Transformer-Based NMT: Modeling, Training and Implementation

Dissertation

zur Erlangung des akademischen Grades eines

Doktors der Philosophie

der Philosophischen Fakultäten

der Universität des Saarlandes

vorgelegt von

Hongfei Xu

aus

Henan, China

Saarbrücken, 2021

Der Dekan: Prof. Dr. Augustin Speyer

Erstberichterstatter: Prof. Dr. Josef van Genabith

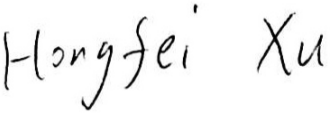
Zweitberichterstatter: Prof. Dr. Deyi Xiong

Tag der letzten Prüfungsleistung: 04.11.2021

Declaration of Independence

I, Hongfei Xu, declare that:

- The work reported in the thesis "Transformer-Based NMT: Modeling, Training and Implementation" is my own work.
- Where I consulted previous work, proper reference to such previous work is provided in the thesis.
- For joint publications resulting from the research presented in the thesis, my contribution is stated clearly.
- No part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution.

Signed: 

Date: 2021.5.25

"Practice is the sole criterion for testing truth."

Xiaoping Deng

Abstract

Transformer-Based NMT: Modeling, Training and Implementation

by Hongfei Xu

Doctor of Philosophy

Department of Language Science and Technology, Fachrichtung Sprachwissenschaft und Sprachtechnologie

Universität des Saarlandes

International trade and industrial collaborations enable countries and regions to concentrate their developments on specific industries while making the most of other countries' specializations, which significantly accelerates global development. However, globalization also increases the demand for cross-region communication. Language barriers between many languages worldwide create a challenge for achieving deep collaboration between groups speaking different languages, increasing the need for translation.

Language technology, specifically, Machine Translation (MT) holds the promise to enable communication between languages efficiently in real-time with minimal costs.

Even though nowadays computers can perform computation in parallel very fast, which provides machine translation users with translations with very low latency, and although the evolution from Statistical Machine Translation (SMT) to Neural Machine Translation (NMT) with the utilization of advanced deep learning algorithms has significantly boosted translation quality, current machine translation algorithms are still far from accurately translating all input. Thus, how to further improve the performance of state-of-the-art NMT algorithm remains a valuable open research question which has received a wide range of attention.

In the research presented in this thesis, we first investigate the long-distance relation modeling ability of the state-of-the-art NMT model, the Transformer. We propose to learn source phrase representations and incorporate them into the Transformer translation model, aiming to enhance its ability to capture long-distance dependencies well.

Second, though previous work (Bapna et al., 2018) suggests that deep Transformers have difficulty in converging, we empirically find that the convergence of deep Transformers depends on the interaction between the layer normalization and residual connections employed to stabilize its training. We conduct a theoretical study about how to ensure the convergence of Transformers, especially for deep Transformers, and propose to ensure the convergence of deep Transformers by putting the Lipschitz constraint on its parameter initialization.

Finally, we investigate how to dynamically determine proper and efficient batch sizes during the training of the Transformer model. We find that the gradient direction gets stabilized with increasing batch size during gradient accumulation. Thus we propose to dynamically adjust batch sizes during training by monitoring the gradient direction change within gradient accumulation, and to achieve a proper and efficient batch size by stopping the gradient accumulation when the gradient direction starts to fluctuate.

For our research in this thesis, we also implement our own NMT toolkit, the Neutron implementation of the Transformer and its variants. In addition to providing fundamental features as the basis of our implementations for the approaches presented in this thesis, we support many advanced features from recent cutting-edge research work. Implementations of all our approaches in this thesis are also included and open-sourced in the toolkit.

To compare with previous approaches, we mainly conducted our experiments on the data from the WMT 14 English to German (En-De) and English to French (En-Fr) news translation tasks, except when studying the convergence of deep Transformers, where we alternated the WMT 14 En-Fr task with the WMT 15 Czech to English (Cs-En) news translation task to compare with Bapna et al. (2018). The sizes of these datasets vary from medium (the WMT 14 En-De, $\sim 4.5M$ sentence pairs) to very large (the WMT 14 En-Fr, $\sim 36M$ sentence pairs), thus we suggest our approaches help improve the translation quality between popular language pairs which are widely used and have sufficient data.

German Summary

(Zusammenfassung)

In den letzten Jahrzehnten haben die weltweite Zusammenarbeit, Handel und Industrie-Verbindungen Ländern und Regionen ermöglicht, ihre Entwicklungen auf bestimmte Branchen zu konzentrieren, und dadurch die jeweilige Spezialisierungen der Länder zu nutzen, was deutlich die globale Entwicklung beschleunigt. Die Globalisierung erhöht jedoch auch die überregionale Kommunikation, und die Sprachbarrieren zwischen den Sprachen erhöhen die Nachfrage nach Übersetzungen, was für eine enge Zusammenarbeit zwischen Gruppen, die verschiedene Sprachen sprechen, von wichtiger Bedeutung ist. Die Sprachtechnologie, insbesondere die maschinelle Übersetzung (Machine Translation, MT), verspricht eine effiziente Überbrückung von Sprachbarrieren in Echtzeit mit minimalen Kosten.

1) Heutzutage können Computer sehr schnell Berechnungen parallel durchführen, was Benutzern von maschineller Übersetzung eine sehr geringe Latenzzeit bringt. 2) Die Entwicklung von statistischer maschineller Übersetzung (Statistical Machine Translation, SMT) zu neuronaler maschineller Übersetzung (Neural Machine Translation, NMT) mit der Nutzung von Deep-Lernalgorithmen hat die Übersetzungsqualität deutlich gesteigert. und 3) Neuronale Modelle, die für NMT entwickelt wurden, haben sich in den letzten Jahren mehrmals weiterentwickelt, und die Menge an parallelen Daten auf Satzebene für das Training ist für einige Sprachpaare sehr groß. Zur gleichen Zeit ist aber das hochmoderne NMT Transformer Modell, das gute Übersetzungsergebnisse liefern kann, noch weit davon entfernt, alle Eingaben richtig zu übersetzen. Daher bleibt die Frage, wie die Leistung des hochmodernen Transformer-Übersetzungsmodells verbessert werden kann, noch eine wichtige offene Forschungsfrage die viel Aufmerksamkeit erhält. Außerdem basiert heutzutage eine breite Palette von NLP-Aufgaben (Natural Language Processing, NLP) auf der Feinabstimmung eines vortrainierten BERT (Devlin et al., 2019) Modells und dadurch ist Forschung den Transformer zu verbessern, sehr relevant, da BERT hauptsächlich auf dem Transformer-Encoder basiert.

In Bezug auf die jüngste Entwicklung des NMT-Modelldesigns begann die Anwendung neuronaler Netze für MT mit rekurrenten neuronalen Netzen (Recurrent Neural Networks, RNNs), wo zwei RNNs (Sutskever et al., 2014) verwendet werden, der eine (nämlich der Codierer), um den Quellensatz zu codieren, der in dem Prozess eine Wortsinn-Disambiguierung im Kontext vornimmt, der andere (der Decodierer), der die codierte Einbettung übernimmt und die entsprechende Übersetzung automatisch regressiv auf Token-für-Token-Weise erzeugt, wie ein Sprachmodell mit einem Eingabevektor mit einer festen Dimension. Da die Informationen eines Quellensatzes, die durch einen Vektor mit fester Dimensionalität dargestellt wird, wahrscheinlich zu einem Informationsverlust führt, insbesondere bei langen Sätzen, wird ein Aufmerksamkeitsmechanismus eingeführt, um den Decoder besser mit dem Codierer zu verbinden und gemeinsam das Übersetzen und die Alignierung (Bahdanau et al., 2014) zu lernen. Der Aufmerksamkeitsmechanismus ermöglicht es dem Decoder, bei jedem Decodierungsschritt auf jeden einzelnen Schritt der gesamten vorherigen Codierungssequenz zurück zugreifen, um kontinuierlich relevante Informationen des Quellensatzes für die Erzeugung von Wörtern in der Zielsprache während der Decodierung bereitzustellen. Während RNNs eine sequentielle Token-für-Token-Weise berechnen, was eine effiziente Parallelisierung des Modells auf modernen GPUs verhindert, werden Convolutional Neural Networks (CNNs) angewendet, um dieses Problem zu lösen und RNNs durch Positionsinformationen zu ersetzen, die durch trainierte Positionseinbettungen bereitgestellt werden (Gehring et al., 2017). CNNs können jedoch nur Kontexte innerhalb eines vorgegebenen Fensters verwenden, und deshalb wurde die Mehrkopf-Aufmerksamkeitsmaschinerie als Teil des Transformer-Übersetzungsmodells vorgeschlagen, um die Modellierung über die gesamte Sequenz zu ermöglichen und gleichzeitig die unabhängige Entwicklung von Token-Repräsentationen zu gewährleisten und eine effiziente Parallelisierung zu ermöglichen (Vaswani et al., 2017). Der Erfolg des Transformers, der state-of-the-art Übersetzungsleistungen erzielt, hat sowohl in der Forschungsgemeinschaft als auch in der Industrie große Aufmerksamkeit gefunden.

Ist das starke Transformer-Übersetzungsmodell nun gut genug und alles, was wir brauchen? Anscheinend gibt es Raum für Verbesserungen: Das Modell kann immer noch nicht alle Übersetzungsanfragen korrekt übersetzen. Das Hauptziel der in dieser Arbeit

vorgestellten Forschung ist es, die Qualität der Übersetzung des Transformer-Modells zu verbessern.

Wir beschreiben unsere Forschungsansätze auf zwei sich ergänzende Weisen: (i) wir entwickeln die Modellarchitektur weiter, um die Fähigkeit des Transformers zu verbessern Fern-Beziehungen zu erfassen, und (ii) die Verbesserung der Optimierung des Lernprozesses des Transformers, einschließlich Parameterinitialisierung und der dynamischen Auswahl der Chargengröße.

Insbesondere konzentrieren wir uns im ersten Teil unserer Forschung auf das Lernen und die Verwendung zusätzlicher Phrasen Darstellungen von Quellensätzen in der Modellarchitektur (beschrieben in Kapitel 3), um deren Fähigkeit zum Lernen von Abhängigkeiten über große Entfernungen zu verbessern unter der Motivation, dass das Modell nach der Aufmerksamkeit auf die Phrasendarstellungssequenz, die kürzer als die entsprechende Token-Darstellungssequenz ist, eine bessere Aufmerksamkeit auf Tokenebene erzielen kann, insbesondere, wenn Fernabhängigkeiten zu erfassen sind.

Tai et al. (2015) zeigt, dass das Long Short-Term Memory Network (LSTM) kurze Sätze besser verarbeitet als lange Sätze. In Linzen et al. (2016)'s Aufgabe zur Vorhersage der Subjekt-Verb-Nummeruskohärenz, die den Nummerus des folgenden Verbs (Plural oder Singular) des Satzes vorhersagt, verschlechtert sich die Genauigkeit von LSTMs konsistent mit zunehmenden Abständen zwischen Subjekt und Verb. Yang et al. (2017) zeigt, dass es für das LSTM-basierte NMT-Modell eine Herausforderung ist, Fernabhängigkeiten zu erfassen. Um die Fähigkeit der Erfassung von Fern-Beziehungen der NMT-Modelle beurteilen zu können, untersucht Tang et al. (2018) die Leistung von RNNs, CNNs und dem Transformer hinsichtlich der Subjekt-Verb-Kongruenz (Subject-Verb Agreement, SVA) Aufgabe, die die beliebteste Wahl für die Bewertung der Fähigkeit ist weitreichende Abhängigkeiten zu erfassen und in vielen Studien verwendet worden ist (Linzen et al., 2016; Sennrich, 2017). Sie zeigen, dass die Transformer-Modelle, die ja entfernte Token über kürzere Pfade als RNNs verbinden, nicht besonders besser als RNN Modelle für große Entfernungen sind, und dass die Anzahl der Köpfe bei Mehrkopfaufmerksamkeit für ihre Leistung über große Entfernungen entscheidend ist. Yang et al. (2019b) zeigt, dass die Genauigkeit von Encodern, einschließlich des Transformer-Self-Attentional-Encoders und

des Gated Recurrent Unit (GRU) (Cho et al., 2014) -Codierers, bei langen Abhängigkeiten über Sprachpaare hinweg abnimmt und sie untersuchen Modellvarianten für die Erkennung von Wortumordnungen, wo die Ergebnisse auch darauf hindeuten, dass sowohl GRU als auch das Self-Attention Network (SAN) Fernabhängigkeiten nicht vollständig erfassen können. Dies führt uns zu unserer ersten Forschungsfrage (Research Question, RQ).

RQ1: *Wie kann die Fähigkeit des Transformers zur Erfassung von Fernbeziehungen verbessert werden?*

In Kapitel 3 entwickeln wir eine Lösung für dieses Problem. In Anbetracht der Tatsache, dass die Modellierung von Phrasen statt Wörtern sehr deutlich die statistische maschinelle Übersetzung (Statistical Machine Translation, SMT) durch die Verwendung von größeren Übersetzungsblöcken ("Phrasen"), als auch die Fähigkeit zur Neuordnung verbessert hat, sollte die Modellierung von NMT auf Phrasen-Ebene ein intuitiver Vorschlag sein dem Modell zu helfen, Fernbeziehungen besser zu erfassen. Daher schlagen wir vor, dass der Transformer neben Token-Darstellungen auch Phrasendarstellungen verwendet.

Es gibt jedoch viel mehr mögliche Phrasen als Tokens, und eine Phrasentabelle ist um Größenordnungen größer als das Wortvokabular. Bei NMT ist es aufgrund von Speicherbeschränkungen nicht möglich, Phraseneinbettungen direkt in GPUs zu verwenden, und die Verteilung über Phrasen ist viel spärlicher als die über Wörter, was zu Datenmangel führen und die Leistung von NMT beeinträchtigen kann. Dies wirft unsere zweite Forschungsfrage auf:

RQ2: *Wie vermeiden wir die große Phrasentabelle, während wir von Phrasendarstellungen profitieren?*

Um das Problem der großen Phrasentabelle zu lösen, schlagen wir ein selektives Merkmalsextraktionsmodell vor und generieren eine Phrasendarstellung basierend auf Token-Darstellungen im laufenden Betrieb (in Kapitel 3). Insbesondere fasst unser Modell zuerst die Darstellung einer bestimmten Token-Sequenz mit der Mittelwert- oder Max-Over-Time-Pooling-Operation zusammen und berechnet dann das Aufmerksamkeitsgewicht

jedes Tokens basierend auf der ursprünglichen Token-Darstellung und der zusammengefassten Phrasendarstellung mit einem Feed-Forward Neural Network (FFNN) und generiert die endgültige Phrasendarstellung durch eine gewichtete Kombination der Token-Darstellungen in der Phrase nach Normalisierung der Aufmerksamkeitsgewichte. Der Haupttrick besteht darin, dass wir Phrasen nicht als Blöcke in Übersetzungen verwenden: die Übersetzung ist immer noch wortbasiert, aber unser Aufmerksamkeitsmodell ermöglicht es, die Darstellung von Wörtern durch die Phrasen zu informieren, die im laufenden Betrieb berechnet werden.

