Karlsruher Institut für Technologie

Vergleich verschiedener Phrasenextraktionsverfahren für die Generierung passender Systemantworten in einem sozialen Dialogsystem

Bachelorarbeit
von

Meng Meng Yan

An der Fakultät für Informatik
Institut für Anthropomatik und Robotik (IAR)

Erstgutachter:	**Prof. Dr. Alexander Waibel**
Zweitgutachter:	**Dr. Sebastian Stüker**
Betreuender Mitarbeiter:	**M.A. Maria Schmidt**

Bearbeitungszeit: 8. Juli 2015 – 7. November 2015

Erklärung:

Ich versichere hiermit, dass ich die Arbeit selbstständig verfasst habe, keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe, die wörtlich oder inhaltlich übernommenen Stellen als solche kenntlich gemacht habe und die Satzung des Karlsruher Instituts für Technologie zur Sicherung guter wissentschaftlicher Praxis beachtet habe.

Karlsruhe, den 7. November 2015

Meng Meng Yan

Abstract:

In order to use machine translation systems for a social dialog system we compared three different phrase pair extraction methods. The parallel corpus was extracted from OpenSubtitles and Twitter. We used Koehn's standard method, *Fisher's Exact Test* and the *G-Test* for the phrase pair extraction. A phrase-based, statistical machine translation system was trained with the different phrasetables using different weights for the language and translation model. Finally, the results were evaluated using a survey in order to compare the different phrase pair extraction methods and the used weights.

Kurzzusammenfassung:

Zur Nutzung von maschinellen Übersetzungssystemen für soziale Dialogsysteme werden verschiedene Phrasenextraktionsverfahren verglichen. Die Daten für den parallelen Korpus wurden OpenSubtitles und Twitter entnommen. Die verwendeten Verfahren waren zum einen der Standard nach Koehn, zum anderen der *Fisher's Exact Test* und der *G-Test*. Ein phrasenbasiertes, statistisches maschinelles Übersetzungssystem wurde anschließend mit Hilfe der extrahierten Phrasenpaare trainiert, wobei unterschiedliche Gewichtungen für das *Sprach-* und *Übersetzungsmodell* erprobt wurden. Zuletzt wurde durch eine Benutzerstudie die Performanz der unterschiedlichen Extraktionsverfahren und ihren Gewichtungen verglichen.

Inhaltsverzeichnis

1 Einleitung

In der heutigen Gesellschaft gibt es zwei gegenläufige Ströme (Burnett, 2004). Die eine Seite akzeptiert den technischen Fortschritt, nimmt diesen an und schöpft dessen Nutzen aus. Technik ermöglicht ein komfortableres Leben, da sie viel mechanische Arbeit übernimmt, komplexe Rechnungen kalkuliert und auch Entertainment bietet. Die andere Seite steht dem Fortschritt kritisch gegenüber und stellt soziale und ethische Fragen. Es wird gefragt, ob unsere Gesellschaft nicht durch moderne Technik vereinsamt (Hampton u. a., 2009). Man befürchtet, dass es zu einer Abkehr von sozialen Kontakten und einer Zuwendung zu virtuellen Realitäten führt. Auch Dialogsysteme werden dieser Hinsicht nach kritisch betrachtet. Durch Dialogsysteme ist es weitgehend möglich mit technischen Geräten, wie Mobiltelefonen, Robotern oder Computern, Gespräche zu führen wie mit einem Menschen (McTear, 2002). Auf der einen Seite bietet es den Vorteil, dass die Handhabung von technischen Geräten auf diese Art und Weise erheblich vereinfacht werden kann, auf der anderen Seite kann bei unverhältnismäßigem Nutzen solcher Systeme ein Ersatz realer sozialer Kontakte durch technische Geräte erfolgen. Die Vorteile, welche Dialogsysteme bieten können, überragen bei weitem. Um einige leicht ersichtliche Vorteile anhand von nicht-funktionalen Anforderungen (Robertson und Robertson, 2012) zu nennen:

Bedienbarkeit Es ermöglicht eine intuitivere Nutzung technischer Geräte. Dies ermöglicht auch technikaversen Menschen die Vorteile von Technik zu nutzen, auch wenn ihnen das Nutzen von technischen Innovationen sonst schwerfällt, wie z.B. Senioren. Außerdem erleichtert es auch Menschen mit eingeschränkten motorischen Fähigkeiten, technische Geräte zu nutzen.

Erlernbarkeit Die Nutzung ist schneller erlernbar, da in der Regel Sprache von den meisten Menschen beherrscht wird. Es wird keine Anpassung alltäglicher Gewohnheiten erfordert. Man kann die Sprache, welche man im Alltag verwendet für die Maschinen wiederverwenden.

Handhabung Es erleichtert die Nutzung von Maschinen, da man nicht umdenken muss, wie eine Eingabe zu formulieren ist. Dadurch können Eingaben schnell erfolgen.

Verständlichkeit Da es sich bei der Eingabe um natürliche Sprache handelt, ist es leicht verständlich, da man nicht von gewohnten Tätigkeiten abweicht.

Diese Vorteile animieren Forscher, Dialogsysteme weiter zu erforschen, zu verbessern und neue Techniken zu testen. In dieser Bachelorthese werden ebenfalls für das Forschungsfeld soziale Dialogsysteme unkonventionelle Techniken erforscht, um den Nutzen von Dialogsystemen zu verbessern.

1.1 Motivation

Wie bereits erläutert bieten Dialogsysteme viele Vorteile. Daher ist es von Interesse Dialogsysteme weiter zu verbessern und neue Techniken zu erproben. Dialogsysteme werden heutzutage weitverbreitet angewandt. Dabei kann man zwischen zielgerichteten Dialogsystemen und nicht-zielgerichteten Dialogssystemen unterscheiden. Zielgerichtete Dialogsysteme findet man z.B. als Buchungssysteme für Fluglinien vor, bei welchen man über Sprache die benötigten Daten an das System weitergibt, um so einen Flug seiner Wahl zu buchen. Nicht-zielgerichtete Dialogsysteme wären beispielsweise Chatbots, wie ELIZA (Weizenbaum, 1966), welches basierend auf der Erkennung von bestimmten Schlüsselworten geeignete Antworten an den Nutzer zurückgibt. Soziale Dialogsysteme sind meist nicht zielgerichtet. Um die Nutzung solcher sozialer Dialogsysteme weiterzuverbessern, werden viele verschiedene Ansätze getestet und evaluiert. Einer davon wäre der Ansatz der Nutzung sozialer Dialogsysteme auf Basis von maschinellen

Übersetzungssystemen (Ritter u. a., 2011). Maschinelle Übersetzungssysteme werden geläufig zur Übersetzung zwischen verschiedenen Quell- und Zielsprachen genutzt, wie beispielsweise von der englischen Sprache in die deutsche Sprache. Ein Beispiel hierfür wäre Google Translate[1]. Mit einem geeigneten Datenkorpus können maschinelle Übersetzungssysteme trainiert werden, um von einer Aussagensprache in eine Antwortsprache zu übersetzen. Dabei gibt es verschiedene Möglichkeiten die Übersetzung zu beeinflussen und zu verbessern. Im Rahmen dieser Arbeit wurde vorrangig die Wahl des Phrasenextraktionsverfahrens erkundet und evaluiert, wie diese das Ergebnis der Übersetzung beeinflusst. Dazu wurden drei verschiedene Verfahren auf einen Korpus angewendet und mit der resultierenden *Phrasentabelle* ein maschinelles Übersetzungssystem trainiert.

1.2 Gliederung

Im Folgenden werden soziale Dialogsysteme und ihre herkömmlichen Mechanismen erläutert sowie maschinelle Übersetzungssysteme dargestellt. Darauffolgend wird erörtert, wie diese Übersetzungssysteme für soziale Dialogsysteme von Nutzen sein können. Anschließend wird kurz auf die Daten eingegangen und dargestellt, inwiefern der parallele Korpus für die *Phrasentabelle* verwendet wird. Im darauffolgenden Kapitel wird dargestellt, wie mit Hilfe von diesem Korpus verschiedene *Phrasentabellen* extrahiert werden, welches den Hauptteil der Implementierung ausmacht. Die Ergebnisse der Implementierung werden mit Hilfe einer Umfrage evaluiert. Abschließend folgt ein Fazit dieser Arbeit.

[1] `https://translate.google.com/`

2 Grundlagen

2.1 Soziale Dialogsysteme

Soziale Dialogsysteme ermöglichen die Kommunikation mit technischen Geräten über alltägliche Themen wie das Wetter, derzeitiges Empfinden („Wie geht es dir?“), o.Ä.. Es verfolgt keine konkreten Ziele, sondern dient lediglich dazu, die Nutzung eines technischen Geräts möglichst natürlich zu gestalten. Soziale Dialoge zielen in der Regel auf keine oder soziale Ziele, wie z.B. Meinungsermittlung oder Gemütsbestimmung des Gesprächspartners, ab. Daher handelt sich bei sozialen Dialogsystemen meist um nicht-zielgerichtete Dialogsysteme. Da soziale Dialogsysteme flexibel auf verschiedene Themen und Verhalten reagieren müssen, sind sie schwierig zu entwickeln.

2.1.1 Herkömmlicher Aufbau

Man kann unterscheiden zwischen Dialogsystemen mit geschriebener oder gesprochener Eingabe. Die Usereingabe wird von dem System verarbeitet, sodass eine Antwort als Ausgabe generiert wird. Sprachdialogsysteme bestehen in der Regel aus folgenden Komponenten: *Automatic Speech Recognition* (ASR), *Natural Language Understanding* (NLU), *Dialog Modeling* bzw. *Management* (DM), *Natural Language Generation* (NLG) und *Text-to-Speech* (TTS) (McTear, 2002). Bei Dialogsystemen mit geschriebener Ein- und Ausgabe entfallen ASR und TTS. Diese Komponenten werden in der Regel nacheinander ausgeführt und liefern basierend auf den Ergebnissen des vorangegangen Moduls Ergebnisse, die an das nächste Modul weitergegeben werden. Der Vorgang ist als ein Kreislauf darstellbar, wie in Abbildung 2.1 erkennbar. Im Folgenden werden die einzelnen Komponenten erläutert.

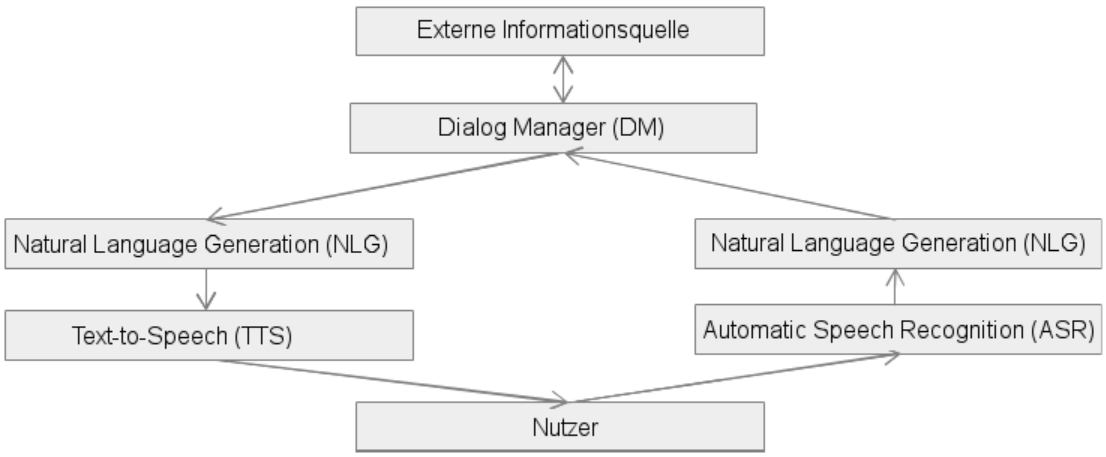

Abbildung 2.1: *Herkömmlicher Aufbau eines Dialogsystems.* Abbildung ist angelehnt an Abbildung aus (McTear, 2002). Es zeigt den Aufbau eines Dialogsystems mit seinen Komponenten ASR, NLU, DM, NLG und TTS und deutet an, wie sie untereinander zusammenhängen.

ASR

Die ASR dient der Erkennung gesprochener Sprache und ihrer Transkribierung. Sie wandelt das Sprachsignal in geschriebene Sprache um und muss in der Lage sein, mit einer hohen Wahrscheinlichkeit Wörter wiederzuerkennen und orthographisch korrekt zu schreiben. Sie baut dazu ein *Sprachmodell* aus den erkannten Worten und ein akustisches Modell um stimmliche Eigenschaften des Nutzers zu erfassen. Der in geschriebene Sprache umgewandelte Text wird anschließend an die NLU weitergegeben.

NLU

Die NLU hat die Aufgabe einen geschriebenen Text semantisch zu verarbeiten und deren semantische Repräsentation zu gewinnen. Die Schwierigkeit bei sozialen Dialogen ist, dass die Thematik sehr weitreichend sein kann, da dem Nutzer in der Regel keine Grenzen gesetzt sind. Daher ist die Umsetzung der NLU ein recht komplexer Schritt. Die semantische Repräsentation wird anschließend an die DM weitergegeben.

