das können Neuerstellung, Zugriff, Änderung oder Löschung von Informationen sein. Dazu legt er Benutzerkonten zur Verifikation an.
2. Der **Datenbankserver** bildet den Langzeitspeicher für die Informationen, über die der Verwaltungsserver verfügt. Er baut auf der NoSQL Software *MongoDB*[24] auf.
3. Der **Dialogserver** leitet die Aussagen aktiver Dialoge in Form von MQTT-Nachrichten weiter und ist auch dafür zuständig, dass nicht

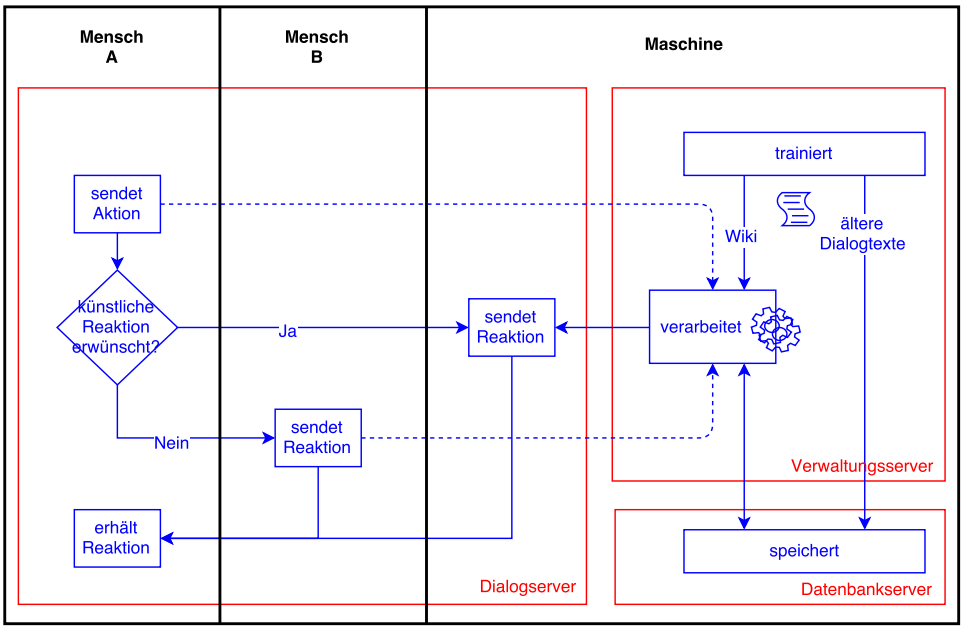

Abbildung 3.2: Die Implementierung der Dialogumgebung.

abgeholte Nachrichten bis zum bestätigten Empfang in einem proprietären Speicher zwischengesichert werden. Er baut auf dem C++ Broker *Mosquitto*[25] auf.

Die Implementierung der natürlichen Clients hängt vom Einsatzgebiet ab. So bietet sich ein mobiler Client an, unabhängig von Zeit und Ort Dialoge führen zu können und dadurch den Reaktionsraum möglichst effektiv zu füllen. Das weitverbreitete Android Betriebssystem mit dritter MQTT Unterstützung wird daher zunächst einem stationären Client vorgezogen.

Die Implementierung des künstlichen Agenten ist ein unabhängiger Client, aufgesetzt mit der Bibliothek *Deeplearning4J*[36] für maschinelles Lernen. Die parallelisierbare Bibliothek verfügt u. a. über ein Vektorraum-Modellierungstoolkit mit Klassen bzw. Methoden zur Darstellung, Speicherung und Verarbeitung[4] von Vektoren im Raum. Dazu besitzt die Bibliothek auch eine Implementierung für das Training und die Speicherung von Worteinbettungen: Entweder werden sie von Grund auf am eigenen Korpus zuerst gebildet und kontrolliert, oder über vorgefertigte Datenpakete eingespeist. In dieser Arbeit werden beide Möglichkeiten eingesetzt. Sprachabhängig wird dabei mit jeweiligen Wikipedia-Enzyklopädien und Nachrichtenartikeln trainiert: In dieser Arbeit ist das die deutsche[42] und die englische[22] Sprache.

[4] Essentiell ist die Möglichkeit zur (De-)Serialisierung von Vektoren auf die Festplatte.

