Forschungsseminar Deep Learning

WS 2017/2018

NLP-MACHINE TRANSLATION

LEARNING PHRASE REPRESENTATIONS USING RNN ENCODER–DECODER FOR STATISTICAL MACHINE TRANSLATION

Yupeng Guo

Betreuerin: Ying-Chi Lin

INSTITUT FÜR INFORMATIK | UNIVERSITÄT LEIPZIG

Inhaltsverzeichnis

1. Einleitung 2
1.1 Thema Beschreibung 2
1.2 Die Richtung dieses Dokuments. 3
1.3 Inhaltsanleitung 3
2. RNN Encoder-Decoder 4
2.1 Recurrent Neural Networks 4
2.2 RNN Encoder–Decoder 6
2.3 Hidden Unit 7
3. Statistical Machine Translation 9
3.1 Definition und Beispiele für SMT 9
3.2 Scoring Phrase Pairs mit RNN Encoder-Decoder 10
4. Experimente 11
4.1 Daten- und Baseline-System 11
4.2 Quantitative Analyse 13
4.3 Qualitative Analyse 14
4.4 Word and Phrase Representations 15
5. Zusammenfassung & Ausblick 18
Literaturverzeichnis 19

1. Einleitung

Dieses Dokument beschreibt, was Rekurrentes neuronales Netz (Abkürzung: RNN) ist, was RNN Encoder-Decoder ist, und was Statistical machine translation (Abkürzung: SMT) ist. Und nach dem Hinzufügen von RNN Encoder-Decoder zu SMT, hatte SMT eine „Qualität“ und „Quantität“ Optimierung.

1.1 Thema Beschreibung

Mit der Entwicklung der Globalisierung wird die Welt immer enger verbunden. Angesichts der rasanten Entwicklung von Technologie und Wirtschaft stellt die Vernetzung in der Welt einen unwiderstehlichen Entwicklungstrend dar. Wie können verschiedene Länder effektiv und kostengünstig miteinander kommunizieren? Die Kosten für manuelle Übersetzungen sind enorm und es sind nicht die optimale Lösung für die meisten Situationen. Die beste Lösung besteht vielleicht darin, die maschinelle Übersetzungstechnologie in vollem Umfang zu nutzen, um intelligente automatische Übersetzungsdienste bereitzustellen.

Die Idee, eine Maschine für die Übersetzung zu verwenden, wurde erstmals 1949 von Warren Weaver vorgeschlagen. Für eine lange Zeit (von den 1950er bis in die 1980er Jahre), die maschinelle Übersetzung wird durchgeführt, indem die linguistischen Informationen der Ausgangssprache und der Zielsprache studiert werden, dh basierend auf dem Wörterbuch und der Grammatik, um die Übersetzung zu erzeugen. Dies wird als „Rule-based machine translation“ (Abkürzung: RBMT) bezeichnet. Mit der Entwicklung der Statistik begannen die Forscher, statistische Modelle auf die maschinelle Übersetzung anzuwenden, die auf der Analyse des zweisprachigen Korpus basiert, um Übersetzungsergebnisse zu generieren. Diese Methode heißt „Statistical Machine Translation“. Es schnitt besser ab als RBMT und es dominierte dieses Gebiet von den 1980er bis 2000er Jahren. Die Komplexität der natürlichen Sprache ist jedoch bekannt, die Menschen werden immer noch missverstanden, wie übersetzt die Maschine? Wird die Maschine denken? Im Jahr 1997 schlugen Ramon Neco und Mikel Forcada die Idee der maschinellen Übersetzung unter Verwendung der

"Encoder-Decoder" -Architektur vor. Ein paar Jahre später, 2003, entwickelte ein Forscherteam um Yoshua Bengio von der Universität von Montreal ein Sprachmodell, das auf neuronalen Netzen basierte, um die traditionelle SMT-Modelle zu verbessern. Ihre Forschung legte den Grundstein für die zukünftige Anwendung von neuronalen Netzen in der maschinellen Übersetzung. Genauso wie in den frühen Jahren Google Translate für Übersetzungen verwenden, insbesondere in Spezialgebieten, ist die Übersetzung immer unzufrieden. Aber in den letzten Jahren wurde die Übersetzung jedoch genauer, nachdem Google angekündigt hatte, Google Translate mit neuronalen Netzwerken auszustatten.

Im Folgenden wird die Realisierung des neuronalen Netzwerks in SMT vorgestellt. Die Hauptreferenz zu diesem Dokument ist (Kyunghyun Cho et al., 2014)

1.2 Die Richtung dieses Dokuments.

Im Folgenden wird die Realisierung des neuronalen Netzwerks in SMT vorgestellt. Die Hauptreferenz zu diesem Dokument ist (Kyunghyun Cho et al., 2014)

1.3 Inhaltsanleitung.

Dieses Dokument konzentriert sich auf die Verwendung neuronaler Netzwerke für SMT mit einer neuartigen neuronalen Netzwerkarchitektur, die als Teil des herkömmlichen phrasenbasierten SMT-Systems verwendet werden kann. Zusätzlich eine ziemlich ausgeklügelte versteckte Einheit, um sowohl die Speicherkapazität als auch die Leichtigkeit des Trainings zu verbessern.

In den Experimenten, der vorgeschlagene RNN Encoder-Decoder mit einer neuartigen hidden unit wird empirisch auf die Aufgabe der Übersetzung von Englisch nach Französisch bewertet. Das Modell wurde trainiert, um die Übersetzungswahrscheinlichkeit eines englischen Satzes zu einem entsprechenden französischen Satz zu lernen. Das Modell wird dann als Teil eines phrase-basierten Standard-SMT-Systems verwendet, indem jedes Phrasenpaar in der Phrasentabelle bewertet wird. Die Auswertung zeigt, dass dieser Ansatz zum Bewerten von Phrasenpaaren mit einem RNN Encoder-Decoder die Übersetzungsleistung verbessert.