DM

Die DM entscheidet basierend auf dem Ergebnis der NLU, wie der Dialog mit dem Nutzer fortgeführt werden muss. Dazu gibt es sowohl regelbasierte als auch statistische Verfahren, welche in Absatz 2.1.2 erläutert werden. Je nachdem kann man verschiedene Arten von DM unterscheiden. Es gibt solche, bei welchen die Initiative vom System ausgeht. Bei solchen lenkt das System die Richtung des Gesprächs. Wiederum gibt es auch Systeme, bei welchen die Initiative vom Nutzer ausgeht. Der Nutzer lenkt den Verlauf des Gesprächs. Letzteres ist schwerer zu implementieren, da das System für erheblich mehr Fälle bereit sein muss, welche der Nutzer initiieren könnte. Im ersteren Fall kann das System die Zusammenhänge der Antwort zu gewissen Themenbereichen zuordnen und die NLU wird leichter den Sinn erfassen. Außerdem gehört zur DM auch das Error Handling. Dies dient dazu, dass im Falle von zu großen Unklarheiten Verständnisfragen an den Nutzer gestellt werden müssen. Ein Beispiel hierfür wäre „Könnten Sie das bitte wiederholen?“ oder „Haben Sie gerade *New York* gesagt?“. Für die Verarbeitung werden häufig weitere Wissensquellen benötigt, wie beispielsweise Allgemeinwissen, um den Dialog fortzuführen.

NLG

Die NLG generiert die Antwort basierend auf dem Ergebnis der DM als geschriebenen Text. Dieser Text wird aus dem semantischen Ergebnis der DM so konstruiert, dass er natürlicher Sprache gleicht. An der NLG wird seitens des Nutzers ermessen wie intelligent das System ist.

TTS

Die TTS wandelt den Antworttext in gesprochene Sprache um. Die geschriebene Sprache wird in ein Sprachsignal umgewandelt. Dabei ist für soziale Dialogsysteme wichtig, dass Emotionen verständlich gemacht werden.

2.1.2 Ansätze zur Umsetzung der DM

Der Kern des Dialogsystems, die DM, kann auf verschiedene Art und Weisen implementiert werden. Man verwendet häufig regelbasierte oder statistische Ansätze.

Regelbasierte soziale Dialogsysteme

Bei regelbasierten Systemen werden manuell Regeln definiert, welche das System in den entsprechenden Situationen befolgt. Solche regelbasierten Systeme können beispielsweise gut für zielorientierte Dialoge verwendet werden, wie die Buchung eines Flugtickets. Bei einem solchen Dialog hat das Dialogsystem meist die Initiative im Gespräch. Das System fragt nach Information und kann die Antwort auf eine

bestimmte Domäne einschränken und abhängig von der Antwort entscheiden, wie das Gespräch weitergeführt wird. Regelbasierte Systeme sind ungeeignet für soziale Dialoge, da es ein sehr umfangreiches Repertoire an Regeln bedürfte, um alle möglichen Situationen, welche in einem sozialen Dialog auftreten könnten, abzudecken. Jedoch gibt es Systeme, welche regelbasiert sind und überzeugende Dialoge führen können. ELIZA (Weizenbaum, 1966) wäre ein Beispiel für ein solches regelbasiertes System.

Statistische soziale Dialogsysteme

Statistische Dialogsysteme hingegen benötigen keine handgeschriebenen Regeln. Stattdessen werden Regeln mit maschinellen Lernverfahren aus einer großen Menge an Dialogdaten abgeleitet (Nagata und Morimoto, 1994), wobei es verschiedene Möglichkeiten gibt dabei vorzugehen. Zum Beispiel ist es möglich, Wörter abhängig von ihrem semantischen Zweck in bestimmte Klassen zu gruppieren. Auch kann man sie nach ihren syntaktischen Zweck gruppieren. Statistische Dialogsysteme sind besser für allgemeine soziale Dialogsysteme geeignet, da sie durch eine entsprechend große Datenmenge auf viele verschiedene Situationen trainiert werden können. Es ist so möglich Dialogsysteme zu entwickeln, die auf einer Vielzahl von Aussagen antworten können. Die Schwierigkeit hierbei ist jedoch eine solch große Datenmenge zu sammeln, welche das System hinreichend auf verschiedenste Situationen vorbereiten kann. Gerade in der Domäne der sozialen Dialoge ergibt sich häufig das Problem, dass Daten sporadisch sind.

Im Rahmen dieser Arbeit wurde an einem System entwickelt, welches vom herkömmlichen Aufbau abweicht. Ein maschinelles Übersetzungssystem wurde mit parallelen Konversationsdaten trainiert, so dass dieses System für eine beliebige Aussage eine passende soziale Antwort generieren könnte. Die Idee ist an den Ansatz statistischer sozialer Dialogsysteme angelehnt. Dazu wird mehr in Kapitel 3 eingegangen, um zunächst die Grundlagen der maschinellen Übersetzungssysteme darzustellen.

2.2 Maschinelle Übersetzungssysteme

Maschinelle Übersetzungssysteme (Koehn, 2009) werden zur Übersetzung zwischen verschiedenen natürlichen Sprachen verwendet, wie zum Beispiel zur Übersetzung eines englischen Textes in die deutsche Sprache. Dies bietet den Vorteil, dass hohe Dolmetscherkosten entfallen können, wenn solche Übersetzungssysteme entsprechend eingesetzt werden. Jedoch stellt die korrekte grammatikalische Übersetzung eine Schwierigkeit dar, da die Systeme häufig bei der Übersetzung die grammatikalische Korrektheit in der Zielsprache nicht einhalten.

2.2.1 Herkömmlicher Aufbau

Maschinelle Übersetzungssysteme (Koehn, 2009) bestehen in der Regel aus einem Modul für die Vorverarbeitung, einem *Übersetzungsmodell* und einem *Sprachmodell*. Die Ergebnisse des *Übersetzungsmodells* und des *Sprachmodells* werden benötigt, um die bestmögliche Übersetzung zu kalkulieren. Der Aufbau wird am folgenden Bild 2.2 veranschaulicht.

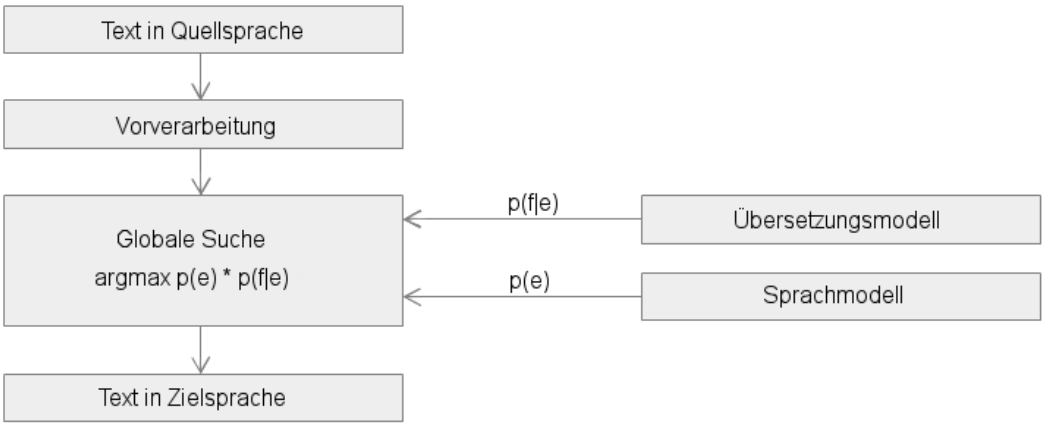

Abbildung 2.2: *Herkömmlicher Aufbau eines maschinellen Übersetzungssystems.* Abbildung ist angelehnt an Abbildung aus Koehn (2009). Es zeigt den Aufbau eines maschinellen Übersetzungssystems mit seinen Komponenten, dem Vorverarbeitungsmodul, dem *Übersetzungsmodell* und dem *Sprachmodell*. Außerdem weißt die Abbildung ihren Zusammenhang auf.

Vorverarbeitung

Bei der Vorverarbeitung wird der Eingabetext aufbereitet, sodass die folgenden Komponenten den Text bearbeiten können. Es kommt zu einer syntaktischen Aufbereitung. Mitunter wird der Text orthographisch korrigiert und normalisiert.

Übersetzungsmodell

Das *Übersetzungsmodell* dient der Übersetzung von Komponenten des Textes in die Zielsprache. Dabei gibt es verschiedene Möglichkeiten. Man kann wortweise oder auch von phrasenweise übersetzen. Der Standard heutzutage sind phrasenbasierte Übersetzungssysteme (Koehn, 2009). Sie bietet der wortbasierten Übersetzung gegenüber mehrere Vorteile, wie ein besserer Einbezug des Kontexts.
Für die phrasenbasierte Übersetzung werden *Phrasentabellen* benötigt. *Phrasentabellen* enthalten Phrasenpaare, wobei eine Phrase in der Quellsprache und die andere Phrase in der Zielsprache geschrieben ist.

Außerdem werden für solche Phrasenpaare Wahrscheinlichkeiten, auch Scores, berechnet, die aufweisen wie wahrscheinlich die Phrase der Zielsprache eine Übersetzung der Phrase der Quellsprache ist. Um solche *Phrasentabellen* zu erzeugen werden parallele Korpora benötigt, aus welchen Phrasen extrahiert werden können.

Paralleler Korpus Ein paralleler Korpus ist wesentlich zur Extraktion von Phrasen. Anhand von einem solchen Korpus können die relevanten Wahrscheinlichkeiten einer Übersetzung einer Phrase der Quellsprache in eine Phrase der Zielsprache berechnet werden. Solche Korpora kann man über verschiedene Resourcen gewinnen. Es gibt öffentlich Ressourcen, z.B. vom europäischen Parlament (Koehn, 2005). Außerdem bietet vor allem auch das Internet viele Möglichkeiten um parallele Daten zu sammeln, wie beispielsweise Wikipedia (Potthast u. a., 2008). Es ist beispielsweise möglich, dass man mehrere Artikel auf Englisch und mehrere Artikel auf Deutsch als potentielle parallele Korpora verwendet. Jedoch muss hierbei beachtet werden, dass eine Vorverarbeitung des Korpus nötig ist, da solche Artikel nicht direkte Übersetzungen voneinander sind.

Sprachmodell

Da die korrekte Anordnung von Worten häufig nicht über komplette Aussagen bei der Übersetzung beachtet werden, sondern nur innerhalb von Phrasen bei der phrasenbasierten Übersetzung, wird ein *Sprachmodell* benötigt. Ein *Sprachmodell* berechnet die Wahrscheinlichkeit, dass bestimmte Wörter in einer bestimmten Reihenfolge auftreten. Für das *Sprachmodell* werden ebenfalls Daten benötigt, wobei hier monolinguale Daten aus der Zielsprache ausreichen, da es lediglich lernt, welche Wortalinierungen in der Zielsprache möglich sind.

2.2.2 Ansätze zur Umsetzung des maschinellen Übersetzungssystems

Es gibt verschiedene Möglichkeiten maschinelle Übersetzungssysteme zu bauen. Eine davon ist der statistische Ansatz. Bei diesem verfügt man über einen parallelen Korpus. Das heißt, man hat verschiedene Aussagen in der Quellsprache und ihre entsprechenden Übersetzungen in der Zielsprache, die einander exakt entsprechen. Das System stellt Wortalinierungen zwischen Quell- und Zieltext her und trainiert ein statistisches Modell mit dem parallelen Korpus. Man erhält so verschiedene mögliche Zuordnungen bzw. Übersetzungen, welche Hypothesen genannt werden. Diese Hypothesen erhalten einen Score, welche ihre Wahrscheinlichkeit widerspiegelt. Diese Hypothesen und ihre Wahrscheinlichkeiten werden wie oben erklärt in der *Phrasentabelle* aufgezeichnet. Bei der tatsächlichen Anwendung wird dann die Hypothese mit dem besten Score für die Übersetzung gewählt. Beim statistischen Ansatz wird also zuerst der Quelltext vorverarbeitet. Anschließend kommt die Anwendung des *Übersetzungsmodells*, um den Text in die Zielsprache zu übersetzen, und dann die Anwendung des *Sprachmodells*, um die grammatikalischen Fehler zu beheben, da eine direkte Übersetzung der einzelnen Bestandteile des Texts häufig zu falschen Wortalinierungen in der Zielsprache führt. Außerdem dient sie dazu, zu prüfen, ob die gewählte Übersetzung tatsächlich im Gesamtsatz sinngemäß ist.

Mathematischer Hintergrund

Der mathematische Hintergrund ist das Bayes'sche Theorem. Dabei ist es das Ziel z die Wahrscheinlichkeit, dass die Übersetzung $e_1^I = e_1 e_2 .. e_I$, mit e_i jeweils ein Wort der Phrase aus der Zielsprache, die tatsächliche Übersetzung der Aussage $f_1^J = f_1 f_2 .. f_J$, mit f_j jeweils ein Wort der Phrase aus der Quellsprache, ist, zu maximieren (Koehn, 2009). Also:

$$z = \arg\max_{e_1^I} p(e_1^I | f_1^J) \tag{2.1}$$

$$= \arg\max_{e_1^I} p(e_1^I) * p(f_1^J | e_1^I) \tag{2.2}$$

Wobei $p(f_1^J|e_1^I)$ von dem *Übersetzungsmodell* und $p(e_1^I)$ von dem *Sprachmodell* geliefert wird. Für *Übersetzungsmodelle* werden, wie oben erläutert, häufig phrasenbasierte Modelle verwendet, da diese nach dem modernsten Stand der Technik derzeit die beste Performanz liefern. Dazu werden aus einem parallelen Korpus Phrasen extrahiert, deren Wahrscheinlichkeit berechnet wird, dass sie Übersetzungen von einander sind, und so eine *Phrasentabelle* erzeugt, in welcher die berechneten Phrasenzuordnungen und deren Wahrscheinlichkeiten aufgeführt werden.