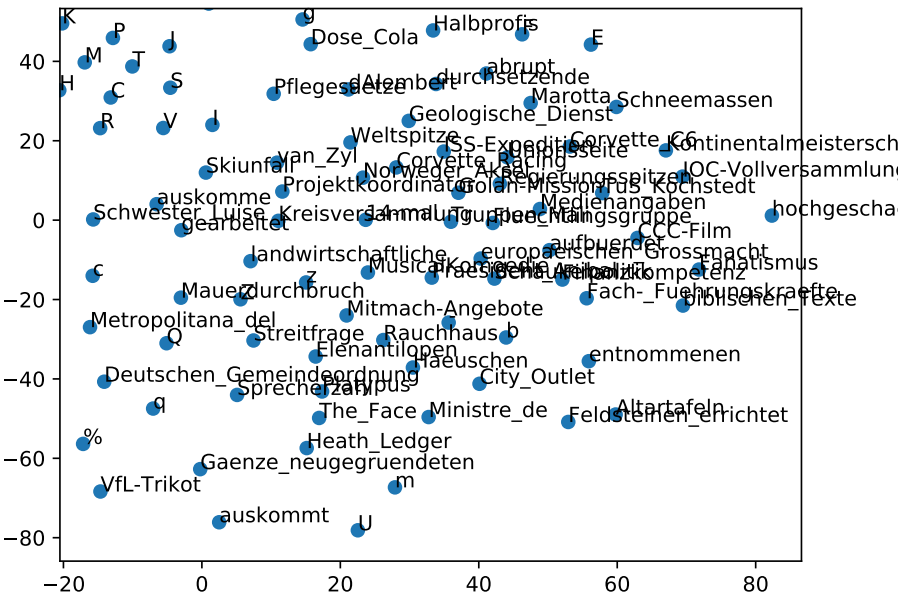

Abbildung 3.3: Ein kleiner Ausschnitt der trainierten Worteinbettungen zur Veranschaulichung des Raumes. Die Dimensionszahl wurde mit t-SNE[5] drastisch reduziert.

Der Einsatz der Reaktionseinbettungen kann in zwei Phasen betrachtet werden:

- Die **Einstellungsphase** verfügt über eine natürliche Lernumgebung um den Reaktionsraum zu füllen: Mindestens zwei natürliche Agenten im aktiven Dialog oder Protokolle vergangener Dialoge bilden die Grundlage für die semantische Genauigkeit des künstlichen Agenten. Daher ist der künstliche Agent in dieser Phase passiver Zuhörer.
- Die **Einsatzphase** verfügt über die künstliche Dialogumgebung mit nun größtenteils semantisch passenden Reaktionen. Der künstliche Agent ist hier aktiver Teilnehmer, der auf Anfragen zeitgleich und parallelisiert reagiert.

Nicht ausgeschlossen ist eine Überschneidung beider Phasen zu einem Einsatz mit gleichzeitiger On-line Einstellung und zusätzlichen Korrektur- und

[5] t-SNE ist eine Einbettung in Räume kleinerer Dimensionen abhängig von stochastischen Nachbarsbeziehungen mit Studentscher t-Verteilung (engl. *t-distributed stochastic neighbor embedding*). Das ermöglicht den Plot zur vereinfachenden Visualisierung.[18]

Moderationsmöglichkeiten vergangener Aussagen. Ein Moderatoragent könnte bspw. als erweitertes Feedback das Pseudotraining überwachen und für gezieltere Reaktionen sorgen.

3.2 Evaluation

Die Evaluation von Modellen für Dialogsysteme ist Gegenstand aktueller Forschung[17]. So wird von überwachter Evaluation gesprochen, wenn menschliche Signale zu Zufriedenheit oder Nutzen als natürliches Feedback in die Bewertung eingehen, und von unüberwachter Evaluation, wenn automatisierte Metriken wie BLEU oder METEOR[6] den Erfolgsgrad bestimmen. Diese Metriken gründen auf der Anzahl an Wortüberlappungen einer Referenzreaktion und der Reaktion des künstlichen Agenten. Daher sind sie ungenau: Eine Aktion kann viele valide Reaktionen besitzen und so eine Diskrepanz zwischen den Metriken und menschlicher Beurteilung verursachen. Es wird vermutet, dass die Evaluation von künstlichen Dialogsystemen dem gleichen Schwierigkeitsgrad unterliegt wie die Entwicklung dieser Systeme[17]. Eine nützliche Evaluation der Reaktionseinbettungen wird deshalb mit einer Kombination aus der Kosinus-Ähnlichkeit

$$\delta_s = \{x \in \mathbb{R} \mid -1 \leq x \leq 1\}$$

zwischen der verarbeiteten Aktion und der gespeicherten Aktion, auf der die ausgewählte Reaktion eigentlich basiert, und einer menschlichen Bewertungsskala angenähert[7].

Die erste Dialogumgebung A wird mit einem Datenpaket für Worteinbettungen installiert, dass mit der deutschen Wikipedia-Enzyklopädie und Nachrichtenartikeln trainiert wurde[42]. Daher kann keinen Einfluss auf die Vorverarbeitung des Korpus genommen werden und alle Aktionen aus Dialogprotokollen oder während des aktiven Dialoges müssen dieselbe Verarbeitung erhalten, um die passende Erkennung erzielen. Das sind hauptsächlich

- die Konvertierung von Umlauten (ä wird zu ae usw.),
- die Konvertierung des scharfen S (ß wird zu ss),
- die Filterung von Stopwörtern,
- die Filterung bestimmter Sonderzeichen
- die Filterung zu langer oder zu kurzer Aussagen und
- die Zusammensetzung von häufigen Wortpaaren zu einer Worteinheit (engl. *sticky pairs*)[8][4].

[6] Diese Metriken werden eigentlich für maschinelle Übersetzungssysteme verwendet.

[7] Evaluiert wird von verschiedenen Benutzern nach einer begrenzten Anzahl an Interaktionen.

[8] So kann *Euro_Münze* ein Wort bzw. Token im trainierten Vokabular sein.

Tabelle 3.1: Menschliche Bewertungsskala pro Test: Jeder Gesichtspunkt wird auf einer Skala von 0 bis 5 bewertet - höhere Werte zeigen eine positivere Erfahrung.