In der Analysephase führten sie qualitative und quantitative Analysen durch. Die qualitative Analyse zeigt, dass die RNN Encoder– Decoder die sprachlichen Regelmäßigkeiten in der Phrasentabelle besser erfassen kann, es erklärt indirekt auch die quantitativen Verbesserungen der gesamten Übersetzungsleistung. Die quantitative Analyse zeigt auch, dass die RNN Encoder– Decoder eine kontinuierliche Raumdarstellung einer Phrase lernt, die sowohl die semantische als auch die syntaktische Struktur der Phrase beibehält.

2. RNN Encoder-Decoder

Im nächsten Teil werde ich zuerst vorstellen, was RNN und RNNEE ist. In (Kyunghyun Cho et al., 2014) schlagen sie ein neuartiges neuronales Netzwerkmodell mit der Bezeichnung RNN Encoder-Decoder vor, das aus zwei rekurrenten neuronalen Netzen (RNN) besteht: Ein RNN kodiert eine Folge von Symbolen in eine Vektordarstellung fester Länge und die andere dekodiert die Darstellung in eine andere Symbolfolge.

In Abbildung 1 ist schematische Darstellung einer einfachen Struktur. Es besteht aus zwei rekurrenten neuronalen Netzen (RNN), Encoder und Decoder. Der „Encoder" bildet eine „variable-length source sequence" auf einen „fixed-length vector" ab, und der „Decoder" bildet die Vektordarstellung zurück auf eine „variable-length target sequence" ab. Die beiden Netzwerke werden gemeinsam trainiert, um die bedingte Wahrscheinlichkeit der Zielsequenz bei einer Quellensequenz zu maximieren. Zusätzlich verwenden sie eine komplexe hidden unit, um sowohl die Speicherkapazität als auch die Leichtigkeit des Trainings zu verbessern.

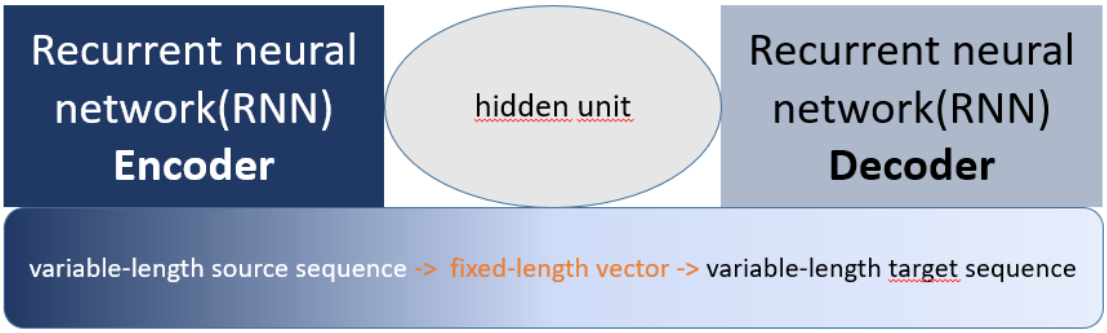

Abbildung 1: schematische Darstellung einer einfachen Struktur

2.1 Recurrent Neural Networks

Ein rekurrentes neuronales Netzwerk (RNN) ist ein neuronales Netzwerk, das aus einem Eingang x, einem versteckten Zustand h und einem Ausgang y besteht. $\vec{\mathrm{x}} = (x_1, \ldots, x_t)$ ist die Eingabesequenz, $h_{<t>} = \mathrm{f}(h_{<t-1>}, x_t)$ ist ein versteckter Zustand, der im Laufe der Zeit aktualisiert wird. Abbildung 2 zeigt seine Grundstruktur. Abbildung 3 zeigt seine hierarchische Struktur.

Wenn eine neue Sequenz in die Gleichung eingegeben wird, wird der vorherige Zustand $\overrightarrow{h_{<t-1>}}$ in $\overrightarrow{h_{<t>}}$ umgewandelt, der der aktuellen Eingabe x_t zugeordnet ist. Je früher die Eingabesequenz ist, desto kleiner ist das Gewicht im aktualisierten Zustand, was die zeitliche Korrelation zeigt. Wobei f eine nichtlineare Aktivierungsfunktion ist. f kann so einfach sein wie eine elementare logistische Sigmoidfunktion ($\tanh$) und so komplex wie eine Long-Term-Memory-Einheit (LSTM) (Hochreiter und Schmidhuber, 1997).

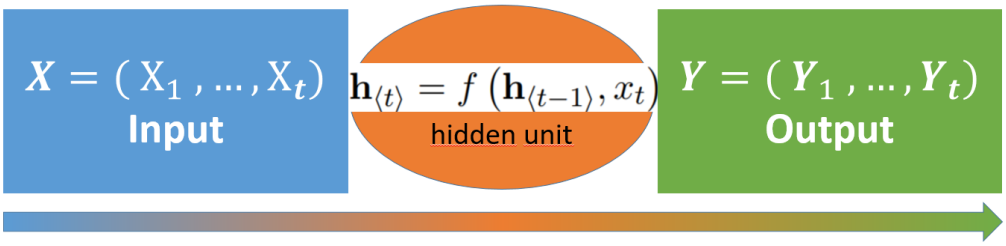

Abbildung 2: schematische Darstellung von ein RNN

Mit der versteckten Zustandsfolge kann das nächste Auftreten des Wortes vorhergesagt werden. In der Formel $\mathrm{p}(y_t) = \mathrm{p}(y_t \mid y_{t-1}, \ldots, y_1)$, y_t ist die Ausgabe an der t-ten Position und ihre Wahrscheinlichkeit $\mathrm{p}(y_t)$ basiert auf allen zuvor ausgegebenen Wörtern. Diese Wahrscheinlichkeit $\mathrm{p}(y_t)$ kann durch Verwendung des versteckten Zustands berechnet werden. Seine Formulierung ist: $\mathrm{p}(y_t) = \mathrm{g}(y_{t-1}, \overrightarrow{h_{<t>}}, \vec{c})$. Wobei $\vec{c}$ ist der Code aller versteckten Zustände, der immer alle versteckten Zustände enthält, wie den einfachen letzten versteckten Zustand $\overrightarrow{h_{<t>}}$ oder den Ausgang $\mathrm{f}(h_t, \ldots, h_1)$ einer nichtlinearen Gleichung. Da der versteckte Zustand t alle Eingangsinformationen vor dem t-ten Eingang codiert, impliziert y_t auch iterativ alle vorherigen Ausgangsinformationen, so dass dieses

Wahrscheinlichkeitsberechnungsverfahren vernünftig ist. Die nichtlineare Gleichung $g(y, h, c)$ kann hier ein komplexes Feedforward-Neuralnetzwerk sein, oder sie kann eine einfache nichtlineare Gleichung sein. Abbildung 3 ist eine gute Illustration der Beziehung zwischen $y_1, \ldots, y_{t-1}$ und y_t und der Informationsquelle in y_t.