Nachdem der Ansatz zur Generierung von Phrasendarstellungen basierend auf Token-Darstellungen vorgeschlagen wurde, ist die dritte Forschungsfrage:

RQ3: *Wie lernen und verwenden wir Phrasendarstellungen im Transformer-Übersetzungsmodell?*

In Kapitel 3 zeigen wir, wie wir das Design des Transformer-Übersetzungsmodells ändern, um das Lernen und die Verwendung von Phrasendarstellungen in das Modell integrieren, damit alle Aspekte des Modells gemeinsam end-to-end trainiert werden können. Besonders schlagen wir ein Aufmerksamkeits-basiertes Kombinationsnetzwerk vor, das sich um Phrasendarstellungen kümmert, und fügen das Aufmerksamkeits-basiertes Kombinationsnetzwerk in jede Codierer- und Decodierschicht des Transformer-Übersetzungsmodells ein, damit jedes Token zuerst auf Phrasen achten kann, bevor es auf die ursprünglichen Quell-Token-Darstellungen achtet. Beachten Sie, dass in unserem Modell Phrasen nur für die Quelleneingabe berechnet werden. Sie werden aber sowohl im Codierer als auch im Decodierer verwendet, aber der Decodierer hat nur Zugriff auf Quellphrasen in der Quer-Aufmerksamkeit. Wir berechnen aber keine Phrasen auf der Zielseite des Decoders.

Die Forschung zur Adressierung von RQ 1, 2 und 3 in Kapitel 3 wurde in Xu et al. (2020c) auf der ACL 2020 veröffentlicht.

Im zweiten Teil unserer Forschung konzentrieren wir uns auf die Konvergenz und die Auswirkungen von Chargengrößen (batch sizes) und die automatische Bestimmung dynamisch veränderlicher Chargengrößen zur Optimierung des Transformers (beschrieben in Kapitel 4 und 5).

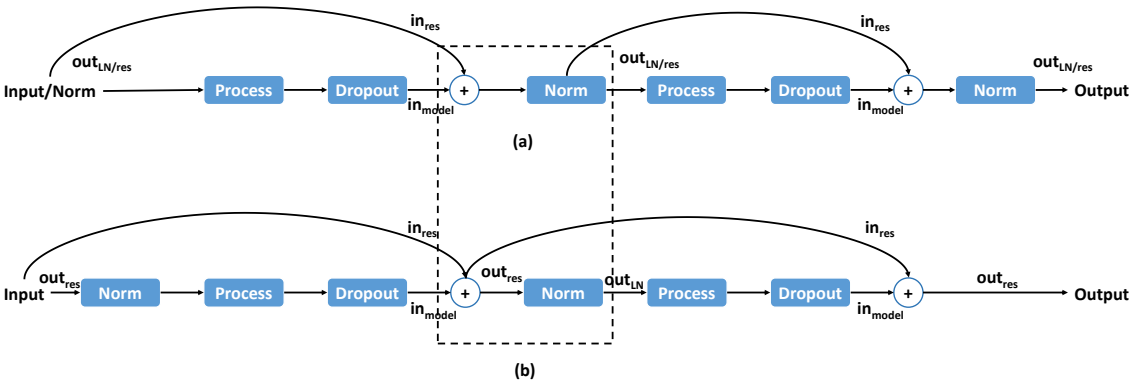


FIGURE 1: Zwei Berechnungssequenzen von Transformer-Übersetzungsmodellen: (a) das in dem ursprünglichen Papier beschrieben wird, (b) die offizielle Umsetzung.

In Bezug auf das Thema der Konvergenz des Transformers werden neuronale Netze (NN) in der Regel zuerst zufällig initialisiert, dann trainiert, um eine vordefinierte Verlustfunktion zu minimieren, weil die Konvergenz neuronaler Modelle natürlich entscheidend ist, um von ihrer Leistung zu profitieren. Ein Modell, das nicht konvergiert, kann keine aussagekräftigen Vorhersagen treffen. Für die Optimierung des Transformers, insbesondere für tiefe Transformer, haben frühere Untersuchungen (Bapna et al., 2018) gezeigt, dass das Transformer-Übersetzungsmodell zwar residuale Verbindungen und Schichtnormalisierung verwendet, um die Optimierungsschwierigkeiten zu verringern, die durch die tiefen mehrschichtigen Codierer/Decodierer verursacht werden. Normale tiefe Transformer haben immer noch Schwierigkeiten beim Training, und insbesondere Transformer-Modelle mit mehr als 12 Encoder/Decoder-Schichten konvergieren nicht. Bapna et al. (2018) schlägt einen TA-Mechanismus (Transparent Attention) vor, der gewichtete Ausgaben von Codiererschichten für jede Decodierschicht kombiniert, anstatt nur die Ausgabe der letzten Codierschicht zu verwenden. Dieser Ansatz bringt die meisten Verbesserungen mit einem Transformer mit 16 Encoder-Schichten. Aber bei Computer Vision (CV) -Aufgaben haben residuale Verbindungen jedoch ihre starke Fähigkeit gezeigt, die Konvergenz tiefer Modelle mit mehr als hundert Schichten sicherzustellen. Warum scheitert es bei tiefen Transformern? Dies führt zu unserer vierten Forschungsfrage:

RQ4: *Warum haben Transformer, speziell tiefe Transformer, Schwierigkeiten, selbst mit Schichtnormalisierung und residualen Verbindungen zu konvergieren?*

In Kapitel 4 zeigen wir zunächst empirisch, dass eine einfache Änderung in der offiziellen Implementierung (Vaswani et al., 2018), die die Berechnungsreihenfolge der residualen Verbindungen und der Schichtnormalisierung ändert, die Optimierung der tiefen Transformers erheblich erleichtern kann. Speziell wird im Originalpapier (Vaswani et al., 2017) in der Reihenfolge *der Verarbeitung* → *dropout* → *residuale Verbindung* → *der Schichtnormalisierung* berechnet, wobei die Verarbeitung die Berechnung der Mehrkopfaufmerksamkeit oder des positionsweisen vorwärtsgerichteten neuronalen Netzwerks ist, während die offizielle Implementierung in der Reihenfolge *der Schichtnormalisierung* → *Verarbeitung* → *dropout* → *residuale Verbindung* berechnet. Abbildung 1 zeigt die beiden Arten von Berechnungsreihenfolgen. Wir empfehlen, die Ausgabe der Ebenennormalisierung ($out_{LN/res}$) als Ausgabe der residualen Verbindung zu betrachten und nicht die Hinzufügung von in_{res} und in_{model} in Abbildung 1 (a), weil es ($out_{LN/res}$) die Eingabe (in_{res}) der nächsten residualen Verbindungsberechnung ist. Wir führen dann eine theoretische Analyse basierend auf der Differenz zwischen den Berechnungsreihenfolgen durch, die darauf hinweist, dass die Schichtnormalisierung über residuale Verbindungen in Abbildung 1 (a) die Auswirkung von residualen Verbindungen aufgrund der nachfolgenden Schichtnormalisierung wirksam reduzieren kann, um eine mögliche Explosion der Gradienten der kombinierten Schichtausgaben zu vermeiden (Chen et al., 2018b). Diese Analyse wirft die fünfte Forschungsfrage auf:

RQ5: *Wie kann man verhindern, dass die Schichtnormalisierung den Beitrag der residualen Verbindungen schmälert?*

In Kapitel 4 stellen wir eine Parameterinitialisierungsmethode vor, die die Lipschitz-Einschränkung für die Initialisierung von Transformerparametern nutzt um zu verhindern dass der Beitrag der residualen Verbindungen geschmälert wird und um die Trainings Konvergenz von tiefen Transformatoren effektiv zu gewährleisten. Im Gegensatz zu früheren Forschungsergebnissen zeigen wir ferner, dass mit der Lipschitz-Parameterinitialisierung tiefe Transformatoren mit der *ursprünglichen* Berechnungsreihenfolge konvergieren und signifikante BLEU-Verbesserungen mit bis zu 24 Schichten produzieren können.

Die in Kapitel 4 vorgestellte Forschungsergebnisse für RQ 4 und 5 wurden in Xu et al. (2020a) auf der ACL 2020 veröffentlicht.

Die Leistung neuronaler Modelle wird durch die Wahl der Hyperparameter stark beeinflusst. Während sich viele frühere Forschungen (Sutskever et al., 2013; Duchi et al., 2011; Kingma and Ba, 2015) auf die Beschleunigung der Konvergenz und die Verringerung der Auswirkungen der Lernrate konzentrieren, fokussieren sich vergleichsweise wenige Artikel auf den Effekt der Chargengröße (batch size). Es wurde jedoch festgestellt, dass die Chargengröße ein wichtiger Hyperparameter für die Leistung des Transformers ist, und einige Chargengrößen führen empirisch zu einer besseren Leistung als andere. Insbesondere wurde gezeigt, dass die Leistung des Transformer-Modells (Vaswani et al., 2017) für die neuronale maschinelle Übersetzung (Bahdanau et al., 2014; Gehring et al., 2017) stark von der Chargengröße (Popel and Bojar, 2018; Ott et al., 2018; Abdou et al., 2017; Zhang et al., 2019a) abhängt. Eine größere Charge führt normalerweise zu einer besseren Leistung. Der Einfluss der Chargengröße auf die Leistung wirft die sechste Forschungsfrage auf:

RQ6: *Wie kann man während des Trainings dynamisch und automatisch die richtigen und effizienten Chargengrößen finden?*

Um diese Forschungsfrage zu lösen, empfehlen wir, die Gradienten während des Trainings zu beobachten, wo sie die Größe und Richtung für die Optimierung aufzeigen, und die Beziehung zwischen der Chargengröße und den Gradienten zu untersuchen. Speziell beobachten wir die Auswirkungen auf die Gradienten mit zunehmender Chargengröße bei der Gradientenakkumulation, die Gradienten kleinerer Mini-Chargen als Gradienten einer größeren Mini-Charge, die diese kleineren Mini-Chargen umfasst, akkumulieren, und stellen fest, dass eine große Charge die Richtung der Gradienten stabilisiert. Basierend auf unserer Beobachtung schlagen wir dann vor, die dynamischen Chargengrößen im Training automatisch zu bestimmen, indem wir die Änderung der Gradientenrichtung überwachen, während Gradienten kleiner Chargen akkumuliert werden, bis die Gradientenrichtung zu schwanken beginnt.

Leider kann ein Transformer Modell in der Praxis eine große Anzahl von Parametern aufweisen, und die Kosten für die Berechnung der Gradientenrichtungsänderung sind relativ hoch, was die Effizienz des Trainings beeinträchtigen kann, wenn die Überwachung der Gradientenrichtungsänderung zur dynamischen Berechnung von Chargengrößen eingesetzt wird. Dies führt zu unserer siebten Forschungsfrage:

RQ7: *Wie kann die Änderung der Gradientenrichtung effizient überwacht werden?*

Um dieses Problem zu lösen, schlagen wir vor, Modellparameter in Gruppen zu unterteilen und die Änderung der Gradientenrichtung nur für eine ausgewählte Gruppe zu überwachen, die in jedem Optimierungsschritt für den Hyperparameter der Chargengröße bedeutsam ist. Insbesondere zeichnen wir die Reduzierung der Gradientenrichtungsänderung der überwachten Parametergruppe in jedem Optimierungsschritt auf. Wir empfehlen, dass die Parametergruppe mit mehr Reduktion empfindlicher auf die Chargengröße reagiert als die anderen mit weniger Reduktion. Somit normalisieren wir die Reduktionen aller Parametergruppen und verwenden sie als Stichprobenwahrscheinlichkeiten für entsprechende Parametergruppen.

Unsere in Kapitel 5 vorgestellten Forschungsergebnisse zu RQ 6 und 7 wurden in Xu et al. (2020b) auf der ACL 2020 veröffentlicht.

Um unsere Forschung zu unterstützen, entwickeln wir die Neutron Implementierung des Transformer-Modells und seiner verschiedenen Varianten. Wir präsentieren unsere technischen Arbeiten in Kapitel 6. Speziell um unsere eigenen Modelle zu entwickeln und mit anderen Ansätzen vergleichen zu können, implementieren wir zusätzlich zur Standardimplementierung des Transformers einige neuere Forschungsergebnisse zum Transformer, während wir unsere Wege zur Verbesserung des Transformers entwickelt haben, einschließlich des Durchschnitts Attention Network (Average Attention Network, AAN) zur Beschleunigung der Dekodierung des Transformers (Zhang et al., 2018a), die hierarchische Schichtaggregation zur Verschmelzung flacher Schichten (Dou et al., 2018), die Verwendung eines rekurrenten Decoders (Chen et al., 2018b), die Modellierung des sententialen Kontexts (Wang et al., 2019e) zur Verbesserung des MT Qualität, transparente Aufmerksamkeit (Bapna et al., 2018), um die Konvergenz von Deep Encodern und dem

Transformer auf Dokumentebene sicherzustellen (Zhang et al., 2018c). Wir unterstützen auch den von Wang et al. (2018c) vorgeschlagenen effizienten Trainingsplaner (dynamischer Stichproben- und Überprüfungsmechanismus) und andere erweiterte Funktionen wie neue Optimierer (z.B. RAdam (Liu et al., 2020), Lookahead (Zhang et al., 2019c) und deren Kombination). Die Implementierung all unserer eigenen Ansätze in dieser Arbeit ist ebenfalls enthalten und im Toolkit Open-Source verfügbar.

English Summary

Over the past few decades, worldwide collaboration, trade and industrial connections have enabled countries and regions to concentrate their developments on specific industries while making the most of the other countries' specializations, which significantly accelerates global development. However, globalization also increases cross-region communication, and the language barriers between languages increase the demand for translation which is crucial to achieving deep collaboration between groups speaking different languages. Language technology, specifically, Machine Translation (MT), holds great promise to bridge between languages efficiently in real-time with minimal cost.

1) nowadays computers can perform computation in parallel very fast, which provides machine translation to users with very low latency, 2) the evolution from Statistical Machine Translation (SMT) to Neural Machine Translation (NMT) with the utilization of advanced deep learning algorithms has significantly boosted translation quality, and 3) neural models designed for NMT have evolved several times over the last years, and the amount of sentence-level parallel data for their training is very large for some language pairs. At the same time, the state-of-the-art NMT model, the Transformer, which can provide good translation results, is still far from accurately translating all input. Thus, how to improve the performance of the state-of-the-art Transformer translation model remains an important open research question, and has received a lot of attention. Furthermore, as nowadays a wide range of Natural Language Processing (NLP) tasks are based on fine-tuning a pre-trained BERT (Devlin et al., 2019) model, research on improving the Transformer is highly relevant, as BERT is mainly based on the Transformer encoder.

In terms of the recent evolution of NMT model design, the application of neural networks for MT started with Recurrent Neural Networks (RNNs), where two RNNs are employed (Sutskever et al., 2014), one (namely, the encoder) to encode the source sentence to a fixed dimension vector which in the process includes word sense disambiguation in context, the other (the decoder) to take the encoded embedding and generate the corresponding translation auto-regressively in a token-by-token manner, like a language model.