E1	**Qualität der Antworten (Informationsgehalt).** Wie viel Sinn im Bezug zur Aktion kann den Reaktionen zugesprochen werden?	sinnfrei(0) - verständlich (5)
E2	**Unterscheidung als künstlicher Agent (sog. Turing-Test).** Ist die dahinterliegende Maschine feststellbar?	einfach(0) - schwierig(5)
E3	**Wiederbenutzungswert.** Ist eine wiederholte Nutzung in Zukunft denkbar?	unwahrscheinlich(0) - sehr wahrscheinlich(5)
E4	**Persönlichkeit (Wiedererkennungswert).** Sind Charakteristiken des ursprünglichen Dialoges bzw. der Urheber erkennbar?	nicht vorhanden(0) - stark ausgeprägt(5)
E5	**Unterhaltungswert.** Wie viel Spaß macht die Nutzung?	neutral(0) - unterhaltsam(5)

Werden unbekannte Wörter erkannt, wird stattdessen jeweils das syntaktisch ähnlichste Wort im Vokabular gesucht und eingesetzt. Dabei können die bekannte Damerau-Levensthein-Metrik oder andere Verfahren benutzt werden. *Deeplearning4J* hat für den Vergleich einen *TF-IDF*-Algorithmus, dass die Vorkommen einzelner Zeichen eines Wortes mit den Vorkommen im ganzen Vokabular vergleicht und dadurch seltenere Zeichen höher gewichtet.

Die künstliche Dialogumgbung B wird hingegen am eigenen Korpus trainiert, sodass die Vorverarbeitung flexibler an das Anwendungsziel angepasst werden kann. Sie reduziert sich hier nun auf

- die Filterung bestimmter Sonderzeichen
- die Filterung zu langer oder zu kurzer Aussagen und
- die Zusammensetzung von häufigen Wortpaaren zu einer Worteinheit.

Zudem wird in dieser Umgebung der syntaktische Vergleich für unbekannte Wörter mit der Jaro-Winkler-Distanz durchgeführt, das sich für kürzere Strings besser eignet[5].

Bei beiden Dialogumgebungen A und B sind im Test die Benutzer gleichzeitig auch die Urheber der Dialogprotokolle, die dem künstlichen Agenten als semantische Basis dienen. Dabei sind die ursprünglichen Dialoge sprachlich lässig geführt - bei den Urhebern steht die schnelle Überlieferung von Inhalt im Vordergrund und Rechtschreibung oder Grammatik besitzen einen

niedrigeren Stellenwert.

Für die Dialogumgebung C werden die bisherigen Reaktionen mit einem öffentlichen Dialogkorpus[15] zusammengeführt, das auch formell geführte Dialoge beinhaltet.

Für die Dialogumgebung D werden als Datenpaket gespeicherte, englische Worteinbettungen[22] mit aktuellen Reaktionen aus dem englischsprachigen, sozialen Netzwerk *Reddit*[9] verwendet. Sie besitzt daher auch die größte Anzahl verfügbarer Reaktionen.

Das Modell mit Reaktionseinbettungen ermöglicht der semantischen Suche die Effizienz und Multidimensionalität neuronal verteilter Wortrepräsentationen. Zusammen mit der Verarbeitung von Dialogprotokollen wird dadurch erhofft, die Einstellung von künstlichen Agenten flexibler und die Erkennung von Aktionen intuitiver zu gestalten. Die folgenden Tabellen fassen die Beobachtungen nun jeweils zusammen:

[9] http://www.reddit.com

Tabelle 3.2: Künstliche Dialogumgebung A.

Sprache	Deutsch
Worteinbettungen	Datenpaket
Verarbeitung	Umlaute ersetzt; Stopwörter gefiltert; Wortpaare verbunden
Reaktionen	12.820
OOV Behandlung	syntaktische Ersetzung (TF-IDF)
Benutzer	5 (= Urheber)
Evaluation (arithm. Mittel)	
Kosinus-Ähnlichkeit[10]	0.8166
E1	1
E2	0,5
E3	2
E4	2,5
E5	3,5
E ∅	1,9
Ursprung	
Art	Mobiler Gruppendialog (privat)
Dialogführung	lässig
Urheber	11

[10]Das arithmetische Mittel der Kosinus-Ähnlichkeit wird anhand der Dialogbeispiele im Anhang A bestimmt.

Tabelle 3.3: Künstliche Dialogumgebung B.

Sprache	Deutsch
Worteinbettungen	neu trainiert
Verarbeitung	Wortpaare verbunden
Reaktionen	10.064
OOV Behandlung	syntaktische Ersetzung (Jaro-Winkler)
Benutzer	3 (= Urheber)
Evaluation (arithm. Mittel)	
Kosinus-Ähnlichkeit	0.8411
E1	0
E2	1
E3	0
E4	2
E5	1
E ∅	0,8
Ursprung	
Art	Mobiler Gruppendialog (privat)
Dialogführung	lässig
Urheber	3

Tabelle 3.4: Künstliche Dialogumgebung C.