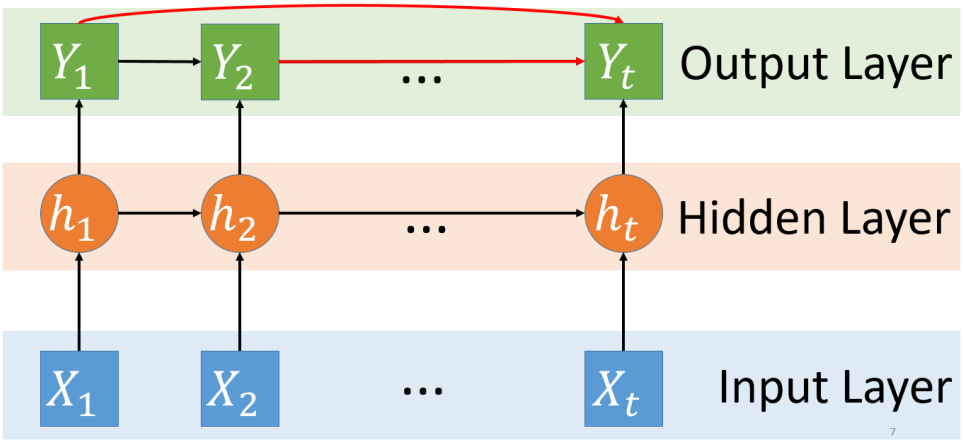

Abbildung 3: RNN Hierarchiediagramme und Workflow.

2.2 RNN Encoder–Decoder

Ein RNN Encoder-Decoder besteht aus drei Teilen, nämlich Encoder, hidden states und Decoder. Es lernt, eine Sequenz variabler Länge in eine Vektordarstellung mit fester Länge zu codieren und eine gegebene Vektordarstellung mit fester Länge zurück in eine Sequenz variabler Länge zu dekodieren.

Der Encoder ist ein RNN, der jedes Symbol einer Eingabesequenz x sequentiell liest. Beim Lesen jedes Symbols ändert sich der hidden state des RNN gemäß Gleichung $h_{<t>} = f(h_{<t-1>}, x_t)$. Nach dem Lesen des Endes der Sequenz (markiert durch ein Ende-der-Sequenz-Symbol) ist der hidden state des RNN eine Zusammenfassung c der gesamten Eingabesequenz.

Der Decoder ist ein weiterer RNN, der trainiert wird, um die Ausgabesequenz durch Vorhersagen des nächsten Symbols y_t zu erzeugen. Der hidden stated des Decoders zum Zeitpunkt t wird berechnet durch Gleichung $h_{<t>} = f(h_{<t-1>}, y_{t-1}, c)$, wobei c ist die Zusammenfassung.

Abbildung 4 zeigt eine typische Hierarchie von ein RNN Encoder-Decoder. Es

zeigt auch den Informationsfluss. Seine Eingabe ist $x_1, \ldots, x_t$, die Ausgabe ist $y_1, \ldots, y_t$. In diesem Diagramm sind sieben hidden states enthalten, wo c ist eine Zusammenfassung.

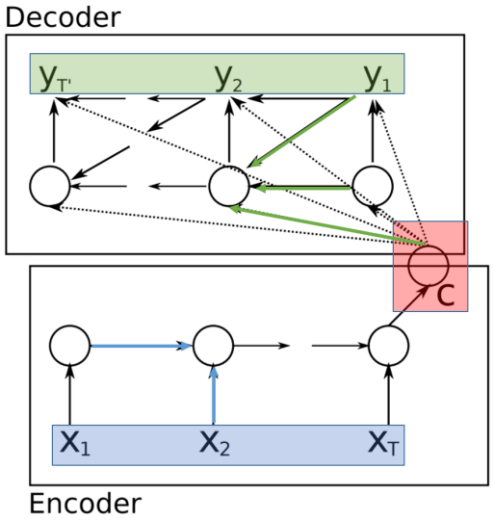

Abbildung 4: RNN Encoder-Decoder Hierarchiediagramme und der Informationsfluss.

Sobald der RNN Encoder-Decoder trainiert ist, kann das Modell auf zwei Arten verwendet werden. Der erste Fall ist, das Modell kann verwendet werden, um eine Zielsequenz bei einer gegebenen Eingabesequenz zu erzeugen. Der zweite Fall ist, das Modell kann verwendet werden, um ein gegebenes Paar von Eingabe- und Ausgabesequenzen zu bewerten, wobei die Bewertung einfach eine Wahrscheinlichkeit $p_\theta(y_n | x_n)$ ist. In dieser Formel, θ ist die Menge der Modellparameter (z.B. Gewichte) und jedes (x_n , y_n) ist ein Paar (Eingabesequenz, Ausgabesequenz) aus dem Training Set.

2.3 Hidden Unit

Die in dem Papier vorgeschlagene hidden unit hat eine fortgeschrittene Struktur. Das wurde von der LSTM-Einheit motiviert, ist aber viel einfacher zu berechnen und zu implementieren. Abbildung 5 zeigt die grafische Darstellung der vorgeschlagenen hidden unit.