Im Folgenden wird erläutert, wie mit Hilfe von maschinellen Übersetzungssystemen soziale Dialogsysteme implementiert werden können.

3 Soziales Dialogsystem basierend auf maschinellen Übersetzungssystemen

In diesem Kapitel werden Parallelen zwischen sozialen Dialogsystemen und phrasenbasierten statistischen Übersetzungssystemen gezogen (Ritter u. a., 2011). Soziale Dialogsysteme erhalten als Eingabe eine vom Nutzer beliebige Aussage. Die Menge solcher Aussagen wird im Folgenden als Aussagensprache bezeichnet. Von dem System hingegen wird als Ausgabe eine Antwort erwartet. Die Menge dieser Antworten wird als Antwortsprache bezeichnet. Diese Aussagen- und Antwortsprache entsprechen der Quell- und Zielsprache bei Übersetzungssystemen. Nehmen wir folgendes Beispiel:

Aussage A Wie geht es dir?

Antwort A Mir geht es gut.

In diesem Beispiel kann man sehr leicht die Bezüge zwischen den einzelnen Wörtern erkennen. Das „*mir*“ und „*gut*“ aus Antwort A beziehen sich jeweils auf „*dir*“ und „*wie*“ aus Aussage A. In diesem Beispiel kommt es sozusagen zu einer Übersetzung wie in 3.1 dargestellt. Die Quadrate zeigen hierbei die Wortalinierung an.

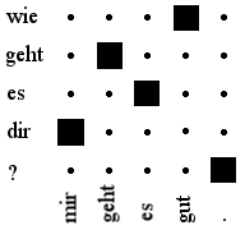

Abbildung 3.1: *Einfache Alinierung der Wörter.* Die Quadrate stellen die Alinierung der einzelnen Wörter zwischen Aussagesatz und Antwortsatz dar.

In dem vorangegangenen Beispiel sind die Zusammenhänge leicht ersichtlich. Wenn man jedoch Konversationen betrachtet, welche über solchen standardmäßigen Austausch hinausgehen, wird auffällig, dass die Alinierung, also die Zuordnung, von Worten bzw. auch kompletten Phrasen aus Aussagen- und Antwortsatz kein seltenes Phänomen ist. Häufig werden Parallelismen und Chiasmen in unserer konversationellen Sprache verwendet. Parallelismen sind Sätze, bei welchen die Syntax gleich aufgebaut ist. Chiasmen hingegen sind Sätze, bei welchen die einzelnen Teile des Satzes eine Überkreuzstellung feststellen lassen. Bei Abbildung 3.1 erkennt man eine Überkreuzstellung bei „*wie*“ - „*gut*“ und „*dir*“ - „*mir*“. Um ein untypischeres Beispiel zu nehmen, welches einer realen Facebook-Konversation entnommen wurde, und keinen allzu häufig vorkommenden Nachrichtenaustausch darstellt:

Aussage B Muss gerade an die Tortellini mit Tonnen Käsesoße denken. Die war gut.

Antwort B Alles mit viel Käse ist göttlich!

Dieses Beispiel lässt auf den ersten Blick nicht sofort rhetorische Stilmittel erkennen. Jedoch kann bei genauerem Hinsehen festgestellt werden, dass sich *alles* auf *die Tortellini*, „*mit viel Käse*“ auf „*mit Tonnen Käsesoße*“ und „*göttlich*“ auf „*Die war gut.*“ bezieht. Man erkennt, dass ein Parallelismus das Grundgerüst der Antwort ist.

Abbildung 3.2: *Mittelschwierige Alinierung der Wörter.* In diesem Beispiel ist die Alinierung der einzelnen Wörter zwischen Aussagesatz und Antwortsatz nicht auf Anhieb ersichtlich. Jedoch können bei genaueren Hinsehen Zusammenhänge zwischen Aussagesatz und Antwortsatz festgestellt werden.

Parallelismen finden sich häufig in unserer Sprache wieder. Sie dienen dazu Resonanz zu aktivieren (Sakita, 2006). Sprecher verwenden Parallelismen unbewusst. Dies liegt an der menschlichen Fähigkeit Muster zu erkennen und so sprachliche Elemente aufeinander abzubilden.
Ähnlich ist es bei Übersetzungen. Bei Übersetzungen zwischen verschiedenen Sprachen werden sehr häufig die Texte in der Zielsprache in einer parallelistischen Weise zu dem Originaltext aus der Quellsprache verfasst, wie am folgendem Beispiel erkenntlich.

Originaltext A Heute ist ein perfekter Tag um Rad zu fahren.

Übersetzung A Today is a perfect day to go cycling.

Nicht nur Parallelismen, sondern auch Chiasmen lassen sich bei Übersetzungen wiederfinden:

Originaltext B In den Bergen wohnt Heidi.

Übersetzung B Heidi lives in the mountains.

Parallelismen lassen sich offensichtlich wiederfinden, da es sich um direkte Übersetzungen handelt, bei denen man in der Regel Wort für Wort vorgeht. Jedoch entstehen aufgrund Beachtung unterschiedlicher grammatikalischer und syntaktischer Regeln auch Chiasmen bei Übersetzungen. Diese Ähnlichkeit zwischen der Übersetzung eines Textes in eine andere Sprache und der Generierung von Antworten zu einer beliebigen Aussage - oder um die Ähnlichkeit noch weiter hervorzuheben: Der Übersetzung eines Textes

aus der Aussagensprache in einen Text der Antwortsprache - verleitet zur Nutzung maschineller Übersetzungssysteme zur Generierung von Antworten für soziale Dialogsysteme (Ritter u. a., 2011).
Die Idee ist, dass das System diese Zusammenhänge zwischen einzelnen Phrasen erlernt, so wie ein Übersetzungssystem Zusammenhänge zwischen Phrasen aus verschiedenen Quell- und Zielsprachen erkennt, und die erlernten Zusammenhänge zur Übersetzung von unbekannten Eingabetexten anwendet. Dazu ist eine große Menge an Daten nötig, welche als ein paralleler Korpus zum Erlernen von Phrasenpaaren und deren Scores dienen kann. Aus einem parallelen Korpus soll folglich eine *Phrasentabelle* gebildet werden, welche anschließend zum Trainieren eines maschinellen Übersetzungssystems verwendet werden soll.
Um Parallelen zwischen der phrasenbasierten maschinellen Übersetzung und sozialen Dialogsystemen zu ziehen, kann man sagen, dass die *Phrasentabelle* den Regeln eines statistischen Dialogsystems entspricht. Das *Sprachmodell* des Übersetzungssystems entspricht der NLG. Beide Komponenten dienen dazu, dem Menschen eine textuell verständliche Repräsentation des Inhalts zu liefern.

Im Folgenden soll kurz auf die Daten für die hier beschriebene Methode eingegangen werden, anschließend auf die Extraktion der Phrasenpaare und zuletzt auf das Training.

3.1 Daten

Für das Experiment wird eine große Menge an Daten benötigt. Hierbei wurden hauptsächlich zwei Quellen verwendet, und zwar OpenSubtitles und Twitter. Aus beiden Quellen wurden englische Konversationsdaten extrahiert, welche zur Extraktion der *Phrasentabelle* und für die Testdaten verwendet wurden. Im Folgenden sollen die verwendeten Quellen kurz vorgestellt und ihr Potenzial erläutert werden.

3.1.1 OpenSubtitles

OpenSubtitles[1] bietet über ihre Server Untertitel für diverse Filme an. Diese Untertitel stellen mehr oder weniger realistische Konversationen dar, aus welchen man Aussage-Antwort-Paare extrahieren kann. Der Korpus von OpenSubtitles wurde für Testzwecke verwendet, um das Nutzen von maschinellen Übersetzungssystemen zu erproben. Bei dem verwendeten Korpus handelt es sich um einen Datensatz, welcher aus Aussage-Nachfrage-Paaren bestand. Also Aussagen, auf welche Fragen zur Klarifizierung der Aussage folgten.
Generell bietet OpenSubtitles eine große Menge an Konversationsdaten. Da es sich bei den Untertiteln zu einem Großteil um die Transkribierung des gesprochenen Dialogs der Akteure in den Filmen handelt und kaum weitere Elemente enthält, ist es möglich, über OpenSubtitles eine große Menge an Konversationsdaten zu sammeln. Text, welcher nicht Konversationsdaten darstellt, ist in der Regel durch eckige Klammern gekennzeichnet. Dabei kann es sich zum Beispiel um Geräuscheffekte handeln, wie zum Beispiel [*Thunder*], welche man leicht aus den Daten entfernen kann. Schwieriger ist die Zuordnung, ob zwei aufeinanderfolgende Aussagen tatsächlich eine Konversation darstellen. Die Untertitel beinhalten Zeitangaben, welche hierfür nützlich sind. Bei sehr langen Zeitdifferenzen kann man davon ausgehen, dass eine Aussage nicht auf die vorangegangene bezogen ist. Eine weitere Schwierigkeit, welche schwieriger zu beheben ist, ist, dass die Untertitel satzweise angegeben werden, sodass nicht klar ist, ob zwei aufeinanderfolgende Aussagen von dem gleichen Sprecher geäußert wurden. Hier gibt es ebenfalls die Möglichkeit auf Zeitunterschiede zu achten. Im Falle des gleichen Sprechers wird der Zeitabstand zwischen der ersten und der zweiten Äußerung wahrscheinlich geringer sein wie bei einem Sprecherwechsel, da ein Sprecher während dem Sprechen zwischen den Sätzen nur kurze Pausen einlegt. Bei natürlicher Sprache werden zwischen Sätzen häufig keine Pausen eingelegt, dafür jedoch innerhalb von Sätzen aufgrund von Zögern, Versprechern, o.ä.. (Shriberg, 2005). Dahingegen warten andere Sprecher kurzzeitig ab, um abzusehen, ob der vorangegangene Sprecher seine Äußerung vollständig beendet hat.
Ein weiterer Fall wäre, dass ein anderer Sprecher dem vorangegangenen Sprecher in dessen Wort fällt. Auch in diesem Fall sind die Zeitabstände zwischen den beiden Sätzen sehr kurz, jedoch wird der Abbruch des vorangegangenen Satzes durch einen Spiegelstrich „-“ gekennzeichnet.
Für den parallelen Korpus wurden wie oben erwähnt Aussage-Nachfrage-Paare verwendet. Das heißt, in den Quelldaten befindet sich eine Aussage und in den Zieldaten eine zugehörige Frage zur Klarifikation.

3.1.2 Twitter

Twitter[2], ein soziales Medium, bietet eine große Menge an öffentlich zugänglichen Konversationsdaten. Aufgrund der Struktur von Twitter, welche lediglich Nachrichten mit maximal 120 Zeichen erlaubt, ist es möglich einen Nachrichtenaustausch zu sammeln, welcher aus weniger Sätzen besteht. Dies wiederum bedeutet, dass die Sätze in der Regel im unmittelbarem Bezug zum vorangegangen Text, dem Aussagentext, stehen. Über die Twitter-API ist es möglich solche Aussage-Antwort-Paare automatisiert zu extrahieren. Für diese Arbeit wurden über 100.000 Aussage-Antwort-Paare extrahiert. Diese Aussage-Antwort-Paare mussten normalisiert werden, damit das System mit den Daten arbeiten konnte. Dazu wurden Labels für Usernamen, Hashtags und Hyperlinks eingeführt, wie am folgenden Beispiel dargestellt.

[1] `http://http://www.opensubtitles.org`
[2] `https://twitter.com/`

Vor Normalisierung Meet Diesel - slightly furrier than your average National Trust ranger... @ArlingtonRanger #dogsoftwitter http://t.co/r9dmNj6IKP

Nach Normalisierung Meet Diesel - slightly furrier than your average National Trust ranger . . . (user) (hashtag) (url)

Der Twitter-Corpus wurde als Hauptkorpus verwendet. Bei der Bereinigung der Daten wurde ein Datensatz mit diesen Labels erzeugt und ein weiterer Datensatz ohne das Label (user), da dessen Vorkommen sehr häufig ist, da bei Twitter jede Antwort mit einer Adressierung des Konversationspartners durch dessen Usernamen eingeleitet wird. Auf die Unterscheidung wird später bei der Evaluierung noch näher eingegangen.
Twitter bietet eine herausragende Menge an natürlichen Konversationsdaten. Mit derzeit über 316 Mio. Nutzern pro Monat und 500 Mio. Tweets pro Tag[3] bietet die soziale Plattform Konversationen unterschiedlichster Menschen über unschiedlichste Themen innerhalb kurzer Zeitspannen. Anders als bei OpenSubtitles handelt es sich hier um natürliche Konversationen, da die Nutzer ihre Tweets offensichtlich nicht eingeprobt haben, sondern aus Spontaneität schreiben. Das heißt, es ergibt sich ein großes Reservoir an natürlichen geschriebenen Konversationsdaten, die sich auch über diverse Themen erstrecken. Falls Bedarf nach Konversationsdaten über bestimmte Themen besteht, ist es möglich mithilfe der Suchfunktion, welche auch über die Twitter-API nutzbar ist, zielgerichtet Konversationsthemen zu konkreten Thematiken zu suchen. Außerdem bietet Twitter auch den Vorteil, dass es eine klare Abgrenzung zwischen dem ersten und zweiten Sprecher gibt, während es bei OpenSubtitles schwerer zu bestimmen ist, ob zwei aufeinanderfolgende Aussagen von unterschiedlichen Sprechern geäußert wurden.
Ein Nachteil an Twitter ist jedoch, dass dadurch, dass es ein soziales Medium ist, also keine Plattform, auf welcher orthographisch und grammatikalisch korrekter Sätze gefordert werden, ein viel höherer Bedarf an Normalisierung entsteht wie bei OpenSubtitles. Bei OpenSubtitles z.B. müsste vermutlich lediglich *„you're“* auf *„you are“* abgebildet werden. Dahingegen müssten bei Twitter Abkürzungen wie *„u're“*, *„u r“*, u.ä. oder sogar *„your“*, welches in einem anderen Kontext keine Korrektur benötigen würde, auf *„you are“* abgebildet werden. Außerdem erhöht sich die Anzahl an Fehlern in Orthographie und Grammatik auch dadurch, dass viele Nutzer nicht Muttersprachler der englischen Sprache sind, sich aber dennoch auf Englisch auf der sozialen Plattform mitteilen, um ein größeres Publikum zu erreichen.
Der parallele Korpus basierend auf den Twitterdaten ist so aufgebaut, dass in den Quelldaten der Inhalt eines Tweets ist und in den Zieldaten der Inhalt des Tweets, welches darauf geantwortet hat. Das bedeutet, dass der Korpus nicht satzweise, sondern aussagenweise aufgebaut it.