Sprache	Deutsch
Worteinbettungen	neu trainiert
Verarbeitung	Wortpaare verbunden
Reaktionen	70.275
OOV Behandlung	syntaktische Ersetzung (Jaro-Winkler)
Benutzer	5
Evaluation (arithm. Mittel)	
Kosinus-Ähnlichkeit	0.8852
E1	1
E2	2
E3	1
E4	2
E5	2
E ∅	1,6
Ursprung	
Art	Mobile Gruppendialoge (privat) & öffentlicher Dialogkorpus (anonymisiert)
Dialogführung	lässig & formell
Urheber	unbekannt

Tabelle 3.5: Künstliche Dialogumgebung D.

Sprache	Englisch
Worteinbettungen	Datenpaket
Verarbeitung	keine
Reaktionen	213.981
OOV Behandlung	syntaktische Ersetzung (Jaro-Winkler)
Benutzer	5
Evaluation (arithm. Mittel)	
Kosinus-Ähnlichkeit	0.8537
E1	2,5
E2	1,5
E3	1,5
E4	1,5
E5	3
E ∅	2
Ursprung	
Art	Soziales Netzwerk
Dialogführung	überwiegend formell
Urheber	unbekannt

Kapitel 4

Fazit

Die Ergebnisse des eigenen Ansatzes in dem Datenumfang fallen als rein abfragendes Modell nüchtern aus. Menschliche Qualität und Rationalität der Reaktionen werden bemängelt - größtenteils weil auf einige Aktionen dieselbe Reaktionen folgen (generische Antworten) oder die Auswahl willkürlich erscheint. Zwar kommen überraschend treffende Reaktionen während der Nutzung vor, aber der Großteil erscheint nicht nützlich. Um das Modell zu verbessern, werden daher folgende Maßnahmen erwägt:

Eine bessere Behandlung unbekannter Wörter (UNK-Token). Bei einem lässigen Umgang von Rechtschreibung und Grammatik, wie es aktuell oftmals bei mobilen Anwendungen der Fall ist, findet das Modell häufig keine Entsprechung der Wörter in seinem trainierten Vokabular. Insbesondere die deutsche Sprache erfordert eigentlich die Großschreibung von Nomen, dass bei zwanglosen schriftlichen Dialogen vernachlässigt wird. So werden aus *Macht* oder *Rechte* die semantisch entfernten Wörter *macht* bzw. *rechte* und die Abbildung wird dadurch ungenauer. Daher kann das eigene Training für diese Anwendung von vornherein nur mit konvertierten Großbuchstaben durchführt werden. Zusätzlich sollten die Aktionen vorbehandelt werden - dazu gehören die Entfernung von Sonderzeichen oder Zahlen und die Anpassung der Umlaute an das Vokabular.

Eine andere Gewichtung von Stopwörtern. Bei dem Training der Worteinbettungen wurden Stopwörter vorher aus dem Korpus entfernt, um den Informationsgehalt seltenerer Wörter nicht zu verfälschen. Das entspricht der konventionellen Empfehlung beim Vergleich alleinstehender Wörter und erklärt auch die unverhältnismäßig hohe, maschinell berechnete Ähnlichkeit der ausgewählten Reaktionen trotz schlechter Ergebnisse bei der menschlichen Bewertung: Denn in einem Satz integriert bzw. als Komposition zu einem Satzvektor tragen sie oft signifikant der Semantik bei. So haben die Sätze 'Wer bin ich?' und 'Wo warst du?' verschiedene Bedeutungen und be-

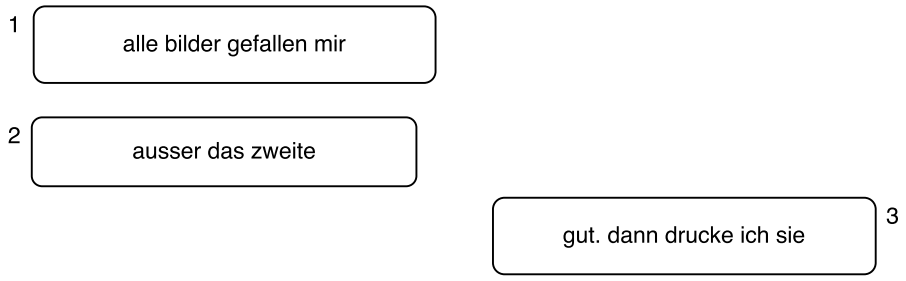

Abbildung 4.1: Aufgeteilte Aktion.

nötigen entsprechend verschiedene Reaktionen, obwohl sie gänzlich mit Stopwörtern gebildet worden sind. Vor allem im alltäglichen Dialog ist die Entfernung von Stopwörtern somit nicht gerechtfertigt. Als Kompromiss könnten sie daher nur schwächer gewichtet in die Aktionserkennung einspielen.

Eine bessere Auswahl der Aktionen und Reaktionen im Pseudotraining. Der ideale Dialogverlauf für das Modell wären unmittelbare Wechsel zwischen Urhebern - genauer sollte kein Urheber mehrere Aktionen hintereinander ausführen dürfen bzw. sollte das Modell sie einem Urheber zuordnen und als Einheit registrieren können. Ist das nicht der Fall, entstehen unpassende Abbildungen innerhalb der semantisch aufgeteilten Aktion. Eine Vorbehandlung des Ursprungsdialoges ist also diesbezüglich nötig. Es sollte aber darauf geachtet werden, dass Teile solcher Sequenzen nicht einfach herausgefiltert werden: Es bleibt der Grundsatz, dass eine Reaktion, wenn auch mit minimaler semantischer Abhängigkeit, besser ist als keine.