As representing the information of a source sentence with a fixed dimension vector is likely to incur information loss, especially with long sentences, an attention mechanism is introduced to better connect the decoder with the encoder to jointly learn to translate and align (Bahdanau et al., 2014). The attention mechanism enables the decoder to attend the whole encoding sequence at each decoding step to continuously provide relevant information of the source sentence to the generation of target tokens during decoding. While RNNs compute in a sequential token-by-token manner, which prevents the model from efficient parallelization on modern GPUs, to tackle this issue, Convolutional Neural Networks (CNNs) are applied to replace RNNs with position information provided by trained positional embeddings (Gehring et al., 2017). However, CNNs can only utilize contexts within a pre-specified window, and the multi-head attention machinery is proposed as part of the Transformer translation model to enable modeling over the whole sequence while ensuring independent evolution of token representations and enabling efficient parallelization (Vaswani et al., 2017). The success of the Transformer which achieves state-of-the-art translation performance, has received wide attention from both academia and industry.

That said, is the strong Transformer translation model good enough and "all we need"? Apparently, there is room for improvement: the model still cannot fully translate all translation queries correctly. The main aim of the research presented in this thesis is to improve the translation quality of the Transformer model.

We describe our research approaches in two complementary ways: (i) the model architecture designed to enhance the ability of the Transformer in capturing long-distance relationships, and (ii) improving the optimization (training) of the Transformer, including parameter initialization and dynamic batch size selection.

Specifically, in the first part of our research, we focus on learning and utilizing additional phrase representations of source sentences into the model architecture (described in Chapter 3) to enhance its ability in long-range dependency learning, under the motivation that the model may perform token-level attention better after attending the phrase representation sequence which is shorter than the corresponding token representation sequence, especially when capturing long-distance dependencies.

Tai et al. (2015) show that the Long Short-Term Memory Network (LSTM) handles short sentences better than long sentences. In Linzen et al. (2016)'s subject-verb number agreement prediction task which predicts the number of the following verb (plural or singular) of the sentence, the accuracy of LSTM degrades consistently with increasing distances between the subject and the verb. Yang et al. (2017) show that it is challenging for the LSTM-based NMT model to capture long-distance dependencies. For assessing the ability of NMT models to capture long-distance relations, Tang et al. (2018) examine the performance of RNNs, CNNs and the Transformer on the Subject-Verb Agreement (SVA) task which is the most popular choice for evaluating the ability to capture long-range dependencies and has been used in many studies (Linzen et al., 2016; Sennrich, 2017). They show that the Transformer models which connect distant tokens via shorter network paths than RNNs are not particularly stronger than RNN models for long distances, and that the number of heads in multi-head attention is crucial for its performance over long distances. Yang et al. (2019b) show that the accuracies of encoders, including both the Transformer self-attentional encoder and the Gated Recurrent Unit (GRU) (Cho et al., 2014) encoder, decrease on long-distance cases across language pairs and they observe the same for model variants on the word reordering detection task, which also suggests that both GRU and the Self-Attention Network (SAN) fail to fully capture long-distance dependencies. This leads us to our first Research Question (RQ).

RQ1: *How to improve the ability of the Transformer in long-distance relation capturing?*

In Chapter 3, we provide a solution to this issue. Considering that modeling phrases instead of words has significantly improved Statistical Machine Translation (SMT) through the use of larger translation blocks ("phrases") and that it has also improved its reordering ability, modeling NMT at phrase level is an intuitive proposal to help the model capture long-distance relationships better. Thus, we propose to let the Transformer utilize phrase representations in addition to token representations.

However, there are many more potential phrases than tokens, and the phrase table is magnitudes larger than the word vocabulary. For NMT due to memory limitations it

is not possible to put phrase embeddings directly into GPUs, and the distribution over phrases is much sparser than that over words, which may lead to data sparsity and hurt the performance of NMT. This raises our second research question:

RQ2: *How to avoid the potentially large phrase table while benefiting from phrase representations?*

To address the large phrase table issue, we propose an attentive feature extraction model and generate phrase representation based on token representations on the fly (in Chapter 3). Specifically, our model first summarizes the representation of a given token sequence with the vanilla mean- or max-over-time pooling operation, then computes the attention weight of each token based on the original token representation and the summarized phrase representation with a Feed-Forward Neural Network (FFNN), and generates the final phrase representation by a weighted combination of the token representations in the phrase after normalizing the attention weights. The main trick is that we do not use phrases as blocks in translations: the translation is still word-based, but our attention model allows the representation of words to be informed by the phrases which are computed on the fly.

After proposing the approach to generating phrase representations based on token representations, the third research question is then:

RQ3: *How to learn and utilize phrase representations in the Transformer translation model?*

In Chapter 3, we also show how we change the design of the Transformer translation model to incorporate the learning and utilization of phrase representations into the model, allowing all aspects of the model to be jointly trained together in an end-to-end manner. Specifically, we propose an attentive combination network that attends phrase representations, and insert the attentive combination network into each encoder layer and decoder layer of the Transformer translation model to let each token pay attention to phrases before paying attention to the original source token representations. Note that in our model phrases are only computed for source input. They are used in both the encoder

and the decoder, but the decoder only has access to source phrases in a cross-attention style manner. We do not compute target side output phrases in the decoder.

The research addressing RQ 1, 2 and 3 presented in Chapter 3 has been published in Xu et al. (2020c) at ACL 2020.

In the second part of our research, we focus on the convergence and the effects of batch sizes and automatic determination of dynamic batch sizes on the optimization of the Transformer (described in Chapter 4 and 5, respectively).

As regards the topic of the convergence of the Transformer, Neural Networks (NN) are normally first randomly initialized, then trained to minimize a pre-defined loss function, and ensuring the convergence of neural models is crucial for benefiting from their performance. A model which fails to converge cannot make meaningful predictions. However, for the optimization of the Transformer, especially for deep Transformers, previous research (Bapna et al., 2018) shows that even though the Transformer translation model employs residual connections and layer normalization to ease the optimization difficulties caused by its deep multi-layer encoder/decoder structure, vanilla deep Transformers still have difficulty in training, and particularly Transformer models with more than 12 encoder/decoder layers fail to converge. Bapna et al. (2018) propose a Transparent Attention (TA) mechanism which combines weighted outputs of encoder layers for each decoder layer rather than only taking the output of the last encoder layer. Their approach brings the most improvements with a Transformer with 16 encoder layers. But in Computer Vision (CV) tasks, residual connection has shown its strong ability in ensuring the convergence of deep models with more than a hundred layers. Why does it fail with deep Transformers? This leads to our fourth research question:

RQ4: *Why do Transformers, specifically deep Transformers, have difficulty in converging even with layer normalization and residual connections?*

In Chapter 4, we first empirically demonstrate that a simple modification made in the official implementation (Vaswani et al., 2018) which changes the computation order of residual connection and layer normalization, can significantly ease the optimization of

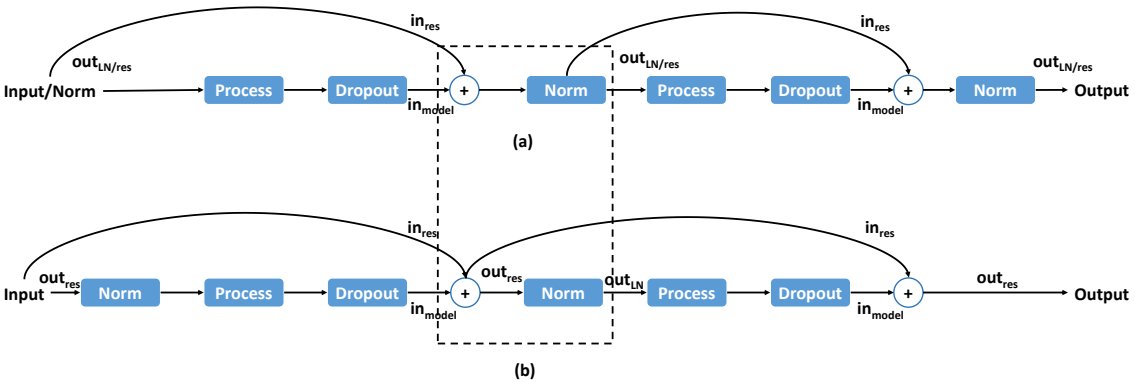


FIGURE 2: Two computation sequences of Transformer translation models: (a) the one used in the original paper, (b) the official implementation.

deep Transformers. Specifically, the original paper (Vaswani et al., 2017) computes in the order of *processing* → *dropout* → *residual connection* → *layer normalization* where processing indicates the computing of multi-head attention or position-wise feed-forward neural network, while the official implementation computes in the order of *layer normalization* → *processing* → *dropout* → *residual connection*. Figure 2 shows the two kinds of computation orders. We suggest to regard the output of layer normalization ($out_{LN/res}$) as the output of residual connection rather than the addition of in_{res} and in_{model} for Figure 2 (a), because it ($out_{LN/res}$) is the input (in_{res}) of the next residual connection computation. We then perform a theoretical analysis based on the difference between computation orders, which points out that layer normalization over residual connections in Figure 2 (a) may effectively reduce the impact of residual connections due to subsequent layer normalization, in order to avoid a potential explosion of combined layer outputs (Chen et al., 2018b). The analysis raises the fifth research question:

RQ5: *How to prevent layer normalization from shrinking residual connections?*

In Chapter 4, we present a parameter initialization method that leverages the Lipschitz constraint on the initialization of Transformer parameters to prevent the layer normalization shrinking residual connections that effectively ensures training convergence of deep Transformers. In contrast to findings in previous research, we further demonstrate that with Lipschitz parameter initialization, deep Transformers with the *original* computation order can converge, and obtain significant BLEU improvements with up to 24 layers.

The research addressing for RQ 4 and 5 presented in Chapter 4 has been published in Xu et al. (2020a) at ACL 2020.

The performance of neural models is likely to be affected by the choice of hyperparameters. While much previous research (Sutskever et al., 2013; Duchi et al., 2011; Kingma and Ba, 2015) focuses on accelerating convergence and reducing the effects of the learning rate, comparatively few papers concentrate on the effect of batch size. However, the batch size has been found to be an important hyperparameter for the performance of the Transformer, and some batch sizes empirically lead to better performance than the others. Specifically, it has been shown that the performance of the Transformer model (Vaswani et al., 2017) for Neural Machine Translation (Bahdanau et al., 2014; Gehring et al., 2017) relies heavily on the batch size (Popel and Bojar, 2018; Ott et al., 2018; Abdou et al., 2017; Zhang et al., 2019a), and a larger batch size normally leads to better performance. The influence of batch size on performance raises the sixth research question:

RQ6: *How to dynamically and automatically find proper and efficient batch sizes during training?*

To address this research question, we suggest to observe the gradients during training as they point out the magnitudes and directions for the optimization, and investigate the relationship between the batch size and gradients. Specifically, we observe the effects on gradients with increasing batch size in gradient accumulation which accumulates gradients of smaller mini-batches as the gradients of a larger mini-batch comprising these smaller mini-batches, and find that a large batch size stabilizes the direction of gradients. Based on our observation, we then propose to automatically determine dynamic batch sizes in training by monitoring the gradient direction change while accumulating gradients of small batches, by accumulating gradients of smaller mini-batches up to and until the gradient direction starts to fluctuate.

Unfortunately in practice, a Transformer model may have a large number of parameters, and the cost of computing the gradient direction change is relatively high, which may hamper the efficiency of training while incorporating the monitoring of gradient direction change to dynamically compute batch sizes. This leads to our seventh research question:

RQ7: *How to efficiently monitor gradient direction change?*

To tackle this issue, we propose to divide model parameters into groups, and monitor gradient direction change only on a selected group which are sensitive to the batch size hyperparameter in each optimization step. Specifically, we record the reduction of gradient direction change of the monitored parameter group in each optimization step. We suggest the parameter group with more reduction is more sensitive to the batch size than the others with less reduction. Thus we normalize the reductions of all parameter groups and use them as sampling probabilities of corresponding parameter groups.

Our research addressing RQ 6 and 7 presented in Chapter 5 has been published in Xu et al. (2020b) at ACL 2020.

In order to support our research, we develop the Neutron implementation of the Transformer model and its several variants. We present our engineering work in Chapter 6. Specifically, to build on and compare our approaches, we implement some recent advanced research related to the Transformer in addition to the standard implementation of the Transformer while we have developed our ways towards improving the Transformer, including the Average Attention Network (AAN) to accelerate the decoding of the Transformer (Zhang et al., 2018a), Hierarchical Layer Aggregation to fuse shallow layers (Dou et al., 2018), the use of a recurrent decoder (Chen et al., 2018b), the modeling of sentential context (Wang et al., 2019e) for improving the MT quality, Transparent Attention (Bapna et al., 2018) to ensure the convergence of deep encoders and the document-level Transformer (Zhang et al., 2018c). We also support the efficient training scheduler (dynamic sampling and review mechanism) proposed by Wang et al. (2018c), and other advanced features like new optimizers (e.g., RAdam (Liu et al., 2020), Lookahead (Zhang et al., 2019c) and their combination). Implementations of all our approaches in this thesis are also included and open-sourced in the toolkit.

Acknowledgements

I first acknowledge the support of **China Scholarship Council** for a doctoral scholarship ([2018]3101, 201807040056) that covers my living expenses with many thanks. Without this funding support, I cannot experience these great and wonderful years of research in Germany. The scholarship does never put any restrictions on my research topic, which enables me to research any topics I am interested in. It is ideal from this point of view.

Second but also very important, I express my sincere gratitude to my supervisor **Prof. Dr. Josef van Genabith** and my co-supervisor **Prof. Dr. Deyi Xiong** (Tianjin University). Josef offers me the great chance to follow him as a Ph.D. candidate and supports me in all aspects he is able to with his best efforts. Josef offers me the PhD candidate chance even though at the time I did not yet have any top-tier publications, educates and leads me to achieve what I dreamed of in these years. He takes good care of me, and introduces me to the DFKI (Deutsches Forschungszentrum für Künstliche Intelligenz, German Research Center for Artificial Intelligence) as a junior researcher, where I get an international research view. DFKI also provides me with GPU servers to run experiments. Josef also kindly covers my semester fee and costs for attending top-tier conferences. For my research, Josef devotes much of his valuable time to discussing my ideas with me and helping me improve the writing of papers with great patience. Deyi also pays a lot of attention to my work. He helps me improve our publications, supports me with valuable discussions, stands for me to win support from the other researchers of the community, and helps very much in paving my way towards a bright future. Without their guidance, I do not think that I can achieve my goal (to complete some nice and valuable work that produces good publications) here in such a pleasant way without pressure.

Additionally, I express my thanks to **Dr. Jingyi Zhang**, an excellent researcher at DFKI. She discusses my research with me in the view of a strict reviewer, telling me why she will reject my papers, which greatly helps me polish and refine my work to make them logically compelling and self-contained.

I sincerely appreciate **Prof. Dr. Qun Liu**, who is currently the chief scientist at Huawei. He selflessly recommended me to **Prof. Dr. Josef van Genabith** and to DFKI. Without his recommendation, I cannot find my way to my doctoral research. I also thank my postgraduate supervisor **Prof. Dr. Hongying Zan** who introduced me to **Prof. Dr. Qun Liu** and **Prof. Dr. Deyi Xiong**.