Wir haben somit einen parallelen Korpus basierend auf den OpenSubtitles-Daten und zwei parallele Korpora basierend auf Twitter-Daten. Bei einem der Twitter-Korpora ist das Label (user) enthalten beim anderen wurde dieser eliminiert.

[3] `https://about.twitter.com/company`

3.2 Phrasentabelle

Die parallelen Korpora basierend auf OpenSubtitles und Twitter wurden zur Extraktion von Phrasenpaaren verwendet. Dafür wurden drei Verfahren genutzt und zwar das Verfahren nach Koehn (Koehn, 2009), der *Fisher's Exact Test* (Fisher, 1925) und der *G-Test* (Sokal und Rohlf, 2009). Die Tabellen, die im folgenden dargestellt werden, wurden ausschließlich aus dem parallelen Twitter-Korpus ohne das Label (user) extrahiert.

3.2.1 Standard nach Koehn

Das Verfahren zur Erzeugung einer *Phrasentabelle* nach Koehn (Koehn, 2009) gliedert sich in drei Abschnitte. Der erste Schritt ist das Aufstellen einer Wortalinierung zwischen den Aussagen im parallelen Korpus. Der zweite Schritt ist die Extraktion der Phrasenpaare. Zuletzt müssen die Phrasenpaare einen Score erhalten.

Wortalinierung

Die Wortalinierung kann mittels verschiedener Methoden berechnet werden, wie zum Beispiel mit einem IBM Modell (Koehn u. a., 2003) oder HMM (Hidden Markov Model) (Vogel u. a., 1996). Das für diese Arbeit verwendete System nutzt IBM Modell 4 (Koehn u. a., 2003).
Nach der Berechnung der Wortalinierung müssen die wahrscheinlichsten Pfade berechnet werden. Dies geschieht mit Hilfe des Viterbi-Algorithmus, welcher die wahrscheinlichen Zustandssequenzen, also die Pfade, abschätzen kann (Forney Jr., 1973).
Da das Verfahren beachtet, dass Wortalinierung asymmetrisch sein kann, ist es nötig, die Wortalinierung auch in die umgekehrte Richtung, also nicht nur von Quell- zu Zielsprache, sondern auch von Ziel- zu Quellsprache zu trainieren. Dabei bedeutet asymmetrisch, dass die Wortalinierung zwischen Quell- und Zielsprache sich von der Wortalinierung zwischen Ziel- und Quellsprache unterscheidet (V Graça u. a., 2010). Anschließend kann man die berechneten Viterbipfade kombinieren. Jedoch muss dabei beachtet werden, dass verschiedene Kombinationsmöglichkeiten verschiedene Vor- und Nachteile mit sich bringen. Man kann hierbei z.B. die *Genauigkeit* (engl. *precision*) und die *Sensitivität* (engl. *recall*) betrachten (Powers, 2011). Ein Ergebnis mit hoher *Genauigkeit* bedeutet, dass man mit hoher Wahrscheinlichkeit relevante Daten auffindet. In unserem Fall wären relevante Daten korrekte Paare:

$$Genauigkeit = \frac{|\{\text{gefundene Paare}\} \cap \{\text{korrekte Paare}\}|}{|\{\text{gefundene Paare}\}|} \tag{3.1}$$

$$= \frac{|\{\text{korrekte, gefundene Paare}\}|}{|\{\text{korrekte, gefundene Paare}\} \cup \{\text{inkorrekte, gefundene Paare}\}|} \tag{3.2}$$

$$\hat{=} \frac{\text{true positive}}{\text{true positive + false positive}} \tag{3.3}$$

Ein Ergebnis mit hoher *Sensitivität* hingegen bedeutet, dass man möglichst alle relevanten Daten auffindet. In unserem Fall wären relevante Daten korrekte Paare:

$$Sensitivität = \frac{|\{\text{gefundene Paare}\} \cap \{\text{korrekte Paare}\}|}{|\{\text{korrekte Paare}\}|} \tag{3.4}$$

$$= \frac{|\{\text{korrekte, gefundene Paare}\}|}{|\{\text{korrekte, gefundene Paare}\} \cup \{\text{korrekte, nicht-gefundene Paare}\}|} \tag{3.5}$$

$$\hat{=} \frac{\text{true positive}}{\text{true positive + false negative}} \tag{3.6}$$

Eine Möglichkeit wäre, die Pfade über ihre Schnittmenge zu kombinieren. In diesem Fall wird die *Genauigkeit* hoch sein, die *Sensitivität* jedoch gering. Eine weitere Möglichkeit wäre es, die Pfade über eine Vereinigung zu kombinieren. Dies würde wiederum zu dem umgekehrten Fall führen, dass die *Genauigkeit* gering, die *Sensitivität* dafür hoch ist. Um beide Verhältnisse zu befriedigen, wurden Heuristiken entwickelt. Das verwendete System hat *Grow-Diag-Final-And* (Koehn, 2009) verwendet. Im Folgenden wird von einer tabellenartigen Anordnung von Quell- und Zielsatz wie in Abbildung 3.1 gesprochen.

Bei *Grow-Diag* gehört die Schnittmenge der Pfade zur Gesamtalinierung. Es werden die Nachbarn der einzelnen Wortalinierungen betrachtet, welche selbst zur Gesamtalinierung gehören. Falls der Nachbar eine potenzielle Wortalinierung ist und in dessen Reihe bzw. Spalte noch keine Wortalinierung enthalten ist, welche zur Gesamtalinierung gehört, dann wird dieser Nachbar in die Gesamtalinierung aufgenommen. Dabei sind potenzielle Wortalinierungen solche, welche ein Element aus der Vereinigung der Viterbi-Pfade sind.

Grow-Diag

generiere *Vereinigung* und *Schnitt*
$a = \{\}$
$a' = Schnitt$
while $a \neq a'$ **do**
  $a = a'$
  for all $p \in a$ **do**
    for all $p' \in nachbar(p)$ **do**
      if $p' \in Vereinigung$ **then**
        if Reihe oder Spalte enthalten noch keine Alinierung **then**
          $a' = a' \cup \{p'\}$
        end if
      end if
    end for
  end for
end while

Final-And hingegen fügt all diejenigen einzelnen, potenziellen Wortalinierungen hinzu, bei welchen sowohl die Reihe als auch die Spalte noch nicht durch eine Wortalinierung aus der Gesamtalinierung abgedeckt sind.

Final-And

generiere *Vereinigung*
$a = derzeitigesAlignment$
for all $p \in Vereinigung$ **do**
  if Reihe und Spalte enthalten noch keine Alinierung **then**
    $a = a \cup \{p\}$
  end if
end for

Extraktion der Phrasenpaare

Für die Extraktion der Phrasenpaare muss gelten, dass sie konsistent zur Gesamtalinierung A sind. Eine Phrase $(\bar{e}, \bar{f})$ ist konsistent, wenn Folgendes gilt :

$$\forall e_i \in \bar{e} : (e_i, f_j) \in A \rightarrow f_j \in \bar{f} \tag{3.7}$$

$$\wedge \forall f_j \in \bar{f} : (e_i, f_j) \in A \rightarrow e_i \in \bar{e} \tag{3.8}$$

$$\wedge \exists e_i \in \bar{e}, f_j \in \bar{f} : (e_i, f_j) \in A \tag{3.9}$$

Anhand von Abbildung 3.2 kann man dies erläutern. 3.7 bedeutet, wenn eine Alinierung aus einer Zeile Teil des Phrasenpaares ist, dann müssen alle Zielworte aus den Alinierungen der gleichen Zeile Teil der Zielphrase sein. Umgekehrt bedeutet 3.8, dass das gleiche analog für alle Spalten gilt. D.h. wenn *„göttlich“* Teil der Zielphrase ist, dann müssen *„die“*, *„war“* und *„gut“* Teil der Quellphrase sein. Außerdem muss nach 3.9 für jedes Wort aus den Phrase gelten, dass auch eine zugehörige Alinierung aus der Gesamtalinierung existiert.
Es ist sehr speicheraufwendig, alle möglichen Phrasenpaare zu speichern. Daher werden nur die kürzeren, bei welchen die Zahl der Worte pro Phrase also geringer ist, und häufigsten Phrasenpaare ausgewählt, um Speicherplatz zu sparen.

Kalkulierung der Scores

Die Scores werden über folgende Formeln berechnet:

Relative Häufigkeit

$$p(\bar{f}|\bar{e}) = \frac{count(\bar{f},\bar{e})}{\sum_{f} count(\bar{f},\bar{e})} \tag{3.10}$$

$$p(\bar{e}|\bar{f}) = \frac{count(\bar{f},\bar{e})}{\sum_{e} count(\bar{f},\bar{e})} \tag{3.11}$$

Lexikalische Gewichtung

$$p(\bar{f}|\bar{e},a) = \prod_{j=1}^{J} \frac{1}{|\{i|(i,j) \in a\}|} \sum_{(i,j)\in a} Pr(f_j|e_i) \tag{3.12}$$

$$p(\bar{e}|\bar{f},a) = \prod_{i=1}^{I} \frac{1}{|\{j|(i,j) \in a\}|} \sum_{(i,j)\in a} Pr(e_i|f_j) \tag{3.13}$$

Die relative Häufigkeit 3.10 gibt an, wie häufig ein Phrasenpaar in allen Sätzen vorkommt. 3.11 zeigt die Formel für die relative Häufigkeit, wobei die Übersetzung umgekehrt, also von Ziel zur Quelle, erfolgt. Die lexikalische Gewichtung 3.12 hingegen gibt die Wahrscheinlichkeit eines Phrasenpaares Wort für Wort an (Koehn u. a., 2003). Analog ist 3.13 das lexikalische Gewicht für die umgekehrte Übersetzung. Beide Werte werden sowohl für die Übersetzung von Quellsprache zu Zielsprache, also Aussagensprache zu Antwortsprache, als auch für die Übersetzung von Zielsprache zu Quellsprache berechnet.
Tabelle 3.1 ist ein Auszug aus der *Phrasentabelle*.

Quellphrase	**Zielphrase**	**Rel. Häufigkeit Q-Z**	**Lex. Gewichtung Q-Z**	**Rel. Häufigkeit Z-Q**	**Lex. Gewichtung Z-Q**
seeing	you're a	0.001	0.002	0.001	0.000
seeing	you're	0.000	0.002	0.001	0.008
seems pretty	Enter your student ID	0.004	0.000	0.001	0.000
seems pretty	Enter your student	0.004	0.000	0.001	0.000
seems pretty	Enter your	0.004	0.000	0.001	0.000
seems pretty	Enter	0.004	0.000	0.001	0.005
seems pretty	me Enter your student	0.004	0.000	0.001	0.000
seems pretty	me Enter your	0.004	0.000	0.001	0.000
seems pretty	me Enter	0.004	0.000	0.001	0.000
seems pretty	offer tells me Enter	0.004	0.000	0.001	0.000
seems pretty	tells me Enter your	0.004	0.000	0.001	0.000
seems pretty	tells me Enter	0.004	0.000	0.001	0.000
seems to have	. Maybe	0.000	0.000	0.002	0.000
seems to have	.	0.000	0.000	0.064	0.1310
seems to have	70 .	0.004	0.000	0.002	0.000
seems to have	Hever 70 .	0.004	0.000	0.002	0.000
seems to have	Stone story . Maybe	0.006	0.000	0.002	0.000
seems to have	Stone story .	0.006	0.000	0.002	0.000
seems to have	Time Stone story . Maybe	0.006	0.000	0.002	0.000
seems to have	Time Stone story .	0.006	0.000	0.002	0.000

Tabelle 3.1: *Auszug aus der* Phrasentabelle *für Twitter-Daten.* Q-Z bedeutet von Quelle nach Ziel und Z-Q von Ziel nach Quelle.

3.2.2 Fisher's Exact Test

Es wurde auch eine *Phrasentabelle* mit Hilfe von *Fisher's Exact Test* (Fisher, 1925; Ritter u. a., 2011) erzeugt. Dafür wurde eine Liste mit allen möglichen Phrasenpaaren benötigt (Ritter u. a., 2011), wobei ein mögliches Phrasenpaar, aus je einer Phrase des Aussage- und Antwortsatzes besteht. Die Phrasen sind je ein bis vier Wörter lang.