Eine Optimierung der Anzahl an Urhebern. Dialoge mit mehreren Urhebern (Gruppendialoge) unterliegen zeitlichen und inhaltlichen Dynamiken, die nicht bestimmbar sind. So entstehen parallele semantische Unterdialoge, wenn ein Urheber auf eine ältere Nachricht antwortet oder ein Unterdialog zwischen anderen Unterdialogen aufhört. Das Modell kann solche Strukturen, die mit wachsender Urheberzahl wahrscheinlicher werden, nicht erkennen. Ein Pseudotraining mit einem echtem Dialog zwischen zwei Urhebern könnte eine bessere Zuordnung gewähren.

Eine hybrides Modell mit abfragender und generativer Logik. Das eigene abfragende Modell beschränkt sich bei der Abbildung auf eine gespeicherte Reaktion, dass sich möglicherweise von den semantisch nächsten Reaktionen aufgrund oben genannter Dynamiken stark unterscheidet. Eine zusammengefasste bzw. generierte Reaktion aus den n nächsten Reaktionen würde die Abbildung genauer werden lassen. Zu beachten ist, dass die Probleme generativer Modelle dann zusätzlich gelöst werden müssen.

Eine Überwachung des Pseudotrainings. Alle Ursprungsdialoge wur-

den nicht in dem Wissen geführt, sie später einem künstlichen Dialogsystem vorzulegen. Daher beinhalten sie eine gewisse Gleichgültigkeit gegenüber Rechtschreibung, Grammatik und Dialogführung[1]. Wird den Urhebern eine Verantwortung im Hinblick zukünftiger Automatisierung zugeteilt, wie es bei Kundendiensten der Fall sein kann, so werden die Dialoge bewusster geführt und die semantischen Bezüge dadurch genauer.

Eine Vergrößerung der Datenmenge. Die Ursprungsdialoge der eigenen künstlichen Umgebungen haben bezüglich der möglichen Aktionen den Reaktionsraum nicht ausreichend gesättigt. Das ist erkennbar an der durchschnittlichen maschinellen Distanz, die die Umgebungen erreichen. So stützt sich der eigene Ansatz auf die Hypothese neuerer Modelle, dass simple Algorithmen mit großer Datenmenge komplexen Algorithmen mit weniger Datenmenge funktional überlegen sind. Eine Sättigung bzw. ein Erreichen eines ausreichend kleinen Intervalls um den Grenzwert der Konvergenz hätte eine minimale Kosinus-Ähnlichkeit zur Folge. Welche Datenmenge dazu benötigt wird, hängt von der Anwendung und den Daten selbst ab.

Eine zufällige Auswahl aus passenden Reaktionen. Durch das Pseudotraining werden vorherige Reaktionen bzw. Reaktionseinbettungen überschrieben, wenn eine neue Reaktion die gleiche Vektorkomposition erzeugt. Die ausgewählte Reaktion ist daher die chronologisch zuletzt gespeicherte. Sie kann aber möglicherweise stärker vom erweiterten Dialogkontext abhängig sein als vorherige Reaktionen und daher trotz hoher maschineller Bewertung als unpassend empfunden werden. Dieser Effekt kann gedämpft werden, indem eine (begrenzte) Anzahl gleich oder ähnlich bewerteter Reaktionseinbettungen gespeichert werden und schließlich aus dieser Menge eine Reaktion zufällig ausgewählt wird. Eine größere Datenmenge lässt den künstlichen Agenten dadurch variationsreicher werden und die menschliche Bewertung bzgl. Persönlichkeit und Unterhaltung verbessern.

Bemerkenswert ist, dass im aktuellen Zustand die künstlichen Agenten einen hohen Unterhaltungsnutzen besitzen und die Persönlichkeit des Ursprungsdialoges prägnant hervorsticht. Es ist demnach nur eine relativ geringe Datenmenge nötig, um menschliche Emotionalität, im Vergleich zur Rationalität, abzubilden. Daher bestätigen die eigenen Ergebnisse wiederum Ergebnisse älterer Arbeiten zu künstlichen Dialogagenten[9][38].

[1] Beispielsweise werden falsche Nachrichten nicht korrigiert. Das können Schreibfehler, aber auch unbeabsichtigte Nachrichten aufgrund falscher Addressierung oder das verzögerte Absenden im Netzwerk sein.

Anhang A

Dialogbeispiele

[Natürliche **Aktion**]

X	[Künstliche **Reaktion** der Dialogumgebung X][1]
Y	...

1. guten tag / good day

A	guten rutsch
B	alles drin gell
C	na ihre frage
D	I love this! Thank you, kind redditor!

2. wie geht es dir / how are you

A	guten rutsch
B	ehrlich gesagt nerven diese situationen und man ist verdammt hilflos in jeder situation
C	grad hier
D	Thank you kindly, I am very well!