I thank authors of related publications, who show me the way to conduct my research and inspire me, and developers and maintainers of the other related popular Natural Language Processing (NLP) toolkits, their experience, outcomes and advice reduce my efforts while implementing our NMT project, the Neutron implementation of the Transformer and its variants.

I thank my colleagues at Saarland University and DFKI for their help. Working with them is pleasant. I also thank our supporters, collaborators and anonymous reviewers for their efforts in helping us improve our work.

Last but not least, I thank my family, my parents (Mr. Zhixiao Xu and Ms. Jinqin Wang) and my wife (Ms. Qiuhui Liu). It is not easy for my parents in their $50s$ and my wife to support me to pursue my dream more than $8,000$ kilometers away in the traditional Chinese custom. Though they miss me so much, they believe in me and insist on me not to give up at all times when I want to until I get the offer from Josef. They have the magic to keep me in a good mood. Qiuhui researches both NLP and recommendation systems. Even though her own work in China is already quite busy, she continuously contributes to my work regardless of working days or holidays, including but not restricted to discussion of research ideas, helping implement, debug new ideas and maintain our NMT toolkit. Sometimes, she completes her own work and returns home later than 9 p.m., but still collaborates with me until 11 p.m. Her support starts years before our marriage. I also thank our baby, Yuanfeng Xu. I got my papers accepted by ACL and IJCAI after my wife was pregnant. Perhaps she brings us lots of good luck :) My family makes me aware that though a Ph.D. degree together with past and future research work might be the most wonderful decorations for the meaning of my life, they are not everything. The other parts of our life, e.g., our existence, well-being, various kinds of experiences and happiness, are also very important.

Contents

List of Figures

List of Tables

Abbreviations

AAN	**A**verage **A**ttenion **N**etwork
AI	**A**rtificial **I**ntelligence
BLEU	**Bi**-**L**ingual **E**valuation **U**nderstudy
CNN	**C**onvolutional **N**eural **N**etwork
CV	**C**omputer **V**ision
DCDL	**D**ynamic **L**inear **C**ombination of **L**ayers
DiSAN	**Di**rectional **S**elf-**A**ttention **N**etwork
DNN	**D**eep **N**eural **N**etwork
DS-Init	**D**epth-**S**caled **Init**ialization
ENAS	**E**fficient **N**eural **A**rchitecture **S**earch
FFNN	**F**eed-**F**orward **N**eural **N**etwork
GELU	**G**aussian **E**rror **L**inear **U**nit
GPU	**G**raphic **P**rocessing **U**nit
GRU	**G**ated **R**ecurrent **U**nit
HMM	**H**idden **M**arkov **M**odel
HTER	**H**uman-targeted **T**ranslation **E**dit **R**ate
LAU	**L**inear **A**ssociative **U**nit
LSTM	**L**ong-**S**hort **T**erm **M**emory
MAtt	**M**erged **Att**ention
MT	**M**achine **T**ranslation
NAS	**N**eural **A**rchitecture **S**earch
NLP	**N**atural **L**anguage **P**rocessing
NMT	**N**eural **M**achine **T**ranslation
NN	**N**eural **N**etwork
ON-LSTM	**O**rdered **N**eurons - **L**ong-**S**hort **T**erm **M**emory

PBSMT	**P**hrase **B**ased **S**tatistical **M**achine **T**ranslation
PoS	**P**art-**o**f-**S**peech
ReLU	**Re**ctified **L**inear **U**nit
RBMT	**R**ule-**B**ased **M**achine **T**ranslation
RNN	**R**ecurrent **N**eural **N**etwork
SAN	**S**elf-**A**ttenion **N**etwork
SMT	**S**tatistical **M**achine **T**ranslation
SVA	**S**ubject-**V**erb **A**greement
TA	**T**ransparent **A**ttenion
TER	**T**ranslation **E**dit **R**ate
UI	**U**ser **I**nterface

To my Family...

Chapter 1

Introduction

The main purpose of the research presented in this thesis is to improve the translation quality of current NMT engines which transform texts between different human languages, aiming to significantly help users speaking different languages exchange information at a low cost. Improved Machine Translation (MT) quality also reduces the additional efforts of human translators or post-editors in computer aided translation scenarios.

The work presented in this thesis specifically aims at improving the state-of-the-art Neural Machine Translation (NMT) model, the Transformer translation model (Vaswani et al., 2017).

Before the Transformer, sequence-to-sequence neural models were mainly based on stacking of recurrent or convolutional layers in an encoder-decoder configuration, with the attention mechanism to align source tokens and target tokens for NMT (Bahdanau et al., 2014; Gehring et al., 2017). However, to collect information from contexts recurrent models have to compute in a token-by-token manner, which prevents them from parallelization across the whole sequence. In comparison, convolutional models can only use contexts in a pre-defined window size for word sense disambiguation. As a result, they lack the ability to use the whole source sentence. The Transformer (Vaswani et al., 2017) which relies entirely on the multi-head attention mechanism advances over both RNNs and CNNs, as it is able to model dependencies between tokens over the whole sequence regardless of their distance, and can be computed in parallel on modern GPUs efficiently.

Even though the widely employed Transformer performs surprisingly well and has attracted wide attention from both the research community and industry, it is still not perfect. For example, Tang et al. (2018) examine the ability of sequence-to-sequence neural models on the subject-verb agreement task (where capturing long-range dependencies is required), and find that although intuitively the multi-head attention machinery connects distant words via shorter network paths than RNNs, self-attentional networks do not outperform RNNs in modeling subject-verb agreement over long distances. Bapna et al. (2018) find that even with residual connections and layer normalization adopted, Transformers with deep encoders suffer from convergence issues, and particularly Transformer models with more than 12 encoder layers fail to converge in their experiments.

In order to improve the performance of the Transformer translation model, we describe our research approaches in this thesis in two areas: (i) improving the long-distance dependency learning ability of the Transformer by learning and incorporating source phrase representations, and (ii) improving the optimization of the Transformer through a proper parameter initialization approach under the Lipschitz constraint and dynamically adjusted training batch sizes by monitoring gradient direction change during gradient accumulation.

Specifically, in the first part of our research described in Chapter 3, we concentrate on improving the long-distance relation capturing ability of the Transformer translation model by learning source phrase representations of source sentences and utilizing the learned phrase representations in addition to token representations. Based on the fact that modeling phrases instead of words has significantly improved the performance, especially the reordering ability, of Statistical Machine Translation (SMT) approaches through the use of larger translation blocks ("phrases"), modeling NMT at phrase level is an intuitive but non-trivial proposal to help the model better capture long-distance relationships.

In the second part of our research, we focus on the optimization issues of the Transformer, specifically, its parameter initialization and dynamic determination of batch sizes during training. For parameter initialization, we first empirically compare the different behaviors in the convergence of two computation orders between layer normalization and residual connection of deep Transformers, then conduct a theoretical analysis of the interaction between layer normalization and residual connection, and propose to initialize Transformers under the Lipschitz constraint which ensures their convergence (described in Chapter

4). For dynamically determining batch sizes during the training of the Transformer, we observe the direction change of gradients during gradient accumulation, and propose to perform optimization steps when it starts to fluctuate (described in Chapter 5).

Over a lot of NLP research with neural models on a wide range of tasks, the long-distance dependency learning ability of neural networks has been widely examined, and a common finding is that neural models normally perform better on short sentences than on long sentences, which indicates that capturing long-distance relation can be challenging for neural approaches. Tai et al. (2015) show that the Long Short-Term Memory Network (LSTM) handles short sentences better than long sentences. In Linzen et al. (2016)'s subject-verb number agreement prediction task which predicts the number of the following verb (plural or singular) of the sentence, the accuracy of LSTM degrades consistently with increasing distances between the subject and the verb. Yang et al. (2017) show that it is challenging for the LSTM-based NMT model to capture long-distance dependencies. For assessing the ability of NMT models in long-distance relation capturing, Tang et al. (2018) examine the performance of RNNs, CNNs and the Transformer on the subject-verb agreement task which is the most popular choice for evaluating the ability to capture long-range dependencies and has been used in many studies (Linzen et al., 2016; Sennrich, 2017). They show that the Transformer models which connect distant tokens via shorter network paths than RNNs are not particularly stronger than RNN models for long distances, and that the number of heads in multi-head attention is crucial for its performance over long distances. Yang et al. (2019b) show that the accuracies of encoders, including both the self-attentional encoder and the GRU (Cho et al., 2014) encoder, decrease on long-distance cases across language pairs and that variants of the models do the same on the word reordering detection task, which also suggests that both GRU and the self-attention network fail to fully capture long-distance dependencies.

In the evaluation of long-distance relation modeling ability specialized to the Transformer, Tang et al. (2018) show that although intuitively the attentional network employed by the Transformer can connect distant words via shorter network paths than RNNs, it does not significantly outperform RNNs in their empirical analysis on the subject-verb agreement task (Sennrich, 2017), and handling long-distance dependencies is still challenging for the Transformer. This leads us to our first Research Question (RQ).

RQ1: *How to improve the ability of the Transformer in long-distance relation capturing?*

Considering that modeling phrases instead of just words has significantly improved the SMT approach through the use of larger translation blocks ("phrases") and that it has also improved its reordering ability (Koehn et al., 2003; Och and Ney, 2004), we propose to model NMT at phrase level in addition to the token level to help the Transformer capture long-distance relationships. In Chapter 3, we describe our approach to using phrase representations in the Transformer. We also provide experiment results and analysis of comparisons between the performance of the Transformer with phrase representations and that without phrase representations in both the WMT 14 English-German news translation task and the subject-verb agreement task (Sennrich, 2017; Tang et al., 2018). Specifically, we show that using phrase representations brings about more BLEU improvements over the baseline Transformer on long sentences than on short sentences, and leads to significant improvements in the subject-verb agreement accuracy when the distances between the subject and the verb are long. Our analysis in Chapter 3 shows the positive effects of incorporating phrase representations in long-distance dependency capturing and machine translation.

Dahlmann et al. (2017) suggest that SMT usually performs better in translating rare words and profits from using phrasal translations, and introduce a hybrid search algorithm for attention-based NMT which extends the beam search of NMT with phrase translations from SMT. Wang et al. (2017d) incorporate SMT into NMT through utilizing recommendations from SMT in each decoding step of NMT to address the coverage issue and the unknown word issue of NMT. Wang et al. (2017e) propose to translate phrases in NMT by integrating target phrases from an SMT system with a phrase memory. We suggest our research based on the Transformer model is different from most previous work focusing on utilizing phrases from SMT in NMT to address its coverage problem (Tu et al., 2016) of the at the time RNN-based NMT.

Our approach to learning phrase representations and integrating them into Transformer-based NMT is non-trivial. Using phrase embeddings in a similar way like using word embeddings is impossible, because there are a magnitude more phrases than tokens in the bilingual parallel corpus, which leads to two main issues which prevent NMT from directly using phrases:

- There are a magnitude more phrases than tokens, and the phrase table is much larger than the word vocabulary. The corresponding embedding matrix is not affordable

for GPU memories.

- Distribution over phrases and co-occurrences between them is much sparser than that over words, which may lead to a data sparsity issue and hurt the training of phrase embeddings. The data may not provide sufficient usage examples of some less frequent phrases for the learning of their high-dimensional dense embeddings.

These two issues raise our second research question.

RQ2: *How to avoid the potentially large phrase table while benefiting from phrase representations?*

In Chapter 3, we address this large phrase table issue. Given that phrases are composed of tokens, our proposal is to generate phrase representations from their corresponding token sequences "on the fly". We first employ the simple mean- and max-over-time pooling approaches, then propose an attentive phrase representation generation algorithm, as simply merging several token vectors into one is very likely to incur information loss, and introducing an importance evaluation mechanism is better than treating tokens equally. To highlight the most important features in a phrase, our attentive phrase representation generation model learns to score tokens differently according to their importance in the phrase. The model first roughly extracts features from all tokens into a vector with the naive mean- or max-over-time pooling approach, then assigns a score to each token by comparing each token vector with the extracted feature vector, and produces the weighted accumulation of all token vectors as the phrase representation according to their scores. We empirically validate the performance of different approaches in Chapter 3 in the ablation study on the WMT 14 English-German news translation task.

After addressing the large phrase table issue by generating phrase representations "on the fly" based on corresponding token representations, the remaining question is how to integrate the learning of phrase representations into the Transformer translation model and to enable the Transformer to benefit from phrase representations in Translation. This is our third research question.

RQ3: *How to learn and utilize phrase representation in the Transformer translation model?*

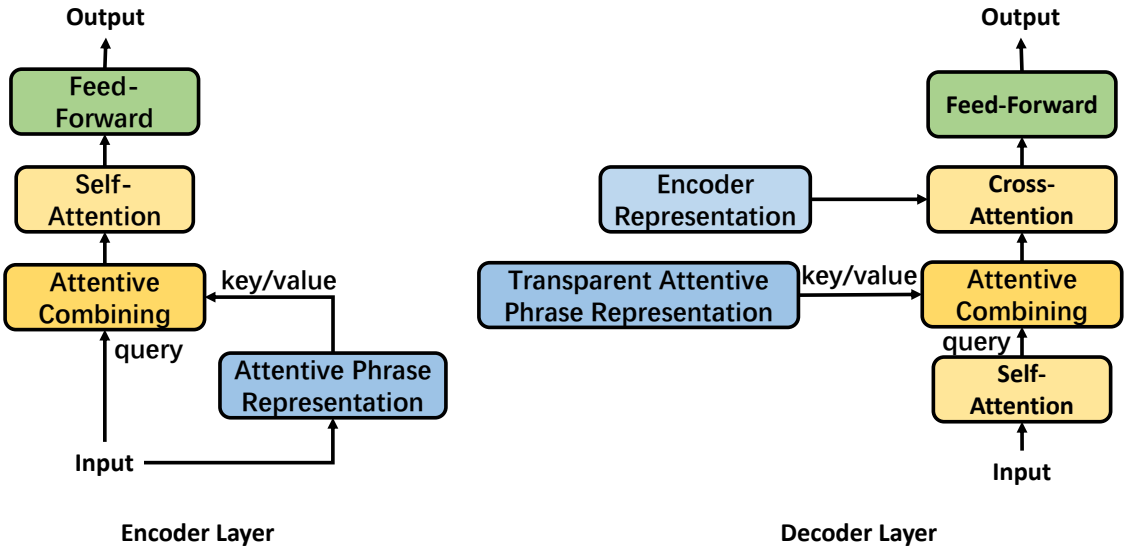


FIGURE 1.1: The encoder/decoder layer of the Transformer model with phrase representation. Residual connection and layer normalization are omitted for simplicity.

In Chapter 3, we propose an attentive combination network to incorporate the phrase representation into the Transformer translation model to aid it modeling long-distance dependencies. The attentive combination network is inserted into each encoder layer and each decoder layer to bring in information from phrase representations. Thus all encoder and decoder layers can benefit from phrase representations.

For the encoder layer with phrase representations, the new computation order is cross-attention from source tokens to source phrases → self-attention over tokens → feed-forward neural network to process collected features, while for a decoder layer it is: self-attention over decoded tokens → cross-attention to source phrases → cross-attention to source tokens → feed-forward neural network to process collected features. Compared to the computation order of the standard Transformer, the new computation order performs additional attending at phrase level before attending source token representations at the token level. This bakes in phrase information into the token representations. We conjecture that attending at phrase level should be easier than at token level, and attention results at phrase level may aid the attention computation at the token level. The structures of the Transformer encoder layer and the decoder layer with phrase representations are shown in Figure 1.1.