Aufwandsschätzung der Vorbereitung

Alle möglichen Phrasenpaare aus dem parallelen Korpus zu extrahieren würde bereits eine große Menge an Daten bedeuten, da es für jeden Satz circa $4n$ Möglichkeiten gibt, wobei n die Anzahl der Wörter im Satz ist. Jede Phrase aus dem Aussagesatz muss mit allen möglichen Phrasen aus dem dazugehörigen Antwortsatz kombiniert werden, was demnach zu circa $4n_S * 4n_R = 16(n_S * n_R)$ Möglichkeiten führt, wobei n_S die Anzahl der Wörter im Aussagesatz und n_R die Anzahl der Wörter im Antwortsatz ist. Sei n_M die durchschnittliche Anzahl der Wörter in den Sätzen des parallelen Korpus und x die Anzahl der parallelen Sätze im Korpus. Dies würde bedeuten, dass es circa $x * 16n_M^2$ mögliche Phrasenpaare gibt. Auf den ersten Blick erscheint der Extraktionsaufwand gering, da er in $O(n_M^2)$ liegt, jedoch werden durch die Landau-Notation die Koeffizienten nicht beachtet. Bei über 100.000 parallelen Sätzen, also über $1.600.000n_M^2$ möglichen Phrasenpaaren, ergibt sich insgesamt ein erheblicher Speicherplatzverbrauch und auch ein hoher Rechenaufwand diese Phrasenpaare zu extrahieren.

Umsetzung der Vorbereitung

Da die Kapazitäten für solch einen Aufwand nicht gegeben waren, musste bereits bei der Extraktion der möglichen Paare eine Filterung durchgeführt werden. Dazu wurden zunächst Phrasen aus den Aussagesätzen extrahiert, deren Häufigkeit des Vorkommens in allen Aussagesätzen eine konstante Schwelle überschritt. Dabei ist mit Vorkommen gemeint, in wie vielen Sätzen eine Phrase vorkommt, d.h. wenn eine Phrase in einem Satz mehrmals vorkommt, wird dies lediglich als ein Vorkommen gezählt. Wir nennen die Menge dieser Phrasen, welche eine bestimmte Häufigkeit überschreitet, K_I. Auf die gleiche Art und Weise wurden auch Phrasen aus den Antwortsätzen extrahiert. Die Menge dieser Phrasen heiße K_O. Anschließend wurden die möglichen Phrasenpaare so extrahiert, dass die Eingabephrase eine Phrase aus K_I und die Ausgabephrase eine Phrase aus K_O ist. Dadurch wurde die Zahl der möglichen Phrasenpaare eingeschränkt. Da Phrasen, welche allgemein selten im Korpus vorkommen, nicht in K_I bzw. K_O vorkommen, werden in die Liste der möglichen Phrasenpaare keine Paare aufgenommen, welche ohnehin sehr selten vorkommen.
Neben der Phrase aus der Aussagensprache und der Phrase aus der Antwortsprache wurden auch die Häufigkeit des Vorkommens jedes Phrasenpaares, die Häufigkeit der Phrase aus der Aussagensprache und die Häufigkeit der Phrase aus der Antwortsprache für jedes Phrasenpaar aufgezeichnet, um eine Kontingenztabelle zu erhalten. Mit dieser Kontingenztabelle lässt sich der *Fisher's Exact Test* durchführen. Man erkennt hier bereits, dass viele Phrasen auf sich selbst abgebildet werden, wie beispielsweise „.“ auf „.“ mit einer Häufigkeit von 26705. In 3.2 werden die häufigsten Phrasen im Aussagekorpus und in 3.3 werden häufige Phrasenpaare, die nicht auf sich selbst abbilden aufgelistet.

Phrase	Vorkommen im Aussagekorpus
.	26705
the	22061
to	20049
I	18729
you	17299
,	16489
a	16389

Tabelle 3.2: *Häufigsten Phrasen im Aussagekorpus.*

Aussagephrase	Antwortphrase	Vorkommen im Korpus
the	for	3586
you	I	3575
a	you	3299
and	,	3253
a	and	3044
and	I	2967
no	no	2893
that	.	2857
is	to	2834
of	a	2831

Tabelle 3.3: *Häufige Phrasen im Korpus.*

Durchführung des Fisher's Exact Tests

Der *Fisher's Exact Test* (Fisher, 1925) ist ein statistischer Signifikanztest, welcher die Wahrscheinlichkeit berechnet, dass eine beobachtete Kontingenztabelle den tatsächlichen Wahrscheinlichkeiten in der Realität entspricht. Im Folgenden sei $C(s,t)$ die Häufigkeit des Vorkommens eines Phrasenpaares im parallelen Korpus. $C(s)$ sei die Häufigkeit des Vorkommens einer Phrase im Korpus der Aussagesätze und $C(t)$ die Häufigkeit einer Phrase im Korpus der Antwortsätze. Eine Linie über s bzw. t steht für die Negation. Beispielsweise steht $C(s,\bar{t})$ für die Anzahl der Satzpaare, bei welchem s im Aussagesatz, aber t nicht im zugehörigen Antwortsatz vorkommt. Unsere Kontingenztabellen haben folglich eine solche Form:

$C(s,t)$	$C(s,\bar{t})$	$C(s)$
$C(\bar{s},t)$	$C(\bar{s},\bar{t})$	$C(\bar{s})$
$C(t)$	$C(\bar{t})$	N

Tabelle 3.4: *Aufbau der Kontingenztabelle.* $C(s,t)$ ist die Häufigkeit des Vorkommens von dem Phrasenpaar (s,t) im parallelen Korpus. $C(\bar{s},\bar{t})$ hingegen ist die Anzahl der Sätze, in welchen das Phrasenpaar (s,t) nicht vorkommt und N die Gesamtzahl aller parallelen Sätze.

1	199	200
143	100085	100228
144	100284	100428

Tabelle 3.5: *Kontingenztabelle des einmalig vorkommenden Phrasenpaares (reason, website). „reason"* im Aussagesatz und *„website"* im dazugehörigen Antwortsatz kommt lediglich einmal vor.

1006	1750	2756
16293	81379	97672
17299	83129	100428

Tabelle 3.6: *Kontingenztabelle des relativ häufig vorkommenden Phrasenpaares (know, you).*

26705	0	26705
0	73723	73723
26705	73723	100428

Tabelle 3.7: *Kontingenztabelle des am häufigsten vorkommenden Phrasenpaares (., .).*

Die Tabellen 3.5, 3.6 und 3.7 stellen Kontingenztabellen für ein seltenes, ein häufiges und das häufigste Phrasenpaar dar. Der *Fisher's Exact Test* berechnet mithilfe der hypergeometrischen probabilistischen Verteilung die Abhängigkeit zweier Parameter.:

$$p_h(C(s,t)) = \frac{\binom{C(s)}{C(s,t)}\binom{C(\bar{s})}{C(\bar{s},t)}}{\binom{N}{C(t)}} \tag{3.14}$$

Das bedeutet in unserem Fall, wenn der errechnete Wert gering ausfällt, ist es unwahrscheinlich, dass die Antwortphrase tatsächlich die Übersetzung der Aussagenphrase ist, da eine niedrige Korrelation durch den *Fisher's Exact Test* errechnet wurde (Johnson u. a., 2007). Auf der anderen Seite bedeutet eine hohe Wahrscheinlichkeit, dass die Antwortphrase sehr wahrscheinlich tatsächlich die Übersetzung der Aussagenphrase ist. Der *Fisher's Exact Test* ist dafür bekannt, dass er auch für einen relativ kleinen Stichprobenumfang gute Ergebnisse liefert.
Der *Fisher's Exact Test* wurde für alle Phrasenpaare durchgeführt. In Tabelle 3.8 sind beispielhaft einige Werte für Twitter-Daten aufgeführt.

Aussagephrase	Antwortphrase	Ergebnis
reason	website	0.216
to the	last	0.048
. How	know	0.150
1	Just	0.260
Why	make	0.017
am a	follow	0.161
or	trip	0.235
Your	work	0.001
had a	very	0.079
you were	you!	0.136
thank you	to see you	0.039
i	running	0.153
. I have	perfect	0.177
Wow	, just	0.177
called	. My	0.277
everything	I had	0.360
on the	indeed	0.262
voice	wait	0.069
works	well .	0.168
bet	is what	0.142

Tabelle 3.8: *Auszug aus* Phrasentabelle *extrahiert mit Hilfe des* Fisher's Exact Tests.

3.2.3 G-Test

Um einen weiteren Vergleich zu haben, wurde auch der *G-Test* (Sokal und Rohlf, 2009) auf dieselben Daten angewandt. Für den *G-Test* werden ebenfalls Kontingenztabellen verwendet, welche analog zu den Kontingenztabellen des *Fisher's Exact Tests* berechnet werden. Für das Experiment wurden die Kontingenztabellen des *Fisher's Exact Tests* wiederverwendet.

Durchführung des G-Tests

Anders als *Fisher's Exact Test* ist der *G-Test* kein exakter Signifikanztest, sondern ein Likelihood Ratio Test. Likelihood Ratio Test bedeutet, dass man das Verhältnis (Ratio) zwischen den beobachteten und den erwarteten Ergebnissen betrachtet. Die erwarteten Ergebnisse werden unter der Null-Hypothese angenommen. Das bedeutet, man erwartet, dass die Parameter unabhängig voneinander sind. Die beobachteten und erwarteten Ergebnisse stellen jeweils Wahrscheinlichkeiten (Likelihoods) dar und deren Verhältnis gibt Auskunft darüber, ob die Wahrscheinlichkeiten übereinstimmen. Der bekannte statistische Test χ^2-Test (Chi-Square-Test) approximiert ein solches Verhältnis. Jedoch gilt der neuere *G-Test* mittlerweile als die bessere Alternative, da die Berechnung effizienter[4] ist (Sokal und Rohlf, 2009). Daher wurde der *G-Test* zur Extraktion einer *Phrasentabelle* ausgewählt.

Für den *G-Test* werden folglich zum einen die beobachteten Werte und zum anderen die erwarteten Werte benötigt. Die beobachteten Werte entsprechen der Kontingenztabelle. Die erwarteten Werte können anhand der beobachteten Ergebnisse berechnet werden. Im Folgenden beschreiben wir dies anhand unseres Beispiels aus Tabelle 3.6.
Man nimmt an, die Wahrscheinlichkeit, dass die Phrase s im Aussagesatz vorkommt ist $\frac{2756}{100428}$, also der Quotient aus der Häufigkeit des Vorkommens und der Gesamtzahl aller Sätze. Analog dazu ist die Wahrscheinlichkeit, dass die Phrase t im Antwortsatz vorkommt, $\frac{17299}{100428}$. Unter Annahme der Null-Hypothese gilt:

$$p(\text{„Vorkommen von s“} \cap \text{„Vorkommen von t“}) = p(\text{„Vorkommen von s“}) * p(\text{„Vorkommen von t“}) \tag{3.15}$$

Das bedeutet, die erwartete Häufigkeit, dass s in einem Aussagesatz und t in dem dazugehörigen Antwortsatz vorkommt, ist $p(\text{„Vorkommen von s“}) * p(\text{„Vorkommen von t“}) * n = \frac{2756}{100428} * \frac{17299}{100428} * 100428 \approx 474,7$. Analog dazu werden die erwarteten Häufigkeiten von $(s,\bar{t})$, $(\bar{s},t)$ und $(\bar{s},\bar{t})$ berechnet und ergeben die Kontingenztabelle mit den erwarteten Häufigkeiten wie in Tabelle 3.10 dargestellt. Man kann an den Beispielen 3.9 und 3.11 bereits Tendenzen erkennen. Bei 3.9 sieht man, dass das Phrasenpaar lediglich einmal im parallelen Korpus vor kam, was bereits die Vermutung nahelegt, dass *„reason“* und *„website“* sehr wahrscheinlich unabhängig voneinander sind. Dementsprechend weicht die Kontingenztabelle mit den erwarteten Werten unter der Null-Hypothese, also dass die Parameter unabhängig sind, kaum von der beobachteten Tabelle ab. Dahingegen sieht es bei 3.11 genau umgekehrt aus. Bei Betrachtung der beobachteten Werte, kann man von einer starken Abhängigkeit ausgehen, also dass „.“ mit einer sehr hohen Wahrscheinlichkeit in „.“ übersetzt wird. Dementsprechend weicht die Kontingenztabelle der beobachteten Werte sehr stark von der Kontingenztabelle der erwarteten Werte unter der Null-Hypothese ab. Man erkennt hier einen guten Kontrast zu 3.9.

[4] Dies liegt mitunter daran, dass heutzutage Logarithmen in der Praxis leicht berechenbar sind.

0,3	199,7	200
143,7	100084,3	100228
144	100284	100428

Tabelle 3.9: *Kontingenztabelle mit erwarteten Häufigkeit des Vorkommens des Phrasenpaares (reason, website).* Die Tabelle ähnelt der Tabelle der beobachteten Werte 3.5 stark.