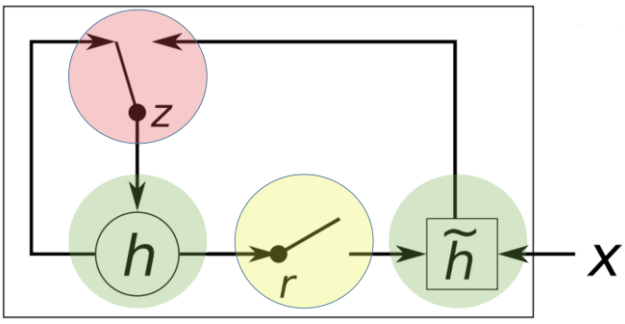

Abbildung 5: die grafische Darstellung der vorgeschlagenen hidden unit

In Abbildung 5 ist der rote Bereich Update-gate z. Das Update-gate z wählt aus, ob der hidden state mit einem neuen hidden state h̃ aktualisiert werden soll. Der gelbe Bereich ist Reset-Gate r. Das Reset-Gate r entscheidet, ob der vorherige hidden state ignoriert wird.

Zuerst wird das Reset-Gate r_j durch die Gleichung $r_j = \sigma([W_r x]_j + [U_r h_{<t-1>}]_j)$ berechnet. Wobei σ die logistische Sigmoidfunktion ist. Und $[W_r x]_j$ bezeichnet das j-te Element eines Vektors. x und $h_{<t-1>}$ sind die Eingabe und der vorherige hidden state. W_r und U_r sind Gewichtsmatrizen, die gelernt werden. In ähnlicher Weise wird das update-gate z_j berechnet durch die Gleichung $z_j = \sigma([W_z x]_j + [U_z h_{<t-1>}]_j)$.

Wenn das Reset-gate r_j nahe 0 ist, wird der neue hidden state h̃ gezwungen, den vorherigen hidden state h zu ignorieren und nur mit der aktuellen Eingabe x zurückzusetzen. Dies ermöglicht dem hidden state, Informationen, die sich später als irrelevant erweisen, effektiv fallen zu lassen, wodurch eine kompaktere Darstellung ermöglicht wird.

Auf der anderen Seite steuert das Update-gate z_j, wie viel Information von dem vorherigen hidden state zu dem gegenwärtigen hidden state übertragen wird. Es hilft dem RNN Encoder- Decoder, sich an "langfristige" Informationen zu erinnern.

Da jede Hidden state separate Reset - und Update - Gates hat, lernt jede Hidden state, Abhängigkeiten über verschiedene Zeitskalen zu erfassen. Diejenigen units, die lernen, kurzfristige Abhängigkeiten zu erfassen, haben tendenziell

Reset-Gates, die häufig aktiv sind. Aber diejenigen, die längerfristige Abhängigkeiten erfassen, haben "Update-Gates", die meistens aktiv sind.

In späteren Versuchen wurde es gefunden,, dass es wichtig ist, hidden unit mit Gattern zu verwenden, ohne Gating kann mankein aussagekräftiges Ergebnis erzielen.

3. Statistical Machine Translation

In diesem Kapitel wird hauptsächlich erläutert, was Statistical Machine Translation ist und wie RNN Encoder-Decoder in Statistical Machine Translation angewendet wird.

3.1 Definition und Beispiele für SMT

Übersetzungen von SMT werden auf der Basis von statistischen Modellen erstellt, deren Parameter aus der Analyse von zweisprachigen parallelen Textkorpora abgeleitet werden. Im Laufe der Zeit, Wir haben jetzt nicht nur SMT basierend auf Wörtern, sondern auch SMT basierend auf Phrase und Syntax. Abbildung 6 ist ein kurze Beispiel für Phrase-based SMT (alignment). Die Eingabe wird in beliebige Mehrworteinheiten (Sätze, Segmente und Blöcken) segmentiert. Jede der Einheiten wird in eine zielsprachliche Einheit übersetzt. Die Einheiten können neu geordnet werden.

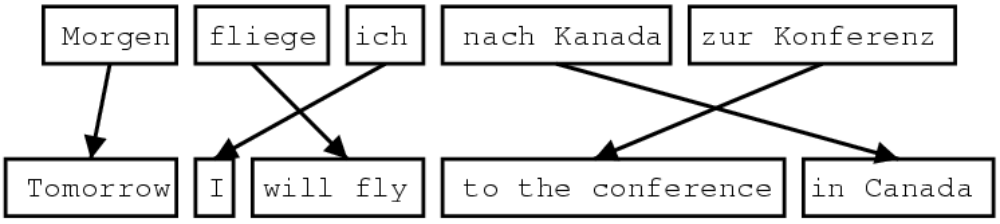

Abbildung 6: Bsp. Phrase-based SMT (alignment)

Das Folgende ist eine detailliertere Beschreibung der Formel für SMT. Die Formel ist $p(f|e) \propto p(e|f)p(f)$. Wobei f die Übersetzung ist und e der Quellsatz ist. $p(e|f)$ heißt Translation Modell, es ist die Übersetzungswahrscheinlichkeiten für “matching” Phrasen in den Quell- und Zielsätzen, vorgeschlagen von (Koehn et al., 2003) (Marcu and Wong, 2002). $p(f)$ heißt Language Modell, es ist eine Wahrscheinlichkeitsverteilung über

Sequenzen von Wörtern, vorgeschlagen von (Koehn, 2005). Das Ziel des Systems (speziell Decoder) ist es, eine Übersetzung f mit einem Quellsatz e zu finden. Und um die Qualität zu maximieren.

In der Praxis modellieren die meisten SMT-Systeme $\log \mathrm{p}\,(\mathrm{f} \mid \mathrm{e})$ jedoch als loglineares Modell mit zusätzlichen Merkmalen und entsprechenden Gewichten: $\log \mathrm{p}\,(\mathrm{f} \mid \mathrm{e}) = \sum_{n=1}^{N} w_n f_n(f, e) + \log Z(e)$. Wobei f_n und w_n das n-te Merkmal bzw. Gewicht sind. $Z(e)$ ist eine Normalisierungskonstante, die nicht von den Gewichten abhängt. Die Gewichtungen werden oft optimiert, um den BLEU-Wert eines Entwicklungssets zu maximieren.