Our Transformer with phrase representations described in Chapter 3 outperforms the vanilla Transformer in multiple evaluations. On the WMT 14 English-German and

English-French news translation tasks, we obtained +1.29 and +1.37 BLEU improvements respectively on top of the strong Transformer Base baseline, which demonstrates the effectiveness of our approach. Our approach helps Transformer Base models perform at the level of Transformer Big models on the English-German task, and even significantly better for long sentences, but with substantially fewer parameters and training steps. It also shows its effectiveness with the Transformer Big setting.

It is worth noting that the WMT 14 English-French task provides a larger dataset ($\sim 36M$ sentence pairs) and achieves a higher baseline BLEU than the English-German task. We suggest that the significant improvements (+1.09 BLEU) obtained by our approach on the English-French task with the Transformer Big setting (a large amount of data plus a very large model) support the effectiveness of our approach in challenging settings (production scenarios).

We also conducted a length analysis, and the results show that our approach to incorporating phrase representations into the Transformer brings more gains with long sentences than with short sentences, which supports our conjecture that phrase representation sequences can help the model capture long-distance relations better, as intuitively, in translating long sentences we should encounter more long-distance dependencies than in short sentences. To further measure the capability of the NMT model to capture long-distance dependencies, we conducted a linguistically-informed verb-subject agreement analysis on the *Lingeval97* dataset (Sennrich, 2017) following Tang et al. (2018). The evaluation results (in Table 3.5) also show that our approach can improve the accuracy of long-distance subject-verb dependencies, especially for cases where there are more than 10 tokens between the verb and the corresponding subject, given that the Transformer Base with phrase representations outperforms the vanilla Transformer Big which has twice the number of heads in multi-head attention networks as that of the Transformer Base (Tang et al. (2018) suggest that increasing the number of attention heads improves the ability of the Transformer in capturing long-distance dependencies). Thus, we suggest that our approach improves the ability of the model, especially in handling long-distance relations and translating long sentences.

Neural Networks (NN) are normally first randomly initialized, then trained to minimize a predefined loss. Thus ensuring the convergence of neural models is crucial for benefiting from their performance. However, for the optimization of the Transformer, especially for

deep Transformers, previous research (Bapna et al., 2018) shows that even though the Transformer translation model employs residual connections and layer normalization to ease the optimization difficulties caused by its multi-layer encoder/decoder structure, deep Transformers still have difficulty in training, and particularly Transformer models with more than 12 encoder layers fail to converge. They propose the Transparent Attention (TA) mechanism which individually aggregates the outputs of all encoder layers for each decoder layer rather than using only that of the last encoder layer to enrich the gradients to shallow encoder layers during backpropagation to further ensure the convergence of deep Transformers. They obtain the most significant improvements with the deep Transformer consisting of a 16-layer encoder and a 6-layer decoder on the WMT 14 English-German and WMT 15 Czech-English news translation tasks. To ensure the convergence of deep Transformer encoders, Wang et al. (2019c) propose an approach which additionally aggregates the features extracted from all preceding layers for all encoder layers based on the Dynamic Linear Combination of Layers (DLCL) mechanism. Wu et al. (2019c) propose a two-stage training strategy which "grows" a well-trained NMT model into a deeper network with three components specially designed to overcome the optimization difficulty and best leverage the capability of both shallow and deep architectures. In more recent work, Zhang et al. (2019a) attribute the convergence issue of deep Transformers to the fact that layer normalization over residual connections effectively reduces the impact of residual connections due to subsequent layer normalization. In order to avoid a potential explosion of combined layer outputs (Chen et al., 2018b), similar in spirit to what we do in our work, they propose a different layer-wise initialization approach to reduce the standard deviation before normalization.

However, how the interaction between layer normalization and residual connection impacts the convergence of the Transformer, especially for deep Transformers, has not been deeply studied before. This leads to our fourth research question.

RQ4: *Why do Transformers, specifically deep Transformers, have difficulty in converging even with layer normalization and residual connections?*

In Chapter 4, we deeply analyze this question based on empirical findings. We first demonstrate that in our experiments, with a simple modification made in the official implementation which changes the computation order of residual connection and layer normalization and thereby avoids normalization of residual connections, can significantly

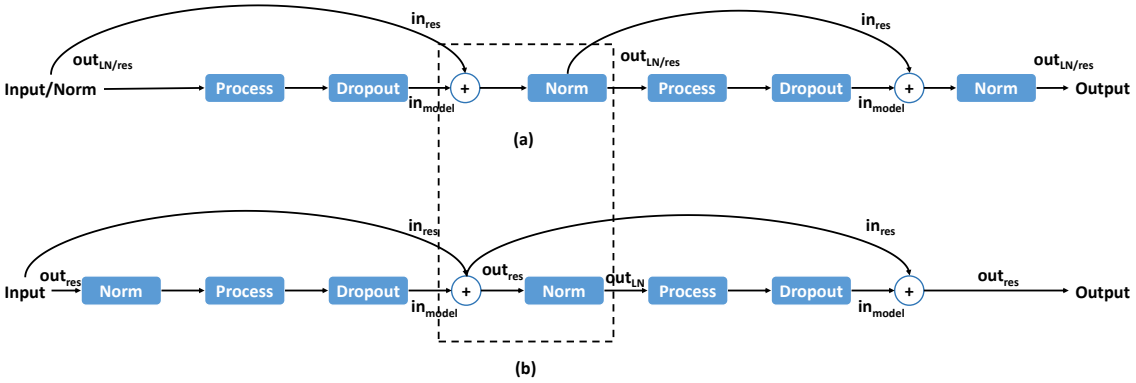


FIGURE 1.2: Two computation sequences of Transformer translation models: (a) the one used in the original paper, (b) the official implementation. We suggest to regard the output of layer normalization ($out_{LN/res}$) as the output of residual connection rather than the addition of in_{res} and in_{model} for (a), because it ($out_{LN/res}$) is the input (in_{res}) of the next residual connection computation.

ease the optimization of deep Transformers. Specifically, the original paper (Vaswani et al., 2017) computes in the order of *processing* → *dropout* → *residual connection* → *layer normalization* where *processing* indicates the computing of multi-head attention or position-wise feed-forward neural network, while the official implementation computes in the order of *layer normalization* → *processing* → *dropout* → *residual connection*. Figure 1.2 shows the two kinds of computation orders. We then compare the subtle differences in computation order in considerable detail, and attribute the convergence issue of deep Transformers to that layer normalization over residual connections in Figure 1.2 (a) effectively reduces the impact of residual connections due to subsequent layer normalization, in order to avoid a potential explosion of combined layer outputs (Chen et al., 2018b), which however, shrinks the gradients from residual connections during backpropagation. Based on findings from our analysis, we raise the natural question which is also our fifth research question.

RQ5: *How to prevent layer normalization from shrinking residual connections?*

To tackle this issue, we propose to constrain the standard deviation of input representations to the layer normalization to be smaller than 1. Thus the computation of layer normalization (shown in Table 4.2) will not shrink the residual connection. To achieve such a constraint, we further propose to initialize sub-models before layer normalization under the k-Lipschitz constraint, and theoretically prove that as long as $k \leq 1$, the goal to constrain the standard deviation can be satisfied with constrained input (described in Chapter 4).

In practice, our simple approach effectively ensures the convergence of deep Transformers with up to 24 layers, and achieves +1.50 and +0.92 BLEU improvements over the 6-layer baseline on the WMT 14 English to German task and the WMT 15 Czech to English task respectively. It is also worth noting that, unlike Zhang et al. (2019a), our parameter initialization approach does not degrade the translation quality of the 6-layer Transformer, and the 12-layer Transformer with our approach already achieves performance comparable to the 20-layer Transformer in Zhang et al. (2019a).

We also investigate the effects of deep decoders for the Transformer in addition to the deep encoders studied in previous works (Bapna et al., 2018; Wang et al., 2019c), and show that deep decoders can also benefit the Transformer.

Another issue related to the optimization of the Transformer is that the performance of the Transformer model (Vaswani et al., 2017) for NMT (Bahdanau et al., 2014; Gehring et al., 2017) relies heavily on the choice of batch sizes (Popel and Bojar, 2018; Ott et al., 2018; Abdou et al., 2017; Zhang et al., 2019a), and a larger batch size normally leads to better performance. Specifically, Popel and Bojar (2018) demonstrate that the batch size affects the performance of the Transformer, and a large batch size tends to benefit performance in their experiments with various but fixed batch sizes. Abdou et al. (2017) propose to use a linearly increasing batch size from 65 to 100, which slightly outperforms their baseline. Smith et al. (2018) show that the same learning curve on both training and test sets can be obtained by increasing the batch size during training instead of decaying the learning rate.

Although the choice of hyperparameters affects the performance of neural models, much previous research (Sutskever et al., 2013; Duchi et al., 2011; Kingma and Ba, 2015) focuses on accelerating convergence and reducing the effects of the learning rate. Comparatively few papers concentrate on the effect of batch size. The influence of batch size on performance raises the sixth research question.

RQ6: *How to dynamically and automatically find proper and efficient batch sizes during training?*

To address this research question, we take into consideration that gradients indicate the direction and the magnitude of parameter updates to minimize the loss function in training. We first investigate the relationship between increasing batch size and direction

k	1	2	3	4	5	6	7	8	9	10
Size	4064	8994	12768	17105	21265	25571	29411	33947	38429	43412
$a(g_0^{k-1}, g_0^k)$		51.52	30.37	27.42	22.61	20.87	19.80	19.59	18.92	19.23
$a(g_0^{k-3}, g_0^k)$				59.53	44.20	41.77	35.34	32.19	32.10	34.29

TABLE 1.1: The direction change of gradients while accumulating mini-batches.

change of gradients to reveal the effects of the increasing batch size on the gradient direction in optimization in Chapter 5. Specifically, we investigate the effects on the direction and change of direction of gradients with increasing batch size during gradient accumulation which incrementally sums up gradients of small mini-batches into that of a large mini-batch consisting of these mini-batches. Table 1.1 shows a typical example. In our study, we find that normally: (i) the gradient direction varies heavily at the beginning of the gradient accumulation, (ii) the gradient direction change reduces with increasing batch size, and (iii) eventually it will start fluctuating (here at k=10). Thus, we suggest that a large batch size stabilizes the direction of gradients.

Table 1.1 shows that the optimization direction is less stable with a small batch than with a large batch. But after the direction of gradients has stabilized, accumulating more mini-batches seems useless as the gradient direction starts to fluctuate.

Thus, we suggest to compute dynamic and efficient batch sizes by accumulating gradients of mini-batches, while evaluating the gradient direction change with each new mini-batch, and stop accumulating more mini-batches and perform an optimization step when the gradient direction fluctuates. The cumulative size of all mini-batches involved is then the batch size of the current training step. We suggest our approach to dynamically adjusting batch sizes during training is complementary to Sutskever et al. (2013); Duchi et al. (2011); Kingma and Ba (2015), as their approaches decide the magnitude of the movement in the optimization direction, while our approach provides a reliable gradient direction.

But in practice, a neural model, and specifically the Transformer model in our work, may have a large number of parameters, in which case the additional computational costs for monitoring the gradient direction change during the gradient accumulation of small

batches are relatively high. This prevents us from training efficiently and leads to our seventh research question.

RQ7: *How to efficiently monitor gradient direction change?*

In Chapter 5, we propose to address this issue by dividing model parameters into groups, and monitoring gradient direction change only on a dynamically selected group which is sensitive to the batch size in each optimization step. Thus the costs in both memory and computation for monitoring gradient direction change can be reduced by an order of magnitude. For a multi-layer model, i.e., the Transformer, a group may consist of parameters of one layer or several layers. In our approach, we regard parameters of an encoder layer or a decoder layer as a parameter group, in which case, the cost for monitoring only a parameter group is less than 1/10 of that for monitoring all parameters.

To sample the parameter group which is more sensitive to the batch size more frequently, we propose to record the angle reduction of gradient direction change during the gradient accumulation in optimization steps, and to normalize the average angle reduction of parameter groups during gradient accumulation into a probability distribution as sampling probabilities of corresponding parameter groups. Thus the parameter group which has a larger reduction in gradient direction change angle with the increasing batch size will be more frequently sampled.

In our experiments (described in Chapter 5) on the WMT English-German and English-French news translation tasks, we compared the results of our dynamic batch size approach to two fixed batch size baselines, the $25k$ batch size which is the empirical value of Vaswani et al. (2017) and the $50k$ batch size which is investigated by Zhang et al. (2019a). Our dynamic batch size approach yielding +0.73 and +0.82 BLEU improvements respectively over the fixed $25k$ batch size setting also outperforms the $50k$ batch size setting, while being more efficient than the $50k$ batch size with average batch sizes of only around $26k$ and $30k$ respectively.

For our research in this thesis, we also implement our own NMT toolkit, the Neutron implementation of the Transformer and its variants, and introduce it and its various features as our engineering work in Chapter 6. In addition to providing the basis of our implementations for the approaches presented in this thesis, we support many advanced features from recent cutting-edge research work, including the average attention decoder

(Zhang et al., 2018a), RNMT decoder (Chen et al., 2018b), transparent attention (Bapna et al., 2018), hierarchical layer aggregation (Dou et al., 2018), modeling sentential context (Wang et al., 2019e), context-aware NMT (Zhang et al., 2018c), etc. In addition to these we also include fundamental features, like label-smoothing loss, beam search, length-penalty, ensemble, gradient accumulation, averaging checkpoints, multi-GPU parallelization, etc.

1.1 Publications Resulting from the Research Presented in this Thesis

1.1.1 Chapter 3

- **Hongfei Xu**, Josef van Genabith, Deyi Xiong, Qiuhui Liu, and Jingyi Zhang. Learning Source Phrase Representations for Neural Machine Translation. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 386–396, Online, July 2020. Association for Computational Linguistics.

  Contributions: I am the first author of the paper, I developed the basic idea, I developed the implementation, I carried out the experiments and evaluations reported in the paper, I wrote each of the main drafts of the paper, I discussed and refined the ideas and the writeup of the papers with my co-authors. In this paper, (i) we propose an attentive feature extraction model and generate phrase representations based on token representations. Our model first summarizes the representation of a given token sequence with the mean- or max-over-time pooling operation, then computes the attention weight of each token by employing a 2-layer feed-forward neural network to assign a weight for the token given the token representation and the summarized representation, and generates the phrase representation by a weighted combination of token representations, (ii) to help the Transformer translation model better model long-distance dependencies, we propose to let both encoder layers and decoder layers of the Transformer attend the source phrase representation sequence which is shorter than the token sequence, in addition to the original token representation. Since the phrase representations are produced and attended at each encoder layer, the encoding of each layer is also enhanced with phrase-level attention computation, (iii) to the best of our knowledge, our work is the first to model

phrase representations and incorporating them into the Transformer to improve its long-distance dependency learning ability, and (iv) our approach empirically brings about significant and consistent improvements over the strong Transformer baseline in our experiments on the WMT 14 English-German (+1.29 and +1.11 BLEU for Base and Big settings respectively) and English-French (+1.37 and +1.09 BLEU respectively) news translation tasks. Our evaluation on the subject-verb agreement task demonstrates the effectiveness of our approach in improving the long-distance relation capturing ability of the Transformer.