3. schönes wetter heute / beautiful weather today

A	wo seids
B	nobodys perfekt richtig lustig
C	ihr ******
D	Actually I think most of the Internet trolls that they are using are in St

[1] Lange Reaktionen wurden gekürzt. Eine genauere Beschreibung der Dialogumgebungen steht auf den Seiten 35 und 36.

4. wie viel uhr ist es / what time is it
   - A | war sehr gut
   - B | er hat nicht explizit danach gefragt
   - C | er hat nicht explizit danach gefragt
   - D | Business time!

5. was ist der sinn des lebens / what is the purpose of life
   - A | schick mal
   - B | die ist doch kaes
   - C | das leben leben zu können und vor allem jeden tag zu geniessen
   - D | Why isn't there an alternate universe where he can get his wife back?

6. was ist dein zweck / what is your purpose
   - A | bierbrauer
   - B | ein spiel
   - C | lacht chatten bei bluewin im ernst den ellbogen gebrochen ohne dieses malheur hätte ich meinen freund nicht kennen gelernt
   - D | To serve me butter.

7. wie viel ist 1 plus 1 / What is 1 plus 1
   - A | schreib dann v zurück auf die erste pos und mach des selbe mit dem 2ten element oder was
   - B | und viel ist gut
   - C | master im ausland geht wenn die uni im ausland sie akzeptiert
   - D | ...adopted? my doggies id say are siblings! one is a pom the other a collie

8. ich liebe dich / i love you
   - A | frohes neues jahr
   - B | mh
   - C | frohes neues jahr
   - D | I love you :)

9. ich hasse dich / i hate you
   - A | könnte auch pulledpork werden
   - B | zu merken auf welche seite sie stehen
   - C | zu merken auf welche seite sie stehen
   - D | The first thing you ever did in this world was touch your mom's ******.

10. was ist liebe / what is love
    - A | bin in der stadt btw
    - B | parasitaere wuermer
    - C | ich kann auch wasganz toll das wirste danns ehen wenn du nichta ufhörst zu nerven lovebaby
    - D | Baby don't hurt me

11. wie alt bist du / how old are you
    - A | und bestellen dann pizza oder so
    - B | sweet twenty
    - C | sweet twenty
    - D | 11 so shut the **** up.

12. lol
    - A | na dann
    - B | flixbud
    - C | lool
    - D | I laughed when I read the comment too, don't know why you're getting downvo

13. omg
    - A | aber ich bin ein tiefes wasser
    - B | alles weg
    - C | lol
    - D | I heard this exact statement last week. Gave a ride to a lady walking down

14. was machst du gerade / what are you doing
    - A | group exercise is immer ent doof
    - B | oh
    - C | oh
    - D | BUD lights *** fights

15. was machst du morgen / what will you do tomorrow
    - A | i seh nur gold weiß
    - B | oh
    - C | oh
    - D | Yeah that's the same point.

16. danke / thank you
    - A | guck mal mal
    - B | ihr
    - C | guck mal mal
    - D | Wait a second..

17. wie riecht rot / how does red smell
    - A | jemand der für was illegales ins gefängnis kommen könnte
    - B | ja das geht bei uns auch grad rum
    - C | vielleicht steht es aber auch die bezeichnung der einzelnen teile
    - D | I don't know. I feel like it smells more like updog.

18. ich gehe nach hause / i go home
    - A | apokalypso
    - B | hab mum gesagt dass sie einen kuchen holen soll
    - C | und was ist dein lieblingswein kurt
    - D | r/unexpectedHamilton

19. ich gehe nicht nach hause / i dont go home
    - A | mir is immer noch sxhlecht
    - B | und welcher neue laptop
    - C | was isch i dere saison stärki vom zsc und wo chönter euch no verbesser
    - D | thanks fam, had to sort by controversial and come down and find the first

20. du und ich / you and me
    - A | wie lang bist du da noch
    - B | hast du neuen rekord gezeigt
    - C | ich bin auch heut ein wenig langsam verzeih
    - D | I'm a grown up man, you can't molest me. You can only do what God commands

21. was ist deine lieblingsmusik / what is your favorite music
    - A | meine normale
    - B | ja man
    - C | ich höre gerne musik kenne sie zwar wenig ich bin zwar ein leichter
    - D | I'd say, pretty much anything but pop, pop rock, pop rap, new country and m

22. was ist dein lieblingsessen / what is your favorite meal
    - A | sry mate
    - B | hm
    - C | joey thai livio italian food of course sarah thai -italian - indian tracy english roast-dinners
    - D | Pepperoni, sausage, mushrooms, green chile (hold the olives)

23. ich bin dein vater / i am your father
    - A | dachte du meinst das wäre das erste mal dassd pünktlich bist
    - B | überall aussenrum is weisser rans
    - C | überall aussenrum is weisser rans
    - D | *** and you shall receive.

24. android oder ios / android or ios
    - A | isn messenger like
    - B | come
    - C | isn messenger like
    - D | While I'm not a fan of repurchasing really old ported games, and think emul

25. tabs oder spaces / tabs or spaces
    - A | im maggie
    - B | imma qt3 14
    - C | im maggie
    - D | Orange you glad I didn't say banana?