BLEU (Bilingual Evaluation Understudy) ist ein Algorithmus zur Bewertung der Qualität von Text, der maschinell von einer natürlichen Sprache in eine andere übersetzt wurde. Die Ergebnisse werden für einzelne übersetzte Segmente - im Allgemeinen Sätze - berechnet, indem sie mit einer Reihe von Referenzübersetzungen guter Qualität verglichen werden. Diese Werte werden dann über den gesamten Korpus gemittelt, um eine Schätzung der Gesamtqualität der Übersetzung zu erhalten. Die Ausgabe von BLEU ist immer eine Zahl zwischen 0 und 1. Dieser Wert gibt an, wie ähnlich der Kandidatentext zu den Referenztexten ist, wobei Werte, die näher bei 1 liegen, ähnlichere Texte darstellen.

3.2 Scoring Phrase Pairs mit RNN Encoder-Decoder

In Abschnitt 2.2 haben wir zwei Anwendungen des RNN Encoder-Decoder Modells erwähnt. In diesem Abschnitt werden im Detail beschrieben, wie können wir den RNN Encoder-Decoder trainieren. Auf diese Weise können wir für jedes Phrasenpaar eine neue Bewertung zu der vorhandenen Phrasentabelle hinzufügen.

Der RNN Encoder– Decoder wurde an einer Tabelle von Phrasenpaaren trainiert und verwendet seine Bewertungen als zusätzliche Merkmale in dem loglinearen Modell in letzte Gleichung $\log \mathrm{p}\,(\mathrm{f} \mid \mathrm{e})$.

Wenn wir den RNN-Encoder-Decoder trainieren, ignorieren wir die normalisierten Frequenzen jedes Phrasenpaars in den ursprünglichen Korpora. Es stellt sicher, dass der RNN-Encoder-Decoder nicht einfach lernt, die Phrasenpaare nach ihrer Häufigkeit zu ordnen. Ein grundlegender Grund für diese Wahl war, dass die existierende Übersetzungswahrscheinlichkeit in der Phrasentabelle bereits die Häufigkeiten der Phrasenpaare im ursprünglichen Korpus wiedergibt. Mit einer festen Kapazität des RNN Encoder-Decoders versuchen wir sicherzustellen, dass der Großteil der Kapazität des Modells auf das Lernen sprachlicher Regelmäßigkeiten fokussiert ist, d. H. Auf plausible und unplausible Übersetzungen zu unterscheiden.

Sobald der RNN Encoder-Decoder trainiert ist, fügen wir der vorhandenen Phrasentabelle eine neue Bewertung für jedes Phrasenpaar hinzu. Dies ermöglicht, dass die neuen Bewertungen in den existierenden Abstimmungsalgorithmus mit minimalem Zusatzaufwand bei der Berechnung eingehen.

4. Experimente

Nach dem Abschluss des theoretischen Wissens ist das nächste die Experimente. Der Ansatz wurde auf der Englisch / Französisch -Übersetzungsaufgabe des *WMT'14-Workshops* evaluiert. Es ist ein zweisprachiges Korpora. Dazu gehören Europarl (61 Millionen Wörter), Nachrichtenkommentare (5,5 Millionen), UN (421 Millionen) und zwei „crawled“ Korpora von 90 Millionen Wörtern und 780 Millionen Wörtern.

4.1 Daten- und Baseline-System

Es wird allgemein anerkannt, dass das Trainieren statistischer Modelle zu all diesen Daten zu schlechter Leistung führt und extrem großen Modellen führt, die schwierig zu handhaben sind. Daher wurde das von (Moore and Lewis, 2010), (Axelrod et al., 2011) vorgeschlagene Datenauswahlverfahren verwendet. Ein „baseline phrase-based SMT system“ wird etabliert.

Eine Untermenge von 418M Wörtern aus mehr als 2G Wörtern wurde für die

Sprachmodellierung ausgewählt und eine Untermenge von 348M Wörtern aus 850M Wörtern zum Trainieren des RNN Encoder-Decoders. *Newstest 2012 und 2013* wurden für die Datenauswahl und Gewichtsabstimmung für SMT verwendet, und *Newstest2014* als Testset. Jeder Set hat mehr als 70.000 Wörter und eine einzige Referenzübersetzung.

Zum Training der neuronalen Netze und RNN-Encoder-Decoder. Die Quelle war begrenzt und zielte auf Vokabeln zu den häufigsten 15.000 Wörtern für Englisch und Französisch. Dies deckt ungefähr 93% des Datensatzes ab.

Zur gleichen Zeit, im Training, hat das „baseline phrase-based SMT system“ auch die folgenden Attribute. Der RNN-Encoder-Decoder, der in dem Experiment verwendet wurde, hatte 1000 hidden units mit den vorgeschlagenen Toren. Die Eingabe-/Ausgangsmatrix (zwischen jedem Eingabesymbol $x_{<t>}$ und hidden unit) wird mit zwei lower-rank Matrizen approximiert. Rank-100 -Matrizen wurden verwendet, entspricht einer Einbettung der Dimension 100 für jedes Wort. Bei jedem Update wurden 64 zufällig ausgewählte Phrasenpaare aus einer Phrasentabelle (die aus 348 Millionen Wörtern erstellt wurde) verwendet. Das Modell wurde für ca. 3 Tage trainiert.

Bislang hatte das gesamte neue Modell die Möglichkeit einer praktischen Bedienung. Als nächstes werden wir die Optimierung und den Unterschied zwischen dem neuen Modell und dem traditionellen Modell durch experimentelle Daten vergleichen. Das Ergebnis ist in Abbildung 7 dargestellt.