1.1.2 Chapter 4

- **Hongfei Xu**, Qiuhui Liu, Josef van Genabith, Deyi Xiong, and Jingyi Zhang. Lipschitz Constrained Parameter Initialization for Deep Transformers. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 397–402, Online, July 2020. Association for Computational Linguistics.

  Contributions: I am the first author of the paper, I developed the basic idea, I developed the implementation, I carried out the experiments and evaluations reported in the paper, I wrote each of the main drafts of the paper, I discussed and refined the ideas and the writeup of the papers with my co-authors. In this paper, (i) we empirically demonstrate that a simple modification made in the Transformer's official implementation Vaswani et al. (2018) which changes the computation order of residual connection and layer normalization can effectively ease its optimization, (ii) we deeply analyze how the subtle difference of computation order affects convergence in deep Transformers, and point out that the convergence issue of deep Transformers in the computation order of the original paper (Vaswani et al., 2017) is because the layer normalization over residual connections may effectively reduce the impact of residual connections due to subsequent layer normalization, in order to avoid a potential explosion of combined layer outputs (Chen et al., 2018b), (iii) we propose to initialize deep Transformers under the Lipschitz constraint to prevent the layer normalization shrinking residual connections that effectively ensures training convergence of deep Transformers, (iv) in contrast to previous works, we empirically show that with proper parameter initialization, deep Transformers with the original computation order can converge. (v) Our simple approach effectively

ensures the convergence of deep Transformers with up to 24 layers, and achieves +1.50 and +0.92 BLEU improvements over the baseline on the WMT 14 English to German task and the WMT 15 Czech to English task, and (vi) we further investigate deep decoders for the Transformer in addition to the deep encoders studied in previous work, and show that deep decoders can also benefit the Transformer.

1.1.3 Chapter 5

- **Hongfei Xu**, Josef van Genabith, Deyi Xiong, and Qiuhui Liu. Dynamically Adjusting Transformer Batch Size by Monitoring Gradient Direction Change. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 3519–3524, Online, July 2020. Association for Computational Linguistics.

  Contributions: I am the first author of the paper, I developed the basic idea, I developed the implementation, I carried out the experiments and evaluations reported in the paper, I wrote each of the main drafts of the paper, I discussed and refined the ideas and the writeup of the papers with my co-authors. In this paper, (i) we propose to observe the effects of increasing batch size on the direction of gradients, and find that a large batch size stabilizes the direction of gradients, (ii) we propose to automatically determine dynamic batch sizes in training by monitoring the gradient direction change while accumulating gradients of small batches. Specifically, we suggest to compute dynamic and efficient batch sizes by accumulating gradients of mini-batches, while evaluating the gradient direction change with each new mini-batch, and stop accumulating more mini-batches and perform an optimization step when the gradient direction fluctuates, (iii) to measure gradient direction change efficiently with large models, we propose an approach to dynamically select those gradients of parameters/layers which are sensitive to the batch size. Our sampling approach significantly reduces the costs for monitoring gradient direction change, and (iv) in machine translation experiments on the WMT 14 English to German and English to French tasks, our approach improves the Transformer Base with a fixed $25k$ batch size by +0.73 and +0.82 BLEU respectively, while being significantly more efficient than the $50k$ batch size.

Chapter 2

Literature Survey

This chapter provides an overview of previous work related to the research presented in this thesis. Specific literature relevant to the topics of Chapter 3, 4, 5 and 6 in this chapter will be further described in more detail in the corresponding sections of these chapters.

2.1 Introduction

MT is an Artificial Intelligence (AI) problem aiming to translate between languages with computers. MT technologies have evolved from Rule-Based Machine Translation (RBMT) to Statistical Machine Translation (SMT) and modern Neural Machine Translation (NMT).

We first present an overview of research in machine translation in the order of the evolution of MT technologies in the following sections. We start with early RBMT technology in Section 2.2, followed by research in the SMT era (Brown et al., 1993; Koehn et al., 2003; Och, 2003; Chiang, 2007) in Section 2.3, and mainly focus on the more recent advances in NMT (Sutskever et al., 2014; Bahdanau et al., 2014; Gehring et al., 2017; Vaswani et al., 2017) in Section 2.4. Then we present several evaluation metrics for MT in Section 2.5.

2.2 Rule-Based Machine Translation

Early MT systems perform translation by replacing words or phrases in the source language with their target translations with a translation dictionary constructed in advance (Hutchins, 1995). However, such approaches do not perform any word reordering and word sense disambiguation using contexts, and the translation in the target language normally does not follow the same word order of source sentence in fact.

In order to improve translation quality, some local syntactic and morphological analysis has been introduced to perform local reordering before translation with the bilingual dictionary in Rule-Based Machine Translation (RBMT). Applying limited syntactic analysis in an RBMT system improves the fluency and readability of translations in the target language to some extent, but it is hard to design linguistic rules to cover all cases for reordering in translation well, especially when translating between "distant" languages. It is also very challenging to find a good solution for source word sense disambiguation with a limited local context. As a result, RBMT systems which may translate simple source sentences well often lack capabilities of both word reordering in the target language and word sense disambiguation in the source language important for translating long or complex source sentences.

Early RBMT systems rely heavily on well-developed dictionaries and linguistic rules which have to be constructed from scratch with very high manual efforts of linguistic experts for every translation system/domain between language pairs. The costs for individually building such systems for each use case are high.

The interlingua approach is proposed to extract a language-independent representation of the source language text capturing all information required to generate the appropriate target language translation (Hutchins, 1995). In theory, with a single language-agnostic representation (that works for all languages), the costs for building multiple translation systems, e.g., a multilingual system translating between many language pairs, can be reduced. The source sentence is first converted to an interlingual representation, and the translation is then produced from the interlingual representation. However, it is extremely difficult to create a language-independent representation detailed enough for all human languages, parse the source sentence into such a representation, and generate the target translation with it (Dorr et al., 2006).

2.3 Statistical Machine Translation

SMT models in general consist of a translation model which aligns between source tokens (phrases) and target tokens (phrases) and a target language model which aims to assess the fluency of target language strings.

SMT research starts with word-based models which estimate word alignments from a large volume of bilingual parallel data, but due to their limited capability in capturing long-range contexts well (the translation unit is an individual word), they often result in poor lexical selection and may fail to maintain phrasal cohesion between the phrases of source and target languages. The use of larger translation blocks ("phrases") evolves SMT to phrase-based models which improves the performance, especially the reordering ability of SMT approaches. Instead of the original formulation of the translation problem as a noisy-channel model in word-based models, Phrase-Based Statistical Machine Translation (PBSMT) employs a log-linear interpolation over a set of features. Phrase-based models can robustly perform translations localized to sub-strings frequently appearing in training data, but they lack the ability to capture the recursive structure of languages. Syntax-based SMT models are further proposed to model the syntax of source or target language for SMT.

Word-Based Models. Brown et al. (1993) describe a series of five statistical models (a.k.a. IBM 1 to 5) of the translation process and give algorithms for estimating the parameters of these models given a set of parallel sentence pairs to make use of the growing availability of bilingual, machine-readable texts and to extract linguistically valuable information from such texts. Vogel et al. (1996) address the problem of word alignments for a bilingual corpus in statistical translation, they use a first-order Hidden Markov Model (HMM) which has no monotonicity constraint for the possible word orderings for the word alignment problem to make the alignment probabilities dependent on the differences in the alignment positions rather than on the absolute positions. Och and Ney (2003) present different methods for combining word alignments to perform a symmetrization of directed statistical alignment models, and propose to measure the quality of an alignment model by comparing the quality of the most probable alignment, the Viterbi alignment, with a manually produced reference alignment. Liang et al. (2006) propose an unsupervised approach to symmetric word alignment in which two simple asymmetric

models are trained jointly to maximize a combination of data likelihood and agreement between the models. Dyer et al. (2013) propose a simple log-linear reparameterization of IBM Model 2 that overcomes problems arising from Model 1's strong assumptions and Model 2's over-parameterization, which provides efficient inference, likelihood evaluation and parameter estimation algorithms.

Phrase-Based SMT. Koehn et al. (2003) propose a phrase-based translation model and decoding algorithm. Their empirical comparison with several previously proposed phrase-based translation models suggests that the highest levels of performance can be obtained through relatively simple means: heuristic learning of phrase translations from word-based alignments and lexical weighting of phrase translations. Galley and Manning (2008) present a novel hierarchical phrase reordering model aimed at improving non-local reorderings to perform the kind of long-distance reorderings possible with syntax-based systems.

Syntax-Based Models. Wu (1997) introduces a novel stochastic inversion transduction grammar formalism for bilingual language modeling of sentence-pairs and the concept of bilingual parsing with a variety of parallel corpus analysis applications. Galley et al. (2006) construct a large number of derivations that include contextually richer rules, and account for multiple interpretations of unaligned words. They also propose probability estimates and a training procedure for weighting the very large rule sets. Liu et al. (2006) present a linguistic syntax-based translation model based on tree-to-string alignment templates which describe the alignment between a source parse tree and a target string. Xiong et al. (2006) propose a novel reordering model for phrase-based statistical machine translation that uses a maximum entropy model to predict reorderings of neighboring blocks (phrase pairs). Chiang (2007) presents a model that uses hierarchical phrases—phrases that contain subphrases. Huang and Chiang (2007) develop faster approaches for both phrase-based and syntax-based MT systems based on k-best parsing algorithms. Mi and Huang (2008) propose a forest-based approach that translates a packed forest of exponentially many parses, and encodes many more alternatives than standard n-best lists. Zhang et al. (2008) present a translation model based on tree sequence alignment which automatically learns aligned tree sequence pairs with mapping probabilities from word-aligned bi-parsed parallel texts. Shen et al. (2008) propose a novel string-to-dependency

algorithm for SMT which employs a target dependency language model during decoding to exploit long-distance word relations. Mi and Huang (2008) propose a novel approach to extract rules from a packed forest that compactly encodes exponentially many parses for translation rule extraction. Liu et al. (2009) propose a forest-based tree-to-tree model based on a probabilistic synchronous tree substitution grammar that uses packed forests. Chiang (2010) explores how to use source syntax and target syntax together.

2.4 Neural Machine Translation

Unlike SMT, NMT jointly models translation and target language reordering in an end-to-end manner with neural models. NMT research has many research topics, in the following paragraphs of this section, we will first introduce those most influential models (the evolution of baseline models in NMT research). Then, we will present some widely studied research topics in NMT including: model architecture, positional encoding, attention mechanisms, layer aggregation, knowledge integration, deep models, improving efficiency, robust NMT, back-translation, training of NMT models, non-auto-regressive translation, empirical studies, analysis of NMT models and context-aware NMT.

The Evolution of Model Architectures. To utilize powerful neural models for MT, Sutskever et al. (2014) propose to employ two LSTMs as encoder and decoder respectively for sequence-to-sequence MT, where the encoder encodes the source to a vector, and the decoder auto-regressively generates the corresponding translation in a token-by-token manner. Their simple approach performs comparably to previous PBSMT systems carefully tuned with many engineering efforts. As compressing the information of the source sentence into a fixed-dimension vector is very likely to incur information loss, Bahdanau et al. (2014) integrate the attention mechanism into the NMT decoder to jointly learn to translate and align. The attention mechanism attends the source encoding in every decoding step, which brings information from the source side and improves the translation quality especially for long sentences. Given that RNNs have to compute in a token-by-token manner which prevents RNN-based sequence-to-sequence from being efficiently parallelized on GPUs, Gehring et al. (2017) propose to use CNNs which evolve token representations independently rather than sequentially as in RNNs for NMT, which provides improved parallelization on GPUs. However, CNNs can only utilize contexts

of a fixed window for word sense disambiguation, which prevents them from capturing long-distance relations well. Vaswani et al. (2017) propose the Transformer based on the multi-head attention mechanism which is able to model over the whole sequence rather than contexts in a fixed window with CNNs for NMT while keeping the advantage of parallelization, and establish the new state-of-the-art.

Improving NMT Models. Many approaches have been explored to improve the performance of NMT by enhancing the design of its model architecture. Different from the vanilla encoder-decoder model that generates target translations from hidden representations of source sentences alone, Zhang et al. (2016) propose a variational model which introduces a continuous latent variable to explicitly model underlying semantics of source sentences and to guide the generation of target translations. Given that the vanilla sequence-to-sequence model lacks a mechanism to copy source fragments to the target, Gu et al. (2016) incorporate a copy mechanism into NMT which can choose subsequences in the input sequence and put them at proper places in the output sequence. Considering that the input to a neural sequence-to-sequence model is often determined by an upstream system, e.g., a word segmenter, Part-of-Speech (PoS) tagger, or speech recognizer, and these upstream models are potentially error-prone, Sperber et al. (2017) suggest that representing inputs through word lattices allows making this uncertainty explicit by capturing alternative sequences and their posterior probabilities in a compact form, and extend the Tree LSTM (Tai et al., 2015) into a Lattice LSTM that is able to consume word lattices for the NMT encoder. Tu et al. (2017) propose a novel encoder-decoder-reconstructor framework reconstructing the input source sentence from the hidden layer of the output target sentence. Wang et al. (2018d) show that the attention component can be effectively replaced by the neural Hidden Markov Model (HMM) consisting of neural network based alignment and lexicon models which are trained jointly using the forward-backward algorithm. Bahar et al. (2018) treat translation as a two-dimensional mapping and employ a multi-dimensional LSTM layer to define the correspondence between source and target words. He et al. (2018) explicitly coordinate the learning of hidden representations of the encoder and decoder together layer by layer, gradually from low level to high level. Shah and Barber (2018) introduce a latent variable architecture to model the semantics of the source and target sentences. Yang et al. (2019c) suggest Self-Attention Networks

(SANs) can be further enhanced by allowing the model to attend to information from different representation subspaces, and propose convolutional self-attention networks which offer SANs the abilities to strengthen dependencies among neighboring elements, and to model the interaction between features extracted by multiple attention heads. Concerning information aggregation, a common practice is to use a concatenation followed by a linear transformation, which may not fully exploit the expressiveness of multi-head attention, Li et al. (2019a) propose to improve the information aggregation for multi-head attention with the routing-by-agreement algorithm. Hao et al. (2019c) suggest that the Transformer model solely based on attention mechanisms lacks the ability of recurrence modeling, which hinders its further improvement of translation capacity, and propose to model recurrence for Transformers with an additional recurrence encoder. Given that either word-level or subword-level segmentations have multiple choices to split a source sequence with different word segmenters or different subword vocabulary sizes, Xiao et al. (2019) hypothesize that the diversity in segmentation may affect the NMT performance, and propose lattice-based encoders to explore effective word or subword representation in an automatic way during training. Sperber et al. (2019) also suggest that lattices are an efficient and effective method to encode ambiguity of upstream systems in NLP tasks, and introduce probabilistic reachability masks that incorporate lattice structure into the Transformer together with a method for adapting positional embeddings to lattice structures. Wang et al. (2019e) show that a shallow sentential context extracted from the top encoder layer only, can improve translation performance via contextualizing the encoding representations of individual words, and propose to exploit sentential context for NMT. Wang et al. (2019b) investigate a novel capsule network with dynamic routing for linear time NMT. Given that a hybrid of SANs and RNNs outperforms both individual architectures, Hao et al. (2019b) suggest that modeling hierarchical structure is an essential complementary between SANs and RNNs, and propose to further enhance the strength of hybrid models with an advanced variant of RNNs – Ordered Neurons LSTM (ON-LSTM) which introduces a syntax-oriented inductive bias to perform tree-like composition.