26. ich weiss nicht / i dont know
    - A | warum bist du kein coach
    - B | sie is fitter
    - C | hast du einen bestimmten lieblings gegner
    - D | Han Solo the **** out of them

27. wo wohnst du / where do you live
    - A | 5 etage links erstes büro links
    - B | villingen hbf
    - C | in zürich
    - D | Finland.

28. bist du / are you
    - A | bevor eleven o clock or gastgeb will punish youuuu
    - B | oh
    - C | machst du in der nächsten zeit mal eine ausgefflipte frisur
    - D | Am I though?

29. warum / why
    - A | hahahaha
    - B | ja man
    - C | lieg im krankenhaus mit gebrochenen schlüsselbein
    - D | He couldn't get a fire started for cremation.

30. lorem ipsum dolor sit amet

A	is anders
B	idiot
C	que pensez-vous de la mentalit suisse-allemande en quoi la trouvez diff rente de la mentalit fran aise
D	From Google Translate: > Lorem ipsum dolor sit amet, consectetur adipis

Abkürzungsverzeichnis

AGI	Künstliche allgemeine Intelligenz (engl. *artifical general intelligence*)
KI	Künstliche Intelligenz
KB	Wissensbank (engl. *knowledge base*)
POS	Wortart, Wortklasse (engl. *part-of-speech*)
NLP	Verarbeitung von natürlicher Sprache (engl. *natural language processing*)
BOW	Wörtersack (engl. *bag-of-words*)
OOV	ausserhalb des Vokabulars (engl. *out-of-vocabulary*)
COD	Fluch der Dimensionalität (engl. *curse of dimensionality*)
KNN	Künstliches Neuronales Netz
RNN	Rekurrentes (künstliches) Neuronales Netz
SGD	Stochastischer Gradientenabstieg (engl. *stochastic gradient descent*)
HRED	Hierarchisches rekurrentes Enkoder-Dekoder
ML	(ML-Methode) Parameterschätzung mit maximaler Wahrscheinlichkeit (engl. *maximum-likelihood estimation*)

Quellenverzeichnis

Literatur

[1] Satanjeev Banerjee und Ted Pedersen. „An adapted Lesk algorithm for word sense disambiguation using WordNet“. In: *International Conference on Intelligent Text Processing and Computational Linguistics*. Springer. 2002, S. 136–145 (siehe S. 15).

[2] Yoshua Bengio u. a. „A neural probabilistic language model“. *Journal of machine learning research* 3.Feb (2003), S. 1137–1155 (siehe S. 7, 19).

[3] Michael L. Brodie und John Mylopoulos. „Knowledge Bases vs Databases“. In: *On Knowledge Base Management Systems: Integrating Artificial Intelligence and Database Technologies*. Hrsg. von Micheal L. Brodie und John Mylopoulos. New York, NY: Springer New York, 1986, S. 83–86. URL: http://dx.doi.org/10.1007/978-1-4612-4980-1_9 (siehe S. 9).

[4] Peter F Brown u. a. „Class-based n-gram models of natural language“. *Computational linguistics* 18.4 (1992), S. 467–479 (siehe S. 16–18, 32).

[5] William Cohen, Pradeep Ravikumar und Stephen Fienberg. „A comparison of string metrics for matching names and records“. In: *Kdd workshop on data cleaning and object consolidation*. Bd. 3. 2003, S. 73–78 (siehe S. 33).

[6] Yammer Inc. Dropwizard Team. *Dropwizard*. 2014-2016. URL: http://www.dropwizard.io (siehe S. 29).

[7] Christiane Fellbaum. *WordNet*. Wiley Online Library (siehe S. 15).

[8] Jonathan Ginzburg und Raquel Fernández. „Action at a distance: the difference between dialogue and multilogue“. In: (Siehe S. 3).

[9] Bob Heller u. a. „Freudbot: An investigation of chatbot technology in distance education“. In: *Proceedings of the World Conference on Multimedia, Hypermedia and Telecommunications*. 2005, S. 3913–3918 (siehe S. 40).

[10] Jiyou Jia. „The study of the application of a web-based chatbot system on the teaching of foreign languages“. In: *Proceedings of SITE*. Bd. 4. 2004, S. 1201–1207 (siehe S. 26).

[11] Hugh Kenner und Joseph OROURKE. „A travesty generator for micros“. *Byte* 9.12 (1984), S. 129 (siehe S. 24).

[12] Tom Kenter, Alexey Borisov und Maarten de Rijke. „Siamese cbow: Optimizing word embeddings for sentence representations“. *arXiv preprint arXiv:1606.04640* (2016) (siehe S. 23).

[13] Cyril Labbé und Dominique Labbé. „Duplicate and fake publications in the scientific literature: how many SCIgen papers in computer science?“ *Scientometrics* 94.1 (2013), S. 379–396 (siehe S. 24).

[14] Quoc V Le und Tomas Mikolov. „Distributed Representations of Sentences and Documents.“ In: *ICML*. Bd. 14. 2014, S. 1188–1196 (siehe S. 23).

[15] Claudia Leacock, George A Miller und Martin Chodorow. „Corpora of Computer-Mediated Communication.“ *Corpus Linguistics. An International Handbook*. 1 (2008). http://www.chatkorpus.tu-dortmund.de/ (besucht am: 20-03-2017), S. 292–308 (siehe S. 34).