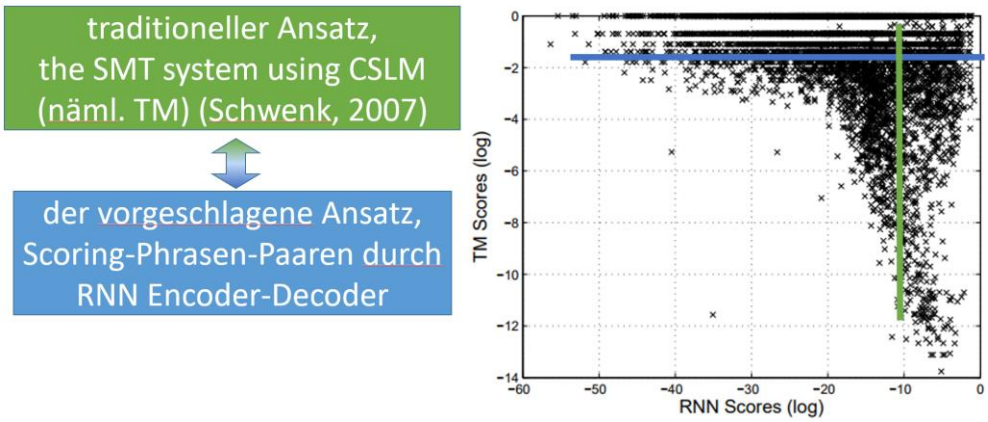

Abbildung 7: Der Vergleich der beiden Ansätze

Um die Effektivität von Scoring-Phrasen-Paaren mit dem vorgeschlagenen RNN Encoder-Decoder zu beurteilen, Ein traditioneller Ansatz the SMT system using CSLM (CSLM, nämlich Kontinuierliche Space Language Modeling) wurde zum Vergleich verwendet. Der Vergleich wird klären, ob sich die Beiträge von mehreren neuronalen Netzen in verschiedenen Teilen des SMT-Systems addieren oder redundant sind.

Abbildung 7 ist die Darstellung der Phrase Paare nach ihrem log-probabilities-Scores durch den RNN Encoder-Decoder und das TM (Translation Modell). Wir können es in Abbildung 7 finden, viele Phrasenpaare wurden sowohl vom Translationsmodell als auch vom RNN-Encoder-Decoder ähnlich bewertet, aber es gab ebenso viele andere Phrasenpaare, die radikal unterschiedlich bewertet wurden. Der Grund könnte sein, den RNN-Encoder-Decoder auf "einer Menge von einzigartigen" Phrasenpaaren zu trainieren und den RNN-Encoder-Decoder davon abzuhalten, einfach die Frequenzen der Phrasenpaare aus dem Korpus zu lernen.

4.2 Quantitative Analyse

In dem Experiment, die folgenden Kombinationen wurden ausprobiert. 1> Baseline configuration, 2> Baseline + RNN, 3> Baseline + CSLM + RNN. Die Ergebnisse sind in Abbildung 8 dargestellt.

Models	BLEU	
	dev	test
Baseline	30.64	33.30
RNN	31.20	33.87
CSLM + RNN	31.48	34.64

Addieren,
Nicht Redundant

Abbildung 8: Der Vergleich der 3 Kombinationen

Die beste Leistung wurde erzielt, wenn sowohl CSLM als auch die Phrasen-Scores vom RNN Encoder-Decoder verwendet wurden. Dies deutet darauf hin,

dass die Beiträge des CSLM und des RNN Encoder-Decoders nicht zu stark korreliert sind und d.H. die Beiträge von mehreren neuronalen Netzen in verschiedenen Teilen des SMT-Systems addieren sind. Man kann auch bessere Ergebnisse erwarten, wenn man jede Methode (namlich CSLM und RNN) unabhängig voneinander verbessert.

4.3 Qualitative Analyse

Im vorigen Abschnitt zeigten experimentelle Daten, dass nach der Verwendung von RNN Encoder-Decoder eine signifikante Optimierung demonstriert wurde. Aber woher die Leistungsverbesserung kommt?

Um zu verstehen, woher die Leistungsverbesserung kommt, analysieren wir die vom RNN Encoder-Decoder berechneten Phrasenpaarwerte mit dem entsprechenden $p(f|e)$ aus dem Translation Modell.

Wir erwarten, dass die häufigen Phrasen bessere Scores haben. Die seltenen Phrasen haben schlechte Scores. Wir erwarten ferner, dass der RNN Encoder-Decoder ohne Frequenzinformation trainiert wurde, um die Phrasenpaare eher auf der Grundlage der linguistischen Regelmäßigkeiten als auf der Statistik ihrer Vorkommen im Korpus zu bewerten. Deshalb konzentrieren Wir uns auf diejenigen Paare, deren Quellphrase lang ist (mehr als 3 Wörter pro Quellphrase) und häufig ist. Und auch, Die Paare, deren Quellphrase im Korpus lang, aber selten ist.

Source	Translation Model	RNN Encoder–Decoder
at the end of the	[a la fin de la] [f la fin des années] [être supprimés à la fin de la]	[à la fin du] [à la fin des] [à la fin de la]
for the first time	[r © pour la premirëre fois] [été donnés pour la première fois] [été commémorée pour la première fois]	[pour la première fois] [pour la première fois ,] [pour la première fois que]
in the United States and	[? aux ?tats-Unis et] [été ouvertes aux États-Unis et] [été constatées aux États-Unis et]	[aux Etats-Unis et] [des Etats-Unis et] [des États-Unis et]
, as well as	[?s , qu'] [?s , ainsi que] [?re aussi bien que]	[, ainsi qu'] [, ainsi que] [, ainsi que les]
one of the most	[?t ?l' un des plus] [?l' un des plus] [être retenue comme un de ses plus]	[l' un des] [le] [un des]

(a) Long, frequent source phrases

parts of the world .	[© gions du monde .] [régions du monde considérées .] [région du monde considérée .]	[parties du monde .] [les parties du monde .] [des parties du monde .]
the past few days .	[le petit texte .] [cours des tout derniers jours .] [les tout derniers jours .]	[ces derniers jours .] [les derniers jours .] [cours des derniers jours .]
on Friday and Saturday	[vendredi et samedi à la] [vendredi et samedi à] [se déroulera vendredi et samedi ,]	[le vendredi et le samedi] [le vendredi et samedi] [vendredi et samedi]

(b) Long, rare source phrases

Abbildung 9: Long, frequent source phrases und Long, rare source phrases

Abbildung 9 listet die Top-3-Zielphrasen pro Quellphrase auf, die entweder vom Translation modell oder vom RNN Encoder-Decoder bevorzugt werden. Die Quellphrasen wurden zufällig ausgewählt, die länger als 4 oder 5 Wörter waren. In den meisten Fällen bevorzugen RNN Encoder-Decoder tatsächliche oder wörtliche Übersetzungen und auch kürzere Phrasen im Allgemeinen.