Positional Encoding. For CNN-based translation models and the Transformer, the position information of tokens relies heavily on positional encoding, and the approach for positional encoding can impact the translation performance. Shaw et al. (2018) extend the

self-attention mechanism with relative positional embeddings to efficiently consider representations of the relative positions, or distances between sequence elements. They describe an efficient implementation of their method and cast it as an instance of relation-aware self-attention mechanisms that can generalize to arbitrary graph-labeled inputs. Wang et al. (2019f) propose to augment SANs with structural position representations to model the latent structure of the input sentence in representations that are complementary to the standard sequential positional representations. They use dependency trees to represent the grammatical structure of a sentence, and propose two strategies to encode the positional relationships among words in dependency trees. Considering that the reordering model plays an important role in PBSMT, Chen et al. (2019a) propose a reordering mechanism to learn the reordering embedding of a word based on its contextual information. As positional embeddings only involve static order dependencies based on discrete numerical information, which is independent of word content, Chen et al. (2019b) propose a recurrent positional embedding approach to encode word content-based order dependencies with an RNN and integrate them into the multi-head attention model as independent heads or part of each head. Wang et al. (2020) generalize word embeddings, previously defined as independent vectors, to continuous word functions over their positions for modeling both the global absolute positions of words and their order relationships.

Attention Mechanisms. Cross-attention networks align between source tokens and target tokens, and their ability to produce correct alignments is crucial to generating accurate translation. In Transformer variants, self-attention networks are crucial for context modeling and word sense disambiguation. Mi et al. (2016) improve the attention of NMT by utilizing the alignments (human-annotated data or machine alignments) of sentence pairs in the training data and minimizing the distance between the machine attention matrices and the "true" alignments of training sentence pairs. Shen et al. (2018a) introduce the Directional Self-Attention Network (DiSAN) which is composed of a directional self-attention with temporal order encoded by masking. Miculicich Werlen et al. (2018) suggest that the target-side context in NMT is solely based on the sequence model, which is prone to a recency bias and lacks the ability to effectively capture non-sequential dependencies among words, and propose a target-side-attentive residual recurrent network for decoding, where attention over previous words contributes directly to the prediction of the next word. Li et al. (2018b) suggest the attention model used to identify the aligned source

words for a target word (target foresight word) in order to select translation context does not make use of any information of this target foresight word at all, and propose a new attention model enhanced by the implicit information of the target foresight word. Yang et al. (2018a) propose to model localness for self-attention networks to enhance the ability of capturing useful local context, given that SANs have proven to be of profound value for their strength of capturing global dependencies. They cast localness modeling as a learnable Gaussian bias which indicates the central position and scope of the local region to be paid more attention. Lin et al. (2018) suggest that the conventional attention mechanism, which treats the decoding at each time step equally with the same matrix, is problematic since the softness of the attention for different types of words (e.g., content words and function words) should differ, and propose to control the softness of attention with an attention temperature. Shankar et al. (2018) show that a simple beam approximation of the joint distribution between attention and output is an easy, accurate and efficient attention mechanism for NMT. Their method combines the advantage of sharp focus in hard attention and the implementation ease of soft attention. Yang et al. (2019a) suggest that SANs calculate the dependencies between representations without considering the contextual information which has proven useful for modeling dependencies among neural representations in various natural language tasks, and propose to incorporate the context representation into the transformations of the query and key of SANs. Shankar and Sarawagi (2019) present Posterior Attention Models that after a principled factorization of the full joint distribution of the attention and output variables in which the position where attention is marginalized is changed from the input to the output, and the attention propagated to the next decoding stage is a posterior attention distribution conditioned on the output. Xu et al. (2019) suggest that the hidden states of each word hierarchically calculated by attending to all words in the sentence, which assembles global information and takes all signals into account, may lead to overlooking neighboring information (e.g., phrase pattern), and further propose a hybrid attention mechanism using a gating scalar to dynamically leverage both local and global information. Given that in NMT, words are sometimes dropped from the source or generated repeatedly in the translation, Malaviya et al. (2018) address the coverage problem that changes only the attention transformation by allocating fertilities to source words to bound the attention they can receive, and propose a constrained sparsemax. Peters et al. (2019) suppose that dense alignments and strictly positive output probabilities resulting from the softmax transformation in both

attention mechanisms and the classifier respectively are wasteful, make models less interpretable and assign probability mass to many implausible outputs. As a result, they introduce entmax sparse attention which includes softmax and sparsemax as particular cases into sequence-to-sequence models. Correia et al. (2019) suggest that the standard softmax attention leads to a dense alignment matrix, and assigns a non-zero weight to all context words, and propose to replace softmax with sparse alpha-entmax in multi-head attention. Thus alignment matrices have flexible, context-dependent sparsity patterns. Indurthi et al. (2019) propose a hard-attention-based NMT model which selects a subset of source tokens for each target token to effectively handle long sequence translation.

Layer Aggregation. Advanced NMT models generally employ multi-layer encoders and decoders. However, usually only the top layers of the encoder and decoder are leveraged in the subsequent process, even though exploiting useful information embedded in other layers may benefit performance. Wang et al. (2018b) design three multi-layer representation fusion functions to fuse stacked layers. Dou et al. (2018) propose to use outputs of all encoder layers with layer aggregation and multi-layer attention mechanisms, and introduce an auxiliary regularization term to encourage different layers to capture diverse information. Dou et al. (2019) propose to use routing-by-agreement strategies to aggregate layers dynamically.

Knowledge Integration. Utilizing knowledge beyond the training data for machine translation may help produce accurate and meaningful translations. He et al. (2016a) incorporate SMT features, including a translation model and an n-gram language model, with the NMT model under the log-linear framework. Arthur et al. (2016) propose to improve the translation of low-frequency content words by augmenting NMT systems with discrete translation lexicons that efficiently encode translations of these low-frequency words. Stahlberg et al. (2016) investigate the use of hierarchical phrase-based SMT lattices in NMT. Li et al. (2017) show that source syntax can be explicitly incorporated into NMT effectively, which brings significant improvements. Wu et al. (2017) jointly construct and model the target word sequence and its corresponding dependency structure with sequence-to-dependency NMT. Eriguchi et al. (2017) learn to parse and translate by combining recurrent neural network grammar into NMT. Zhang et al. (2017c) explicitly incorporate the word reordering knowledge into attention-based NMT. Chen et al. (2017a)

improve NMT by explicitly incorporating source-side syntactic trees. Chen et al. (2017b) integrate source dependency representation into NMT. Wang et al. (2017c) propose to incorporate the SMT model into the NMT framework utilizing an auxiliary classifier to score the SMT recommendations and a gating function to combine the SMT recommendations with NMT generations. Zhang et al. (2017b) investigate how to integrate multiple overlapping, arbitrary prior knowledge sources, and propose to use posterior regularization to provide a general framework for integrating prior knowledge into NMT. Hokamp and Liu (2017) extend beam search to allow the inclusion of pre-specified lexical constraints. Feng et al. (2017) propose a memory-augmented NMT architecture to integrate the knowledge learned from conventional SMT into NMT. Dahlmann et al. (2017) introduce a hybrid search algorithm for attention-based NMT, which extends the beam search of NMT with phrase translations from SMT. Wang et al. (2017e) propose to translate phrases in NMT by integrating target phrases from an SMT system with a phrase memory. Yang et al. (2017) propose a hierarchical attentional neural translation model focusing on enhancing source-side hierarchical representations by covering both local and global semantic information using a bidirectional tree-based encoder. Chen et al. (2018a) improve NMT by incorporating multiple levels of granularity. Kiperwasser and Ballesteros (2018) propose a framework in which the model begins learning syntax and translation interleaved, gradually putting more focus on translation. Pu et al. (2018) demonstrate that Word Sense Disambiguation (WSD) can improve NMT by widening the source context considered when modeling the senses of potentially ambiguous words. Zhang et al. (2018d) propose a method for recalling previously seen translation examples and incorporating them into the NMT decoding process. Ugawa et al. (2018) incorporate named entity tags of source-language sentences. Cao and Xiong (2018) propose a novel method to combine the strengths of both translation memories and NMT for high-quality translation. Marcheggiani et al. (2018) incorporate semantic-role representations into NMT. Currey and Heafield (2018) incorporate source syntax into NMT using linearized parses. Wang et al. (2018a) employ a shared reconstructor and jointly learn to translate and predict dropped pronouns. Wang et al. (2019a) propose a unified and discourse-aware zero pronoun translation approach for NMT. Song et al. (2019) study the usefulness of Abstract Meaning Representation (AMR) on NMT. Guo et al. (2019c) propose a densely connected Graph Convolutional Network (GCN) for syntax-based neural machine translation and AMR-to-text generation. Yang et al. (2019e) introduce a new latent variable model to

capture the co-dependence between syntax and semantics. Hao et al. (2019a) present multi-granularity self-attention for the Transformer. Zhang et al. (2019b) propose to use the intermediate hidden representations of a well-trained end-to-end dependency parser for NMT. Zhu et al. (2020) fuse representations extracted from BERT with each layer of the encoder and decoder of the NMT model through attention mechanisms.

Deep NMT models. Increasing the depth of models allows to model complicated functions and increases their capacity but may also cause optimization difficulties. Zhou et al. (2016) introduce fast-forward linear connections for deep LSTM networks and an interleaved bi-directional architecture for stacking the LSTM layers. Wang et al. (2017b) propose a novel linear associative unit that uses linear associative connections between input and output of the recurrent unit and allows unimpeded information flow through both space and time to reduce the gradient propagation path inside the recurrent unit. Bapna et al. (2018) show that Transformer models with more than 12 encoder layers fail to converge, and propose the Transparent Attention (TA) mechanism which combines outputs of all encoder layers into weighted encoded representations. Wang et al. (2019c) find that deep Transformers with proper use of layer normalization are able to converge and propose to aggregate previous layers' outputs for each layer. Wu et al. (2019c) explore incrementally increasing the depth of the Transformer Big by freezing pre-trained shallow layers. Zhang et al. (2019a) attribute the convergence issue of deep Transformers to the fact that layer normalization over residual connections effectively reduces the impact of residual connections due to subsequent layer normalization, and propose a layer-wise initialization approach to reduce the standard deviation before normalization.

Efficiency. Improving the efficiency of NMT models reduces translation cost and latency. Kaiser et al. (2018) study how to apply depthwise separable convolutions into NMT which enables a significant reduction of the parameter count and amount of computation. Shen et al. (2018b) propose a densely connected NMT architecture to improve training efficiency. Zhang et al. (2018b) propose an addition subtraction twin-gated recurrent network which heavily simplifies the number of weight matrices among units of all existing gated RNNs. Zhang et al. (2018a) propose average attention as an alternative to the self-attention network in the Transformer decoder to accelerate decoding. Wang et al. (2018c) propose an efficient method to dynamically sample sentences in order to

accelerate NMT training. Zhang et al. (2018e) apply cube pruning into NMT to speed up translation. Zhang et al. (2018f) use a simple n-gram suffix based equivalence function and adapt it into beam search decoding. Wu et al. (2019b) introduce dynamic convolutions which are simpler and more efficient than self-attention. Guo et al. (2019b) replace the fully-connected attention structure with a star-shaped topology, in which every two non-adjacent nodes are connected through a shared relay node. Tay et al. (2019) propose the lightweight and memory efficient Quaternion Transformer. Bai et al. (2019) propose the deep equilibrium model that directly finds these equilibrium points via root-finding which has the notable advantage that training and prediction in these networks require only constant memory, regardless of the effective "depth" of the network, as analytical backpropagation can be performed through the equilibrium point using implicit differentiation. Gu et al. (2019) develop the Levenshtein Transformer, a partially autoregressive model devised for more flexible and amenable sequence generation. Dehghani et al. (2019) introduce a dynamic per-position halting mechanism to the Transformer. Elbayad et al. (2020) train Transformer models which can make output predictions at different stages of the network and investigate different ways to predict how much computation is required for a particular sequence. Kitaev et al. (2020) replace dot-product attention with one that uses locality-sensitive hashing, and use reversible residual layers instead of the standard residuals, which allows storing activations only once in the training process.

Robustness. Small perturbations in the input can severely distort intermediate representations and thus impact the translation quality of NMT models. Belinkov and Bisk (2018) confront NMT models with synthetic and natural sources of noise and find that state-of-the-art models fail to translate even moderately noisy texts that humans have no trouble comprehending. Heigold et al. (2018) investigate the robustness of NLP against perturbed word forms. Zhao et al. (2018) propose a framework to generate natural and legible adversarial examples that lie on the data manifold, by searching in semantic space of dense and continuous data representation, utilizing recent advances in Generative Adversarial Networks (GANs). Cheng et al. (2018) propose to improve the robustness of NMT models with adversarial stability training. Michel and Neubig (2018) propose a benchmark dataset for Machine Translation of Noisy Text (MTNT). Liu et al. (2019a) propose to improve the robustness of NMT to homophone noises by jointly embedding both textual and phonetic information of source sentences, and augmenting the training

dataset with homophone noises. Cheng et al. (2019) propose to improve the robustness of NMT models by attacking the translation model with adversarial source examples and defending the translation model with adversarial target inputs to improve its robustness against the adversarial source inputs. Zhang et al. (2019d) improve the decoding robustness by sampling context words not only from the ground truth sequence but also from the predicted sequence by the model during training. Sato et al. (2019) investigate approaches to apply adversarial perturbation to NMT. Michel et al. (2019) propose an evaluation framework for adversarial attacks on sequence-to-sequence models that takes the semantic equivalence of the pre-perturbation and post-perturbation input into account. Vaibhav et al. (2019) propose methods to enhance the robustness of MT systems by emulating naturally occurring noise in otherwise clean data.

Back-Translation. Back-translation has been proven effective in improving NMT performance by utilizing monolingual data. Sennrich et al. (2016b) propose to utilize target monolingual data by back-translating. Edunov et al. (2018) investigate a number of methods to generate synthetic source sentences, and find that in all but resource-poor settings, back-translations obtained via sampling or noised beam outputs are most effective. Fadaee and Monz (2018) explore different aspects of back-translation, showing that words with high prediction loss during training benefit most from the addition of synthetic data, and introduce several variations of sampling strategies targeting difficult-to-predict words using prediction losses and frequencies of words. Wang et al. (2019d) propose to quantify the confidence of NMT model predictions based on model uncertainty, and to better cope with noise in back-translation with word-level and sentence-level confidence measures based on uncertainty. Zheng et al. (2020) propose the mirror-generative NMT which simultaneously integrates the source to target translation model, the target to source translation model, and two language models.