474,7	2281,3	2756
16824,3	79145,7	97672
17299	83129	100428

Tabelle 3.10: *Kontingenztabelle mit erwarteten Häufigkeit des Vorkommens des Phrasenpaares (know, you).*

7101,2	19603,8	26705
19603,8	54119,2	73723
26705	73723	100428

Tabelle 3.11: *Kontingenztabelle mit erwarteten Häufigkeit des Vorkommens des Phrasenpaares (., .).* Die Tabelle unterscheidet sich stark von den beobachteten Werten 3.7

Mit Hilfe der beobachteten und erwarteten Werte lässt sich nun der *G-Test* ausführen mit folgender Formel (Sokal und Rohlf, 2009):

$$G = 2 * \sum_{i \in \{s,\bar{s}\}} \sum_{j \in \{t,\bar{t}\}} O_{ij} * ln(\frac{O_{ij}}{E_{ij}}) \tag{3.16}$$

Dabei ist O_{ij} der entsprechende beobachtete Wert und E_{ij} der entsprechende erwartete Wert in der Kontingenztabelle. Tabelle 3.12 ist ein Auszug aus der *Phrasentabelle* berechnet mit Hilfe des *G-Tests*.

Aussagephrase	Antwortphrase	Ergebnis
reason	website	0.299
to the	last	0.185
. How	know	0.914
1	Just	0.761
Why	make	0.042
am a	follow	0.302
or	trip	0.340
Your	work	0.001
had a	very	0.229
you were	you!	0.239
thank you	to see you	0.072
i	running	0.708
. I have	perfect	0.223
Wow	, just	0.225
called	. My	0.446
everything	I had	0.824
on the	indeed	0.776
voice	wait	0.122
works	well .	0.208
bet	is what	0.166

Tabelle 3.12: *Auszug aus* Phrasentabelle *extrahiert mit Hilfe des* G-Tests.

3.3 Training

Wir verfügen nun über unterschiedliche *Phrasentabellen*. Das *Sprachmodell* wird über das maschinelle Übersetzungssystem nach Koehn (2009) berechnet. Der nächste Schritt wäre das Decoding, bei welchen mit Hilfe der Komponenten eine Übersetzung berechnet wird (Koehn, 2009). Für das Decoding werden jedoch gewisse Parameter benötigt, und zwar die Gewichte für das Log-Lineare Modell. Das Log-Lineare Modell leitet sich aus der Formel für die globale Suche ab indem man den einzelnen Komponenten Gewichte λ_i zuteilt, um ihren Einfluss auf die Übersetzung zu ändern:

$$\arg\max p(e)^{\lambda_1} * p(f|e)^{\lambda_2} \tag{3.17}$$

$$= \arg\max \exp(\lambda_1 \log(p(e)) + \lambda_2 \log(p(f|e))) \tag{3.18}$$

Diese Gewichte müssen so gewählt werden, dass die Übersetzung zu einem optimalen Ergebnis führt. Das bedeutet in unserem Fall, dass bestimmt werden muss, wie stark das *Sprach-* und das *Übersetzungsmodell* jeweils die Übersetzung beeinflussen dürfen, so dass die generierte Antwort passend zu der dazugehörigen Aussage ist.

Parameter Tuning

Es gibt Algorithmen, um die Wahl der Gewichte automatisch zu optimieren. Solch eine automatische Konfiguration wurde auf dem OpenSubtitles-Korpus verwendet (Koehn, 2009), was auf den ersten Blick zu keinen herausragenden Ergebnissen führte. Darauf wird näher in der Evaluation 4.2 eingegangen. Daher wurden für das Training mit dem Twitter-Korpus verschiedene Gewichte für das *Sprach-* und *Übersetzungsmodell* getestet. Das System wurde manuell mit verschiedenen Gewichtungen für das Log-Lineare Modell konfiguriert und ausgeführt. Die Resultate werden in den Sektionen 4.2.1, 4.2.2 und 4.2.3 ausgewertet. Die gewählten Gewichte werden in Tabelle 3.13 dargestellt.

λ des *Sprachmodells*	λ des *Übersetzungsmodells*
0,1	0,9
0,0	1,0
0,5	0,5
1,0	0,1
0,2	-0,8

Tabelle 3.13: *Verwendeten Gewichte für die Übersetzung*

4 Evaluation

Die berechneten Übersetzungen wurden mit Hilfe einer Benutzerstudie evaluiert. Für soziale Dialogsysteme gestaltet sich eine automatische Evaluation schwierig, da man anders als bei der maschinellen Übersetzung, bei welcher mit automatisierten Verfahren wie BLEU (Papineni u. a., 2002) evaluiert werden kann, nicht den generierten Antwortsatz mit dem originalen Antwortsatz vergleichen kann. Bei der maschinellen Übersetzung werden die Sätze in der Zielsprache der originalen Übersetzung inhaltlich gleichen, daher ist die Wahrscheinlichkeit, dass die automatische Übersetzung der menschlichen Übersetzung gleicht, höher. Bei sozialen Dialogen hingegen, kann auf eine Aussage mit vielen unterschiedlichen Varianten geantwortet werden, wie im Folgenden dargestellt.

Aussage Heute ist ein schöner Tag.

Mögliche Übersetzung 1 Today is a nice day.

Mögliche Übersetzung 2 Nice weather today! (lose Übersetzung)

Mögliche Übersetzung 3 Today is a beautiful day.

Mögliche Übersetzung 4 Today it is nice.

Mögliche Antwort 1 Findest du?

Mögliche Antwort 2 Es soll heute Abend noch regnen.

Mögliche Antwort 3 Gott sei Dank.

Mögliche Antwort 4 Aber trotzdem steckst du immer in deinem Zimmer und gehst nie raus!

Man sieht, dass die möglichen Übersetzungen sich ähneln, während die möglichen Antworten sich stark voneinander unterscheiden können. Daher wurde für das Experiment eine menschliche, subjektive Evaluation gewählt.

4.1 Aufbau der Evaluation

Die Benutzerstudie wurde mit Hilfe von LimeSurvey[1] erstellt. Jedem Teilnehmer wurden insgesamt 16 Aussagen präsentiert, bei welchen der Teilnehmer die unterschiedlichen, möglichen Antworten hinsichtlich der Eignung als Antwort bewertet. Jedes der Antworten repräsentierte ein anderes Extraktionsverfahren bzw. das Original. In Abbildung 4.1 ist beispielhaft der Aufbau einer Frage dargestellt. Insgesamt haben 16 Teilnehmer alle möglichen Antworten bewertet.
Es gab sechs Fragen zu den Ergebnissen basierend auf dem OpenSubtitles-Korpus und zehn Fragen zu den Ergebnissen basierend auf dem Twitter-Korpus. Der Twitter-Korpus, welches das Label (user) beinhaltet, führte zu Übersetzungen, bei welchen alle dem System bekannten Worte auf das Label (user) abgebildet wurde. Dies liegt daran, dass bei Twitter jede Antwort mit einer User-Adressierung eingeleitet wird. Dadurch ergab sich eine hohe Wahrscheinlichkeit für die Übersetzung jeglicher Worte auf das Label (user). Aus diesem Grund wurde für die Evaluation der Twitter-Korpus verwendet, bei welchem dieses Label entfernt wurde.
Die Fragen wurden so aufgebaut, dass zuerst eine zufällige Aussage aus den Testdaten ausgewählt wurde.

[1] https://www.limesurvey.org/

Anschließend wurden die unterschiedlichen Antworten, welche durch das maschinelle Übersetzungssystem generiert wurden, als Antwortmöglichkeiten hinzugefügt. Für jede Antwort konnte eine Bewertung auf der Skala 1 bis 5 getroffen werden, wobei 1 bedeutet, dass die Antwort unpassend und 5 bedeutet, dass die Antwort sehr passend ist.
Für die Ergebnisse auf Basis des OpenSubtitles-Korpus wurde die Gewichtung der verschiedenen Modelle automatisch optimiert. Bei den Übersetzungen basierend auf dem Twitter-Korpus hingegen wurden wie im Absatz 3.3 zu Parameter Tuning erläutert verschiedene Gewichte verwendet. Bei den Fragen zum Twitter-Korpus hatte jede Gewichte-Kombination jeweils zwei Fragen.

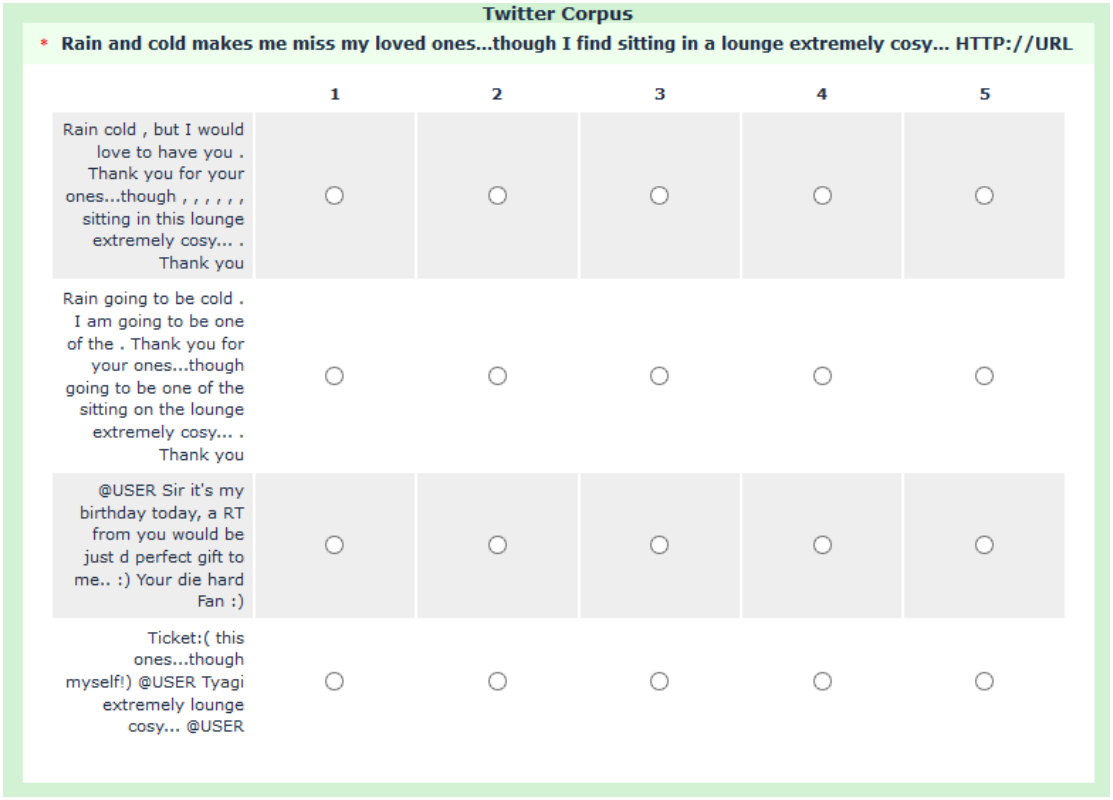

Abbildung 4.1: *Aufbau einer Frage in der Benutzerstudie.*

4.2 Ergebnisse der Evaluation

Zu beachten ist, dass die Studie lediglich Tendenzen aufweist und keine klaren, statistisch signifikanten Rückschlüsse erlaubt. Dies ist zurückzuführen auf die relativ kleine Frage- und Teilnehmeranzahl. In Tabelle 4.1 wird dem Betrachter schnell ersichtlich, dass die Bewertungen nicht zufriedenstellend schlecht ausfallen. Was jedoch ebenfalls auffällt, ist, dass selbst die originalen, von einem Menschen verfassten Antworten auf die Aussagen nur häufig mittelmäßig mit einem Score knapp über 3 bewertet wurden. Dies lässt vermuten, dass die Teilnehmer der Benutzerstudie womöglich dazu tendierten schlechte Bewertungen abzugeben. Dazu kommt, dass es zu relativ starken Schwankungen zwischen den Entscheidungen der Teilnehmer kam, da es durchschnittlich eine Standardabweichung von ca. 1,022 gab. Die höchste Standardabweichung von 1,58 wurde bei folgendem Aussage-Antwort-Paar, welche mit dem *G-Test* generiert wurde, erreicht:

Aussage @USER Many congrats Matt. You know where I am if I can offer any help or advice. Keep up the good work.

Antwort @USER Many thanks Matt. #HASHTAG #HASHTAG ... #HASHTAG offer #HASHTAG #HASHTAG ... #HASHTAG advice. . #HASHTAG #HASHTAG ... #HASHTAG work.

Ungefähr die Hälfte der Teilnehmer hat das oben gezeigte Paar mit einer 1 bewertet. Dies legt die Vermutung nahe, dass die generierte Antwort sehr unpassend ist, jedoch hat knapp ein Viertel der Teilnehmer dieses Paar auch mit einer 5 bewertet, was unerwartet ist, wenn ein Großteil der Teilnehmer dieses Paar derart schlecht bewertet hat. Dies zeigt auf, dass die Meinung bei der Bewertung der generierten Antworten stark auseinandergehen kann.
Was ebenfalls auffällt, ist, dass die unten gezeigte Übersetzung mit Hilfe der nach Koehns Standard generierten *Phrasentabelle* bei einem Aussage-Antwort-Paar einen Score von 3,44 erreicht hat. Dies ist der höchste Wert für eine generierte Antwort basierend auf dem Twitter-Korpus. Zudem ist dieser Score sogar höher als der Score für den originalen Satz.

Aussage @USER Just getting ready for hospital.

Antwort Original @USER nearly home give you a ring in 5 minutes x

Antwort Koehn @USER mmhmmm hospital.