Source	Samples from RNN Encoder–Decoder
at the end of the	[à la fin de la] (×11)
for the first time	[pour la première fois] (×24) [pour la première fois que] (×2)
in the United States and	[aux États-Unis et] (×6) [dans les États-Unis et] (×4)
, as well as	[, ainsi que] [,] [ainsi que] [, ainsi qu'] [et UNK]
one of the most	[l' un des plus] (×9) [l' un des] (×5) [l' une des plus] (×2)

(a) Long, frequent source phrases

Source	Samples from RNN Encoder–Decoder
, Minister of Communications and Transport	[, ministre des communications et le transport] (×13)
did not comply with the	[n' tait pas conforme aux] [n' a pas respect l'] (×2) [n' a pas respect la] (×3)
parts of the world .	[arts du monde .] (×11) [des arts du monde .] (×7)
the past few days .	[quelques jours .] (×5) [les derniers jours .] (×5) [ces derniers jours .] (×2)
on Friday and Saturday	[vendredi et samedi] (×5) [le vendredi et samedi] (×7) [le vendredi et le samedi] (×4)

(b) Long, rare source phrases

Abbildung 10: Long, frequent source phrases und Long, rare source phrases

Und in Abbildung 10 zeigt es für jeden der Quellenphrasen in Abb. 9 die erzeugten Abtastwerte vom RNN Encoder-Decoder.

Für jede Quellenphrase wurden 50 Proben erzeugt und die Top-5-Sätze entsprechend ihrer Bewertungen wurden gezeigt. Wir können sehen, dass der RNN Encoder-Decoder wohlgeformte Zielphrasen vorschlagen kann, ohne auf die tatsächliche Phrasentabelle zu schauen. Wichtig ist, dass die generierten Phrasen nicht vollständig mit den Zielphrasen aus der Phrasentabelle des „standard phrase-based SMT system" überlappen.

In Zukunft könnte es möglich sein, dass der RNN Encoder-Decoder die ganze oder einen Teil der Phrasentabelle ersetzt.

4.4 Word and Phrase Representations

RNN Encoder-Decoder projiziert eine Folge von Wörtern in einen kontinuierlichen Raumvektor und bildet sie dann zurück. d.h. Der RNN Encoder-Decoder kann beliebige Worte oder n-dimensionale Repräsentationen von Phrasen im kontinuierlichen Raum erzeugen.

Bei Wort Repräsentationen. Wir erwarten daher eine ähnliche Eigenschaft mit dem vorgeschlagenen Modell. Das erste Diagramm in Abbildung 11 zeigt die 2D-Einbettung der Wörter, die die vom RNN Encoder-Decoder gelernte Worteinbettmatrix verwendeten. Wir können deutlich sehen, dass semantisch ähnliche Wörter miteinander gruppiert sind. (Der rote, blaue, violette Bereich)

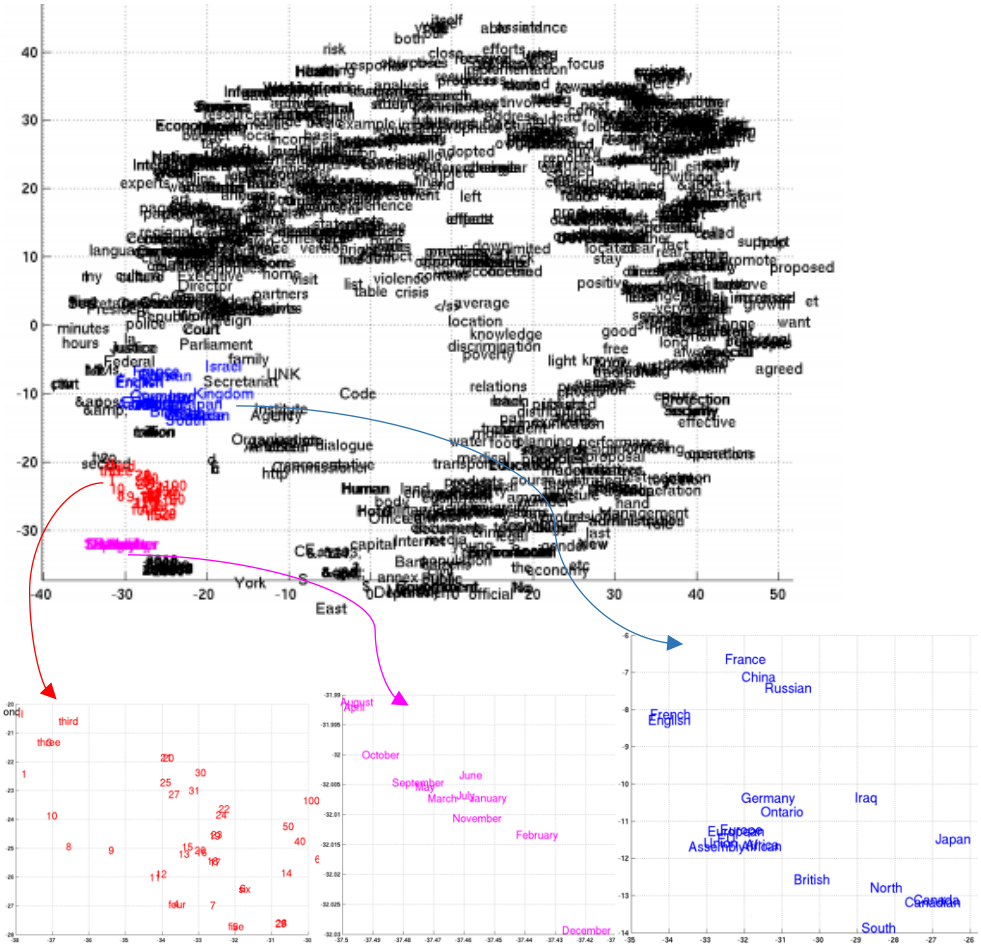

Abbildung 11: Wort Repräsentationen

Und bei Phrase Repräsentationen, aus der Visualisierung (Abb. 12) wird es deutlich, dass der RNN Encoder-Decoder sowohl semantische als auch syntaktische Strukturen der Phrasen erfasst.