[16] Claudia Leacock, George A Miller und Martin Chodorow. „Using corpus statistics and WordNet relations for sense identification“. *Computational Linguistics* 24.1 (1998), S. 147–165 (siehe S. 15).

[17] Chia-Wei Liu u. a. „How NOT to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation“. *arXiv preprint arXiv:1603.08023* (2016) (siehe S. 7, 32).

[18] Laurens van der Maaten und Geoffrey Hinton. „Visualizing data using t-SNE“. *Journal of Machine Learning Research* 9.Nov (2008), S. 2579–2605 (siehe S. 31).

[19] Eric Mays, Fred J Damerau und Robert L Mercer. „Context based spelling correction“. *Information Processing & Management* 27.5 (1991), S. 517–522 (siehe S. 12).

[20] Robert Michaelsen und Donald Michie. „Expert systems in business“. *Datamation;(United States)* 11 (1983) (siehe S. 9).

[21] Tomas Mikolov u. a. „Distributed representations of words and phrases and their compositionality“. In: *Advances in neural information processing systems*. 2013, S. 3111–3119 (siehe S. 19).

[22] Tomas Mikolov u. a. „Efficient estimation of word representations in vector space“. *arXiv preprint arXiv:1301.3781* (2013) (siehe S. 19, 22, 23, 28, 30, 34).

[23] George A Miller. „WordNet: a lexical database for English“. *Communications of the ACM* 38.11 (1995), S. 39–41 (siehe S. 15).

[24] Inc. MongoDB. *MongoDB is a document database with the scalability and flexibility that you want with the querying and indexing that you need*. URL: https://www.mongodb.com/ (siehe S. 29).

[25] Eclipse Mosquitto. *An Open Source MQTT v3.1/v3.1.1 Broker*. URL: https://mosquitto.org/ (siehe S. 30).

[42] Andreas Müller. *GermanWordEmbeddings*. http://devmount.github.io/GermanWordEmbeddings/ (besucht am: 19-03-2017). 2015 (siehe S. 30, 32).

[26] Ted Pedersen, Siddharth Patwardhan und Jason Michelizzi. „WordNet:: Similarity: measuring the relatedness of concepts“. In: *Demonstration papers at HLT-NAACL 2004*. Association for Computational Linguistics. 2004, S. 38–41 (siehe S. 15).

[27] Jean-Charles Pomerol. „Artificial intelligence and human decision making“. *European Journal of Operational Research* 99.1 (1997), S. 3–25 (siehe S. 8).

[28] Claude Sammut. „Managing context in a conversational agent“. *Linkoping Electronic Articles in Computer & Information Science* 3.7 (2001) (siehe S. 10).

[29] Iulian V Serban u. a. „Building end-to-end dialogue systems using generative hierarchical neural network models“. *arXiv preprint arXiv:1507.04808* (2015) (siehe S. 5, 7, 25).

[30] Lifeng Shang, Zhengdong Lu und Hang Li. „Neural responding machine for short-text conversation“. *arXiv preprint arXiv:1503.02364* (2015) (siehe S. 24).

[31] Claude Elwood Shannon. „A mathematical theory of communication“. *ACM SIGMOBILE Mobile Computing and Communications Review* 5.1 (2001), S. 3–55 (siehe S. 24).

[32] Richard Socher u. a. „Parsing with Compositional Vector Grammars.“ In: *ACL (1)*. 2013, S. 455–465 (siehe S. 23).

[33] Alessandro Sordoni u. a. „A neural network approach to context-sensitive generation of conversational responses“. *arXiv preprint arXiv:1506.06714* (2015) (siehe S. 24, 26).

[34] Martin Sundermeyer, Hermann Ney und Ralf Schlüter. „From feedforward to recurrent LSTM neural networks for language modeling“. *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)* 23.3 (2015), S. 517–529 (siehe S. 23).

[35] Ilya Sutskever, Oriol Vinyals und Quoc V Le. „Sequence to sequence learning with neural networks“. In: *Advances in neural information processing systems*. 2014, S. 3104–3112 (siehe S. 23).

[36] Deeplearning4j Development Team. *Deeplearning4j: Open-source distributed deep learning for the JVM*. Apache Software Foundation License 2.0. URL: http://deeplearning4j.org (siehe S. 30).

[37] Richard Wallace. „The elements of AIML style“. *Alice AI Foundation* (2003) (siehe S. 10).

[38] Joseph Weizenbaum. „ELIZA—a computer program for the study of natural language communication between man and machine“. *Communications of the ACM* 9.1 (1966), S. 36–45 (siehe S. 40).

[39] Zhibiao Wu und Martha Palmer. „Verbs semantics and lexical selection“. In: *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*. Association for Computational Linguistics. 1994, S. 133–138 (siehe S. 15).

[40] Yoshua Bengio, Scholarpedia. *Neural Net Language Model direct*. [Online; besucht am 24.01.2017]. 2003. URL: http://www.scholarpedia.org/article/File:Neural_Net_Language_Model_direct.jpg (siehe S. 21).

[41] Will Y Zou u. a. „Bilingual Word Embeddings for Phrase-Based Machine Translation.“ In: *EMNLP*. 2013, S. 1393–1398 (siehe S. 24).