Man erkennt, dass die generierte Antwort relativ generisch ist, während die originale Antwort spezifischer ist. Das *"mmhmmm"* könnte als ein Laut der Abneigung sein, was im Bezug auf Krankenhäuser nicht ungewöhnlich wäre. Dahingegen ist der Bezug zwischen einem Anruf (*„ring“*) und einem Krankenhaus nicht sofort gegeben. Hier hat also das System eine bessere Antwort gegeben als ein Mensch, jedoch ist dies offensichtlich eine Ausnahme.
Außerdem muss beachtet werden, dass die Menge der Daten mit über 100.000 Zeilen an parallen Sätzen nicht ausreichend waren, um herausragende Ergebnisse zu gewährleisten. Man bemerkt bei der Übersetzung häufig, dass Wörter nicht bekannt waren (Out-of-Vocabulary, kurz OOV).

$\lambda_L M$	$\lambda_T M$	Fisher's Exact Test	G-Test	Koehn	Original
0,1	0,9	2,13	1,44	1,13	3,25
0,1	0,9	2,38	1,44	1,5	3,0
0,0	1,0	1,69	1,75	3,44	3,25
0,0	1,0	2,19	1,56	1,63	3,13
0,2	-0,8	1,5	1,44	2,63	4,5
0,2	-0,8	2,0	1,5	1,75	4,13
0,5	0,5	2,0	2,25	1,56	3,38
0,5	0,5	2,13	2,13	1,88	3,44
1,0	0,1	2,38	1,63	1,19	3,44
1,0	0,1	1,31	2,31	1,56	4,38
Durchschnitt		1,971	1,745	1,827	3,59

Tabelle 4.1: *Bewertungen der Sätze generiert auf Basis des Twitterkorpus, sowie Gewichtungen des* Sprach- *(λ_{LM}) und* Übersetzungsmodells *(λ_{TM})*

Die Übersetzungen wurden zuerst mit den Daten basierend auf dem OpenSubtitles-Korpus generiert. Jedoch führten die Ergebnisse zu relativ kurzen Sätzen. Da es sich um ein Aussage-Nachfrage-Korpus handelt, waren die generierten Antworten hauptsächlich Fragen, welche zum größten Teil sehr kurz formuliert wurden. Zumeist wurden Worte auf Fragewörter wie *„what“* abgebildet. Kurze fragende Rückmeldungen kommen zwar in der Realität vor, jedoch ist gewünscht, dass die Antworten komplexer und empathischer wirken. Daher wurden bei den Übersetzungen basierend auf den Twitter-Daten die Gewichte manuell konfiguriert und dies soll nun evaluiert werden.

4.2.1 Gewichtung für Standard nach Koehn

Die Benutzerstudie erlaubt Vermutungen über die geeignete Wahl der Gewichtungen für die verschiedenen Verfahren aufzustellen. Im folgenden sei die Gewichtung des *Sprachmodells* $\lambda_L M$ und die Gewichtung des *Übersetzungsmodells* $\lambda_T M$. Anhand von Abbildung 4.2 kann man bei Koehn auf den ersten Blick vermuten, dass die Gewichte $\lambda_L M = 0{,}0$ und $\lambda_T M = 0{,}1$ geeignet wären. Jedoch muss man wie bereits erwähnt beachten, dass bei einem der Antworten der Score von 3,44 sehr ungewöhnlich erscheint und daher vermutlich nicht repräsentativ ist. Dazu kommt, dass die zweite Antwort, welche mit der gleichen Gewichtung generiert wurde, einen vergleichsweise niedrigen Score hat. Dies legt nahe, dass die einzelne gute Bewertung nicht mit der Gewichtung zusammenhängt. Stattdessen wären $\lambda_L M = 0{,}2$ und $\lambda_T M =$ -0,8 bzw. $\lambda_L M = 0{,}5$ und $\lambda_T M = 0{,}5$ wahrscheinlich geeignetere Gewichte, da die Differenz der Bewertungen zwischen beiden Sätzen jeweils nicht sehr groß ist.

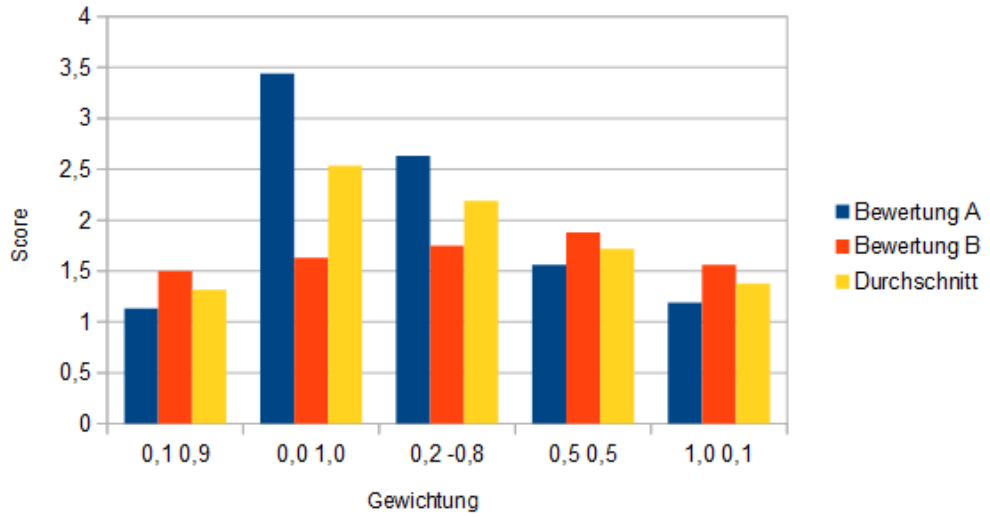

Abbildung 4.2: *Durchschnittliche Bewertung der Antworten basierend auf* Phrasentabelle *generiert durch Koehns Standard.*

4.2.2 Gewichtung für Fisher's Exact Test

Man sieht in Abbildung 4.3, dass mit den Gewichten 0,1 und $\lambda_T M = 0{,}9$ durschnittlich die höchste Bewertung erreicht wurde. Dazu ist die Differenz zwischen den beiden Bewertungen relativ gering, was vermuten lässt, dass es keine irreguläre, besonders gute Bewertung gab. Auch hat die Gewichtung $\lambda_L M = 0{,}5$ und $\lambda_T M = 0{,}5$ einen vergleichsweisen hohen Durchschnitt, jedoch sind die Bewertungen beider Antworten generiert mit den Gewichten $\lambda_L M = 0{,}1$ und $\lambda_T M = 0{,}9$ größer gleich der Bewertungen der Antworten generiert mit den Gewichtungen $\lambda_L M = 0{,}5$ und $\lambda_T M = 0{,}5$. Es zeigt sich also eine leichte Tendenz zur Nutzung der Gewichtung $\lambda_L M = 0{,}1$ und $\lambda_T M = 0{,}9$ im Bezug auf den *Fisher's Exact Test*. Auch kann es zu der Vermutung führen, dass allgemein eine höhere Gewichtung des *Übersetzungsmodells* vorteilhaft ist, da die Gewichtung des *Übersetzungsmodells* mit $\lambda_T M = 0{,}9$, $\lambda_T M = 1$ und $\lambda_T M = 0{,}5$ durchschnittlich zu höheren Bewertungen geführt hat wie bei einer niedrigen Gewichtung mit $\lambda_T M =$ -0,8 oder $\lambda_T M = 0{,}1$.

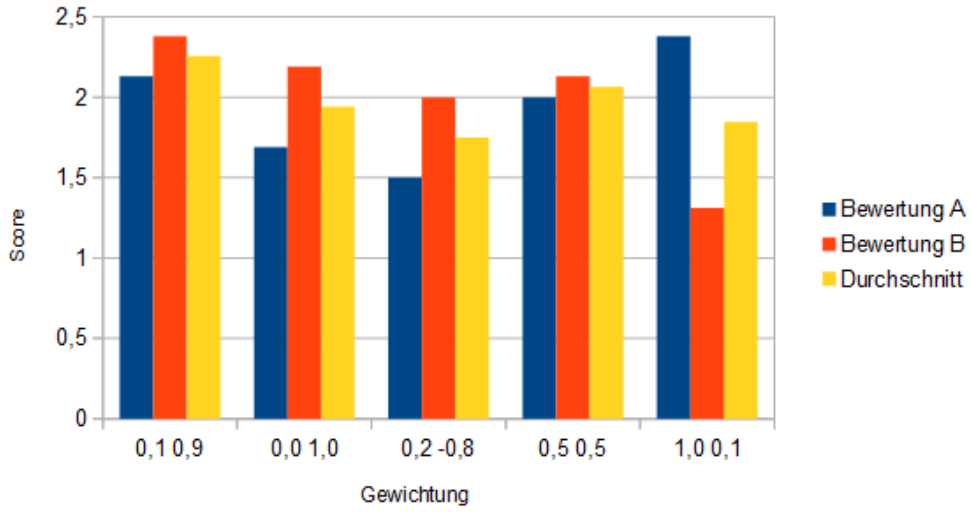

Abbildung 4.3: *Durchschnittliche Bewertung der Antworten basierend auf* Phrasentabelle *generiert durch* Fisher's Exact Test. Für jede Gewichtung wurden jeweils zwei Sätze in der Benutzerstudie abgefragt. Diese werden hier A und B genannt.

4.2.3 Gewichtung für G-Test

Beim *G-Test* kann man in Abbildung 4.4 erkennen, dass es eine Tendenz zur Gewichtung $\lambda_L M = 0{,}5$ und $\lambda_T M = 0{,}5$ gibt. Die Antworten, welche durch diese Gewichtung generiert wurden, haben auch eine geringere Differenz bei den Bewertungen. Es käme zwar auch die Gewichtung $\lambda_L M = 1{,}0$ und $\lambda_T M = 0{,}1$ in Frage, jedoch ist die Differenz bei dieser Gewichtung deutlich höher wie bei der Gewichtung $\lambda_L M = 0{,}5$ und $\lambda_T M = 0{,}5$. Daher ist eine ausgeglichene Gewichtung zwischen *Sprachmodell* und *Übersetzungsmodell* für den *G-Test* vermutlich vorteilhafter.

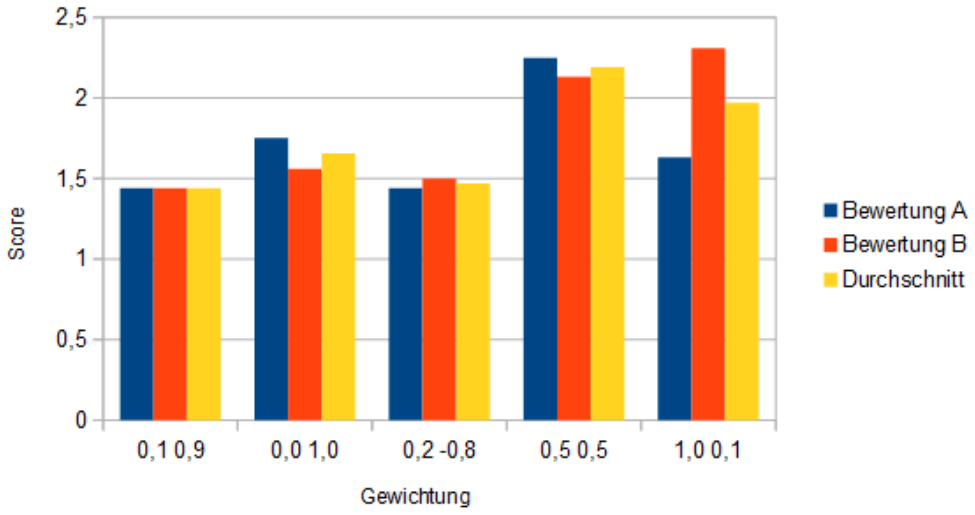

Abbildung 4.4: *Durchschnittliche Bewertung der Antworten basierend auf* Phrasentabelle *generiert durch* G-Test.

4.2.4 Vergleich zwischen den Verfahren

Es ist außerdem möglich einen Vergleich zwischen den drei gewählten Verfahren zu ziehen. Man kann anhand von Abbildung 4.5 erkennen, dass der *Fisher's Exact Test* im Durchschnitt am besten bewertet wird. Dazu gibt es beim *Fisher's Exact Test* kaum mehr Schwankungen zwischen den Bewertungen als beim *G-Test*. Koehns Verfahren weist vergleichsweise sehr hohe Schwankungen auf, wodurch dieser schwer mit den anderen Verfahren zu vergleichen ist. Wenn man jedoch die außergewöhnlich hohe Bewertung der bereits zuvor erwähnten Antwort außer Acht lässt, weist der *G-Test* einen höheren Durchschnitt auf.
Neben den Ergebnissen basierend auf den Twitter-Daten kann hier auch das Ergebnis basierend auf den OpenSubtitles-Daten hinzugezogen werden 4.2. Auch bei der Bewertung der Antworten generiert auf Basis des OpenSubtitles-Korpus hat der *Fisher's Exact Test* einen besseren Durchschnitt als der *G-Test* und der Standard nach Koehn. Jedoch weißt die Bewertung des *G-Tests* hier deutliche Schwankungen auf, weswegen man nicht allein aufgrund des Durchschnitts behaupten kann, dass der *G-Test* besser als der Standard nach Koehn abschneidet. Man kann jedoch an den Ergebnissen sehen, dass der *G-Test* Ähnlichkeiten zum *Fisher's Exact Test* aufweist. Wenn *Fisher's Exact Test* eine Bewertung von ≥ 3 aufweist, trifft dies auch auf den *G-Test* zu. Wenn *Fisher's Exact Test* eine Bewertung von < 3 hat, hat der *G-Test* eine schlechte Bewertung von < 2. Dies ist jedoch lediglich bei den Ergebnissen basierend auf dem OpenSubtitles-Korpus zu erkennen, da die Bewertungen der Antworten basierend auf dem Twitter-Korpus generell schlecht ausfallen. Diese schlechtere Bewertung lässt sich darauf zurückführen, dass die Aussagen basierend auf dem Twitter-Korpus deutlich komplexer sind als die Aussagen basierend auf

dem OpenSubtitles-Korpus. Jedenfalls lassen die Ergebnisse bzgl. der Aussage-Antwort-Paare basierend auf dem OpenSubtitles-Korpus vermuten, dass der *G-Test* zu ähnlichen Ergebnissen führt.
Man könnte also folgern, dass der *Fisher's Exact Test* am besten und das Verfahren nach Koehn am schlechtesten für unsere Zwecke, der Generierung sozialer Antworten, geeignet ist. Jedoch ist diese Auswertung aufgrund der insgesamt durchschnittlich schlechten Bewertung lediglich eine Tendenz.