In der roten Abbildung (Abb. 12.1) sind die meisten Phrasen über die Dauer der Zeit, während jene Phrasen, die syntaktisch ähnlich sind, zusammen gruppiert sind. In der lila Abbildung (Abb. 12.2) zeigt den Cluster von Phrasen, die semantisch ähnlich sind (Länder oder Regionen). Die blaue Abbildung (Abb. 12.3) zeigt sehr deutlich, dass die Phrasen syntaktisch ähnlich sind.

Abbildung 12: Phrase Repräsentationen

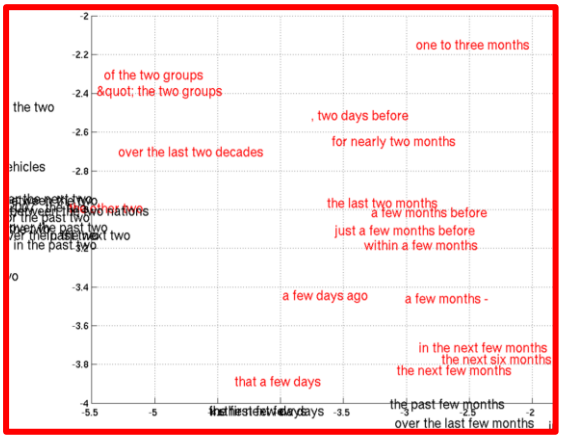

Abbildung 12.1: Phrase Repräsentationen, syntaktisch ähnlich, über die Dauer der Zeit

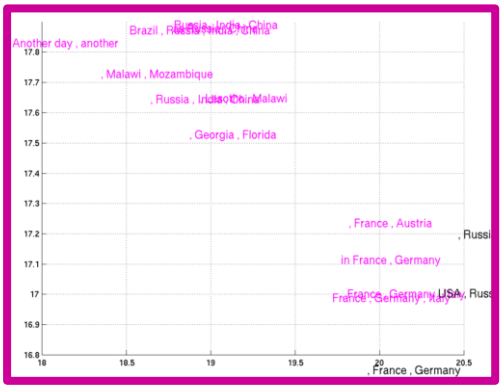

Abbildung 12.2: Phrase Repräsentationen, semantisch ähnlich

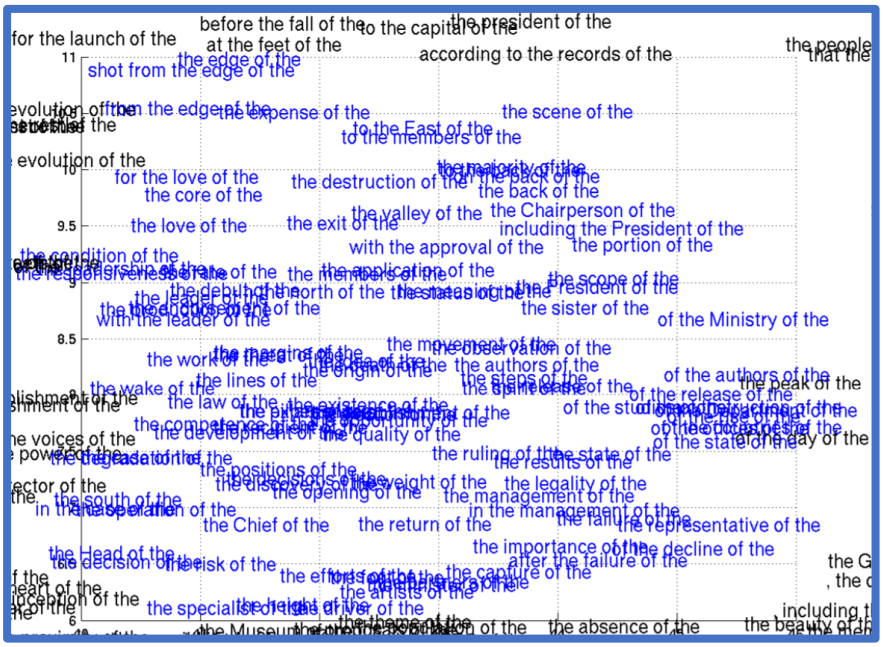

Abbildung 12.3: Phrase Repräsentationen, syntaktisch ähnlich

5. Zusammenfassung & Ausblick

In dieser Studie wurde eine neue neuronale Netzwerkarchitektur, genannt RNN Encoder-Decoder, vorgeschlagen. Er kann das Mapping von einer Sequenz beliebiger Länge zu einer anderen Sequenz lernen, möglicherweise von einer anderen Menge einer beliebigen Länge. Der RNN Encoder-Decoder kann entweder ein Paar von Sequenzen bewerten oder eine Zielsequenz mit einer Quellensequenz erzeugen.

Es wurde auch die neue hidden unit vorgeschlagen, die Reset-gate und Update-gate enthält, die adaptiv steuern, wie viel sich jede hidden unit beim Lesen oder Erzeugen einer Sequenz erinnert oder vergisst.

Das neue Modell hat auch gute Leistung und höhere BLEU-Score.

Blick in die Zukunft, Es zeigt großes Potenzial, dass es die ganze oder einen Teil der Phrasentabelle ersetzt. Und auch, es zeigt noch großes Potenzial, z.B. zu anderen Anwendungen wie Sprachtranskription.

Literaturverzeichnis

- (Kyunghyun Cho et al., 2014) Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, Yoshua Bengio: Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. EMNLP 2014
- https://en.wikipedia.org/wiki/Rule-based_machine_translation
- https://medium.com/@Synced/history-and-frontier-of-the-neural-machine-translation-dc981d25422d
- https://en.wikipedia.org/wiki/Recurrent_neural_network
- https://arxiv.org/pdf/1406.1078.pdf
- https://en.wikipedia.org/wiki/Recurrent_neural_network
- https://en.wikipedia.org/wiki/Nonlinear_system
- https://en.wikipedia.org/wiki/Logistic_function
- https://en.wikipedia.org/wiki/Google_Translate
- http://www.statmt.org/wpt05/mt-shared-task/
- https://en.wikipedia.org/wiki/BLEU
- https://en.wikipedia.org/wiki/Neural_machine_translation
- http://statmt.org/wmt14/translation-task.html
- https://www.quora.com/What-is-the-meaning-of-low-rank-matrix