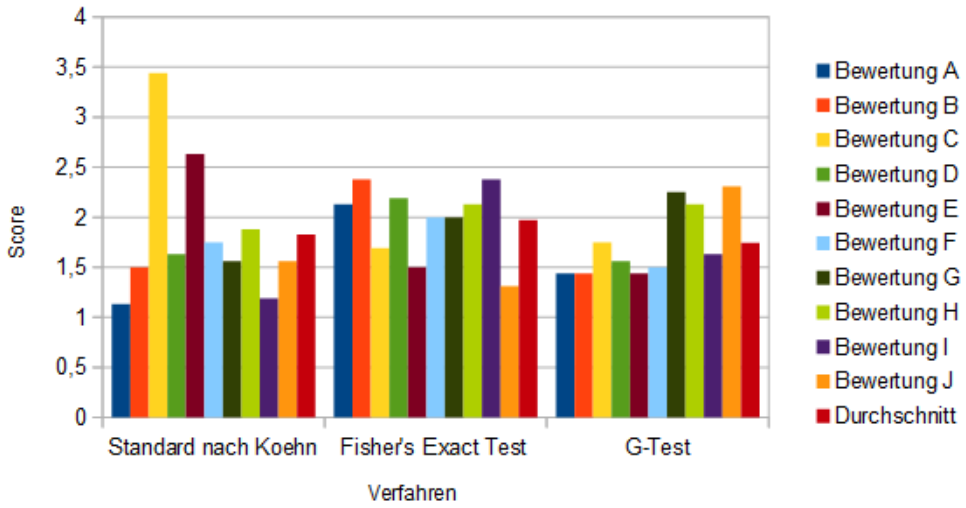

Abbildung 4.5: *Durchschnittliche Bewertung der Antworten basierend auf Verfahren.* A bis J sind Bezeichnungen für die einzelnen Antworten.

	Fisher's Exact Test	G-Test	Koehn	Original
	1,13	1,19	1,44	2,38
	3,31	3,0	2,5	1,88
	3,94	4,31	2,63	3,25
	2,19	1,5	2,19	4,13
	2,44	1,88	2,13	3,44
	4,31	3,5	3,5	4,19
Durchschnitt	2,88	2,56	2,398	3,212

Tabelle 4.2: *Bewertungen der Sätze generiert auf Basis des OpenSubtitles-Korpus*

5 Fazit und Ausblick

Im Rahmen dieser Arbeit wurden aus den Ressourcen OpenSubtitles und Twitter anhand verschiedener Phrasenextraktionsverfahren *Phrasentabellen* extrahiert. Diese Verfahren waren das Extraktionsverfahren nach Koehn, welches den heutigen Standard in der maschinellen Übersetzung darstellt, der *Fisher's Exact Test* und der *G-Test*. Diese *Phrasentabellen* stellen Übersetzungen von Phrasen aus der Aussagensprache in die Antwortsprache dar. Mit diesen *Phrasentabellen* wurden maschinelle Übersetzungssysteme trainiert, wobei die Optimierung der Gewichtung der einzelnen Modelle wie *Sprach-* und *Übersetzungsmodell* bei den OpenSubtitles-Daten automatisiert wurde. Dahingegen wurden die Gewichte bei den Übersetzungen auf Basis der Twitter-Daten manuell bestimmt.
Die Idee der Umsetzung von sozialen Dialogsystemen mit Hilfe von maschineller Übersetzung (Ritter u. a., 2011) hat Potenzial, ist aber noch nicht genügend ausgefeilt. Relativ selten erhalten automatisch generierte Antworten eine bessere Bewertung als die originale menschliche Antwort. Es gibt verschiedene Möglichkeiten die Ergebnisse zu verbessern. Unter anderem hat die Datenseltenheit, wie in der Evaluation dargestellt, sehr wahrscheinlich zu einem schlechten Ergebnis geführt. Daher wäre eine noch weitaus höhere Menge an parallelen Daten ein Ansatz zur Verbesserung. Außerdem können die Twitterdaten auch noch besser aufbereitet werden (Kaufmann, 2010), indem eine intelligentere Normalisierung der Daten eingeführt wird. In unserem Versuch wurden Hashtags beispielsweise durch ein Label ersetzt. Jedoch können Hashtags je nach Tweet syntaktisch zum vollständigen Satz gehören. *„#Today is a good day!“* wäre ein Besipiel, wo ein Hashtag syntaktisch relevant ist. Kaufmann (2010) stellt dar, wie man mit Hilfe von maschineller Übersetzung Twitter-Daten normalisieren kann. Dabei wird die Umgangssprache auf Twitter als Eingabesprache und die korrekte englische Sprache als Zielsprache verstanden. Das System liefert bereits gute Ergebnisse bzgl. der Erkennung, ob Hashtags oder auch Nutzeradressierungen mittels „@“ eine syntaktische Relevanz haben.
Auf der Basis einer größeren und saubereren Datenmenge können die Phrasen Extraktionen mit Hilfe von *Fisher's Exact Test* und *G-Test* erneut ausgeführt werden. Da die Ergebnisse dazu tendieren den *Fisher's Exact Test* zu favorisieren, sollte dessen Nutzung weiter ausgebaut werden. Die bisherigen Ergebnisse wiesen sehr häufig OOV auf, wodurch keine angemessene Übersetzung möglich war. Mit einer größeren Datenmenge würde die OOV-Rate sinken, wodurch die Übersetzungsqualität womöglich steigen würde. Ein Problem, welches dadurch jedoch auftreten würde, wäre der Speicherplatzbedarf der *Phrasentabellen*, welche bei noch größeren Mengen an Daten enorm werden kann. Es gibt jedoch Möglichkeiten die Zahl der Phrasenpaare zu reduzieren, ohne dass die Übersetzungsqualität signifikant vermindert wird. Jedoch sind diese Techniken für maschinelle Übersetzung zwischen zwei Sprachen konzipiert und nicht zur Antwortgenerierung, weshalb geprüft werden müsste, ob es auch zu keiner signifikanten Minderung der Übersetzungsqualität bei einer Antwortgenerierung kommen würde. Sanchis-Trilles u. a. (2011) hat in seiner Arbeit eine Technik entwickelt, welche unabhängig von dem gewählten Extraktionsverfahren die Menge der Phrasenpaare reduzieren kann. Phrasenpaare, welche basierend auf einer quell-getriebenen Segmentierung als wahrscheinlich gelten, werden für die *Phrasentabelle* ausgewählt.
Da der *G-Test* zwar schlechter als *Fisher's Exact Test*, aber besser als Koehns Verfahren abschnitt, sollte man diesen ebenfalls mit einer größeren Datenmenge weiter erproben, da man nicht ausschließen kann, dass der *G-Test* zu guten Ergebnissen führen könnte.
Für das Training können basierend auf den Gewichtungen $\lambda_L M = 0{,}1$ und $\lambda_T M = 0{,}9$ für *Fisher's Exact Test* und $\lambda_L M = 0{,}5$ und $\lambda_T M = 0{,}5$ für den *G-Test* noch weitere feinere Gewichtsanpassungen ausgetestet werden, um die Übersetzung zu optimieren.
Die Evaluation kann in Zukunft eventuell auch mit Hilfe von Amazons Automatic Turks (Snow u. a., 2008) erfolgen, wodurch mehr Daten evaluiert werden, was bei einer lokalen menschlichen Evaluation schwer möglich ist. Außerdem tritt so auch nicht das Problem einer kleinen Teilnehmerquote ein.

Literaturverzeichnis

[Burnett 2004] BURNETT, Bruce M.: Technophiles and Technophobes? In: *New questions for contemporary teachers: taking a socio-cultural approach to education* (2004) 1

[Fisher 1925] FISHER, Ronald A.: *Statistical Methods for Research Workers*. 1925 14, 18, 19

[Forney Jr. 1973] FORNEY JR., G. D.: The viterbi algorithm. In: *Proceedings of the IEEE* 61 (1973), Nr. 3, S. 268–278 14

[Hampton u. a. 2009] HAMPTON, Keith ; GOULET, Lauren S. ; HER, Eun J. ; RAINIE, Lee: *Social Isolation and New Technology - How the internet and mobile phones impact Americans' social networks*. 2009 1

[Johnson u. a. 2007] JOHNSON, J H. ; MARTIN, Joel ; FOSTER, George ; FOSTER, George ; KUHN, Roland: *Improving Translation Quality by Discarding Most of the Phrasetable*, 2007 20

[Kaufmann 2010] KAUFMANN, Max: *Syntactic normalization of Twitter messages*. 2010 35

[Koehn 2005] KOEHN, Philipp: Europarl: A parallel corpus for statistical machine translation. In: *MT summit* Bd. 5 Citeseer (Veranst.), 2005, S. 79–86 7

[Koehn 2009] KOEHN, Philipp: *Statistical Machine Translation*. Cambridge University Press, 2009 6, 7, 14, 15, 25

[Koehn u. a. 2003] KOEHN, Philipp ; OCH, Franz J. ; MARCU, Daniel: Statistical phrase-based translation. In: *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1* Association for Computational Linguistics (Veranst.), 2003, S. 48–54 14, 16

[McTear 2002] MCTEAR, Michael F.: *Spoken Dialogue Technology: Enabling the Conversational User Interface*. 2002 1, 3

[Nagata und Morimoto 1994] NAGATA, Masaaki ; MORIMOTO, Tsuyoshi: First steps towards statistical modeling of dialogue to predict the speech act type of the next utterance. In: *Speech Communication* 15 (1994), Nr. 3, S. 193–203 5

[Papineni u. a. 2002] PAPINENI, Kishore ; ROUKOS, Salim ; WARD, Todd ; ZHU, Wei jing: BLEU: a Method for Automatic Evaluation of Machine Translation, 2002, S. 311–318 27

[Potthast u. a. 2008] POTTHAST, Martin ; STEIN, Benno ; ANDERKA, Maik: A Wikipedia-based multilingual retrieval model. In: *Advances in Information Retrieval*. Springer, 2008, S. 522–530 7

[Powers 2011] POWERS, David M.: Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation. (2011) 14

[Ritter u. a. 2011] RITTER, Alan ; CHERRY, Colin ; DOLAN, William B.: *Data-Driven Response Generation in Social Media*. 2011 2, 9, 11, 18, 35

[Robertson und Robertson 2012] ROBERTSON, Suzanne ; ROBERTSON, James: *Mastering the Requirements Process, Third Edition: Getting Requirements Right*. 2012 1

[Sakita 2006] SAKITA, Tomoko I.: Parallelism in conversation: Resonance, schematization, and extension from the perspective of dialogic syntax and cognitive linguistics. In: *Pragmatics & Cognition* 14 (2006), Nr. 3, S. 467–500 10

[Sanchis-Trilles u. a. 2011] SANCHIS-TRILLES, Germán ; ORTIZ-MARTINEZ, Daniel ; GONZÁLEZ-RUBIO, Jesús ; GONZÁLEZ, Jorge ; CASACUBERTA, Francisco: Bilingual segmentation for phrasetable pruning in statistical machine translation. In: *Proceedings of the 15th Conference of the European Association for Machine Translation*, 2011, S. 257–264 35

[Shriberg 2005] SHRIBERG, Elizabeth: Spontaneous speech: how people really talk and why engineers should care. In: *INTERSPEECH*, 2005, S. 1781–1784 12

[Snow u. a. 2008] SNOW, Rion ; O'CONNOR, Brendan ; JURAFSKY, Daniel ; NG, Andrew Y.: *Cheap and Fast — But is it Good?: Evaluating Non-Expert Annotations for Natural Language Tasks*. 2008 35

[Sokal und Rohlf 2009] SOKAL, Robert R. ; ROHLF, F. J.: *Introduction to Biostatistics - Second Edition*. Dover Publications, INC. Mineola, New York, 2009 14, 22, 23

[V Graça u. a. 2010] V GRAÇA, João ; GANCHEV, Kuzman ; TASKAR, Ben: Learning tractable word alignment models with complex constraints. In: *Computational Linguistics* 36 (2010), Nr. 3, S. 481–504 14

[Vogel u. a. 1996] VOGEL, Stephan ; NEY, Hermann ; TILLMANN, Christoph: HMM-based word alignment in statistical translation. In: *Proceedings of the 16th conference on Computational linguistics-Volume 2* Association for Computational Linguistics (Veranst.), 1996, S. 836–841 14

[Weizenbaum 1966] WEIZENBAUM, Joseph: ELIZA—a computer program for the study of natural language communication between man and machine. In: *Communications of the ACM* 9 (1966), Nr. 1, S. 36–45 1, 5