Training of NMT Models. The training approach and objectives affect the performance of the trained model. Wiseman and Rush (2016) introduce a model and beam search training scheme to learn global sequence scores. Shen et al. (2016) propose minimum risk training for NMT to directly optimize evaluation metrics. Weng et al. (2017) propose to use word predictions as a mechanism for direct supervision. Li et al. (2018a)

introduce a disagreement regularization to explicitly encourage the diversity among multiple attention heads. Kuang et al. (2018a) study how to strengthen the connection between source and target words in translation by bridging source and target word embeddings. Yang et al. (2019d) propose a sentence-level agreement module to directly minimize the difference between the representation of source and target sentence. Yang et al. (2018b) introduce a discriminator to distinguish NMT outputs from golden target sentences. Wieting et al. (2019) introduce a semantic similarity based reward function for optimizing NMT systems. Yang et al. (2019f) propose a contrastive learning approach to reducing word omission errors in NMT. Garg et al. (2019) present an approach to training a Transformer model to produce both accurate translations and alignments. Cohn-Gordon and Goodman (2019) present a method to define a less ambiguous translation system in terms of an underlying pre-trained neural sequence-to-sequence model. Platanios et al. (2019) propose a framework to decide which training samples are shown to the model at different times during training, based on the estimated difficulty of a sample and the current competence of the model. Kumar et al. (2019) use reinforcement learning to learn a curriculum framework that allows examples to appear an arbitrary number of times and generalizes data weighting, filtering and fine-tuning schemes.

Non-Autoregressive Translation. NMT normally conditions each output word on previously generated outputs. Non-autoregressive translation avoids this autoregressive property and produces its outputs in parallel, reducing inference latency. Gu et al. (2018) introduce a model that avoids autoregressive decoding and produces translations in parallel, allowing an order of magnitude lower latency during inference. Lee et al. (2018) propose a conditional Non-Autoregressive Translation (NAT) model based on iterative refinement. Libovický and Helcl (2018) present an NAT architecture based on connectionist temporal classification. Ma et al. (2019) propose a simple, efficient and effective NAT model using latent variable models. Li et al. (2019c) propose to leverage the hints from hidden states and word alignments to help train NAT models. Wei et al. (2019) propose an imitation learning framework for NAT. Shao et al. (2019) propose approaches to retrieve target sequential information for NAT to enhance its translation ability while preserving the fast-decoding property. Guo et al. (2019a) propose to enhance the decoder inputs with a phrase table from SMT and transformed source word embeddings to improve NAT models. Wang et al. (2019g) propose to regularize the similarity between

consecutive hidden states based on the corresponding target tokens and to minimize a backward reconstruction error for NAT.

Empirical Studies. Empirical studies reveal how the specific model design affects performance. Luong et al. (2015) examine global attention (including 3 score functions: dot-product, general and concatenation) and local attention. Britz et al. (2017) provide practical insights into the relative importance of factors including embedding size, network depth, RNN cell type, residual connections, attention mechanism and decoding heuristics. Chen et al. (2018b) tease apart the new architectures of the Transformer and their accompanying techniques in two ways to identify several key modeling and training techniques, and apply them to the RNN architecture. They additionally analyze the properties of each fundamental seq2seq architecture and devise new hybrid architectures to combine their strengths. Pappas and Henderson (2019) investigate the usefulness of more powerful shared mappings for output labels, and propose a deep residual output mapping with dropout between layers to better capture the structure of the output space and avoid overfitting. So et al. (2019) apply Neural Architecture Search (NAS) to search for a better alternative to the Transformer.

Analysis of NMT Models. Analysis of NMT models helps understanding of the behavior and characteristics of NMT models and guides the design of new model architectures. Domhan (2018) show that recurrent and convolutional models can perform very close to the Transformer by borrowing concepts from the Transformer architecture, and that self-attention is much more important on the encoder side than on the decoder side. Tang et al. (2018) examine the capabilities of RNNs, CNNs, and self-attention networks in modeling long-range dependencies and semantic feature extraction on the subject-verb agreement task and the word sense disambiguation task respectively. Tran et al. (2018) find that LSTMs are notably more robust with respect to the presence of misleading features in the agreement task, and that LSTMs generalize better than the fully attentional Transformer to longer sequences in a logical inference task. They suggest that recurrence is a key model property that should not be sacrificed for efficiency when hierarchical structure matters for the task. Bisazza and Tump (2018) performs a fine-grained analysis of how various source-side morphological features are captured at different levels of the NMT encoder. While they are unable to find any correlation between the accuracy

of source morphology encoding and translation quality, they discover that morphological features are only captured in context and only to the extent that they are directly transferable to the target words. Li et al. (2019b) analyze word alignment quality in NMT with prediction difference, and further analyze the effect of alignment errors on translation errors, which demonstrates that NMT captures good word alignment for those words mostly contributed from source, while their word alignment is much worse for those words mostly contributed from target. Voita et al. (2019d) evaluate the contribution of individual attention heads to the overall performance of the model and analyze the roles played by them in the encoder. Yang et al. (2019b) propose a word reordering detection task to quantify how well word order information is learned by the Self-Attention Network (SAN) and RNN, and reveal that although recurrence structure makes the model more universally-effective on learning word order, learning objectives matter more in downstream tasks such as machine translation. Tsai et al. (2019) regard attention as applying a kernel smoother over the inputs with the kernel scores being the similarities between inputs, and analyze individual components of the Transformer's attention with the new formulation via the lens of the kernel. Tang et al. (2019) find that encoder hidden states outperform word embeddings significantly in word sense disambiguation. He et al. (2019) measure word importance by attributing the NMT output to every input word and reveal that words of certain syntactic categories have higher importance while the categories vary across language pairs. Voita et al. (2019a) use canonical correlation analysis and mutual information estimators to study how information flows across Transformer layers and find that representations differ significantly depending on the objectives (MT, LM and MLM).

Context-Aware NMT. NMT systems are generally trained on a large amount of sentence-level parallel data, and during prediction sentences are independently translated, ignoring cross-sentence contextual information. This leads to inconsistency between translated sentences. In order to address this issue, context-aware models have been proposed. Tiedemann and Scherrer (2017) discuss the effect of increasing the segments beyond single translation units, and observe cross-sentential attention patterns that improve textual coherence in translation. Läubli et al. (2018) show that human assessment has a stronger preference for human over machine translation when evaluating documents as compared to isolated sentences. Bawden et al. (2018) present hand-crafted, discourse test sets to

test the models' ability to exploit previous source and target sentences, and highlight the importance of target-side context. Voita et al. (2018) introduce a context-aware NMT model which controls the flow of information from the context to the translation model, and show that the model deals with pronoun translation and implicitly captures anaphora. Voita et al. (2019c) perform a human study on an English-Russian subtitle dataset and identify deixis, ellipsis and lexical cohesion as three main sources of inconsistency, and create test sets targeting these phenomena, and propose the CADec model which demonstrates major gains over a context-agnostic baseline on their benchmarks without sacrificing BLEU. Wang et al. (2017a) summarize the history in a hierarchical way, and propose a cross-sentence context-aware approach to integrate the history representation into NMT. Maruf and Haffari (2018) present a document-level NMT model which takes both source and target document context into account using memory networks. Tu et al. (2018) augment NMT models with a cache-like memory network which stores recent hidden representations as translation history. Kuang et al. (2018b) propose a cache-based approach to modeling coherence for NMT by capturing contextual information either from recently translated sentences or the entire document. Kuang and Xiong (2018) propose an inter-sentence gating model that uses the same encoder to encode adjacent sentences and controls the amount of information flowing from the preceding sentence to the translation of the current sentence. Maruf et al. (2019) propose a hierarchical attention approach for context-aware NMT which first uses sparse attention to selectively focus on relevant sentences in the document context and then attends to key words in those sentences. Zhang et al. (2018c) extend the Transformer model with a new context encoder to represent document-level context which is then incorporated into the original encoder and decoder. Miculicich et al. (2018) propose to integrate a hierarchical attention model into the original NMT architecture to capture context. Tan et al. (2019) propose a hierarchical model consisting of a sentence encoder to capture intra-sentence dependencies and a document encoder to model document-level information. Xiong et al. (2019) propose to train a model to learn a policy that produces discourse coherent text by a reward teacher. Voita et al. (2019b) perform automatic post-editing on a sequence of sentence-level translations with a DocRepair model trained on very large monolingual document-level data in the target language and their round-trip translations of each isolated sentence, and analyze which discourse phenomena are hard to capture using monolingual data only. Xu et al. (2020d) present an efficient architecture for context-aware NMT.

2.5 Evaluation Metrics

Human evaluations of many aspects of translation, including adequacy, fidelity and fluency of the translation, are time-consuming, expensive and hard to reproduce. Thus, automatic machine translation evaluation approaches correlating well with human evaluation and which are quick, inexpensive and language-independent are proposed. As automatic evaluation approaches are usually convenient to use, easy to reproduce and can be used to make comparisons with the work of others, they are popular in machine translation research, including the research presented in this thesis.

Papineni et al. (2002) propose BLEU, to date still the most frequently used MT evaluation metric, which ranks translation outputs using combinations of modified n-gram precisions with a sentence brevity penalty. Koehn (2004) presents bootstrap resampling methods to compute statistical significance of test results, and validates them on the concrete example of the BLEU score to measure differences between test results. Banerjee and Lavie (2005) propose METEOR to enhance the BLEU metric which is solely based on an explicit word-to-word matching between the MT output being evaluated and one or more reference translations. The METEOR matching approach supports not only matching between words that are identical in the two strings being compared, but can also match words that are simple morphological variants of each other (i.e., they have an identical stem), and words that are synonyms of each other. Snover et al. (2006) introduce Translation Edit Rate (TER) which measures the amount of editing that a human would have to perform to change a system output so it exactly matches a reference translation, and propose the Human-targeted Translation Edit Rate (HTER) which yields higher correlations with human judgments than BLEU, even when BLEU is given human-targeted references.

2.6 Conclusion

This chapter presents a literature review of research relevant to the studies presented in the thesis, including RBMT, SMT, NMT and evaluation metrics for MT, but mostly focuses on approaches relevant to NMT.

In the following chapters, we present our research and show: (i) how we learn source phrase representations and how we incorporate source phrase representation learning into

the state-of-the-art Transformer to improve its long-distance dependency capturing ability (in Chapter 3), (ii) our approach to stabilize the convergence of the Transformer model, especially for deep Transformers, by initializing their parameters under the Lipschitz constraint with both empirical results and theoretical analysis (in Chapter 4), (iii) the relation between the batch size and the gradient direction, how we dynamically find proper batch sizes for the training of the Transformer by monitoring gradient direction change in gradient accumulation, and how we achieve efficient monitoring of gradient direction change by sampling (in Chapter 5), and finally (iv) present our Neutron NMT implementation which supports our research (in Chapter 6).

Chapter 3

Learning Source Phrase Representations for Neural Machine Translation

The Transformer translation model (Vaswani et al., 2017) based on a multi-head attention mechanism can be computed effectively in parallel and has significantly pushed forward the performance of Neural Machine Translation (NMT).

Though intuitively the attentional network can connect distant words via shorter network paths than RNNs, empirical analysis demonstrates that its ability to capture long-range dependencies does not significantly outperform RNNs. Instead, self-attentional networks are good at word sense disambiguation and semantic feature extraction, while it is still a problem for the Transformer to fully model long-distance dependencies (Tang et al., 2018).

Considering that modeling phrases instead of words has significantly improved the Statistical Machine Translation (SMT) approach through the use of larger translation blocks ("phrases") and its reordering ability, modeling NMT at phrase level is an intuitive proposal to help the model better capture long-distance relationships. However, using phrases directly leads to large vocabulary size and data sparsity issues which are not acceptable for a deep learning approach.

This chapter addresses **RQ1:** *How to improve the ability of the Transformer in long-distance relation capturing?*, **RQ2:** *How to avoid the potentially large phrase table while benefiting from phrase representations?* and **RQ3:** *How to learn and utilize phrase representation in the Transformer translation model?*

Instead of using phrases directly in NMT, we first propose an attentive phrase representation generation mechanism that is able to generate phrase representations from corresponding token representations. In addition, we incorporate the generated phrase representations into the Transformer translation model to enhance its ability to capture long-distance relationships.

In our experiments, we obtain significant improvements on the WMT 14 English-German and English-French tasks on top of the strong Transformer baseline, which shows the effectiveness of our approach. Our approach helps Transformer Base models perform at the level of Transformer Big models on the En-De task, and even significantly better for long sentences, but with substantially fewer parameters and training steps. The fact that phrase representations help even in the Big setting further supports our conjecture that they make a valuable contribution to long-distance relations.

The core part of the research presented in this chapter has been previously published in Xu et al. (2020c).

3.1 Introduction

Throughout much previous NLP research with neural models on a wide variety of tasks, the long-distance dependency learning ability of neural networks has been widely examined, and a common finding is that neural models normally perform better on short sentences than on long sentences, which indicates that capturing long-distance relations is often challenging for neural approaches. Specifically, Tai et al. (2015) show that the Long Short-Term Memory Network (LSTM) handles short sentences better than long sentences. In Linzen et al. (2016)'s number prediction task which predicts the number of the verb following the subject (plural or singular) of the sentence, the accuracy of LSTM degrades consistently with increasing distances between the subject and the verb. Yang et al. (2017)

show that it is challenging for the LSTM-based NMT model to capture long-distance dependencies. For assessing the ability of NMT models in long-distance relation capturing, Tang et al. (2018) examine the performance of RNNs, CNNs and the Transformer on the subject-verb agreement task which is the most popular choice for evaluating the ability to capture long-range dependencies and has been used in many studies (Linzen et al., 2016; Sennrich, 2017). Tang et al. (2018) show that the Transformer models which connect distant tokens via shorter network paths than RNNs are not particularly stronger than RNN models for long distances, and the number of heads in multi-head attention is crucial for its performance over long distances. Yang et al. (2019b) show that the accuracies of encoders, including both the self-attentional encoder and the GRU (Cho et al., 2014) encoder, decrease on long-distance cases across language pairs and model variants on the word reordering detection task, which also suggest that both GRU and the self-attention network fail to fully capture long-distance dependencies.

In machine translation, the word sense disambiguation of a source word may not only rely on several other tokens nearby, but also on some other words far from it. Some tokens with long-distance relationships in the decoding history may also play an important role in the generation of a target token in addition to several previous tokens.

The Transformer (Vaswani et al., 2017), which has outperformed previous RNN/CNN based translation models (Bahdanau et al., 2014; Gehring et al., 2017), is based on multi-layer multi-head attention networks and can be trained in parallel very efficiently. Though attentional networks can connect distant words via shorter network paths than RNNs, empirical results show that its ability in capturing long-range dependencies does not significantly outperform RNNs. Instead, self-attentional networks have been found good at word sense disambiguation and semantic feature extraction, while it is still a problem for the Transformer to fully model long-distance dependencies (Tang et al., 2018).

Using phrases instead of words enables conventional SMT to condition on a wider range of context, and results in better performance in reordering and modeling long-distance dependencies. It is intuitive to let the NMT model additionally condition on phrase-level representations to capture long-distance dependencies better. However, there are two main issues that prevent NMT from directly using phrases:

- There are more phrases than tokens, and the phrase table is much larger than the word vocabulary, which is not affordable for NMT.
- Distribution over phrases is much sparser than that over words, which may lead to data sparsity and hurt the performance of NMT.

Instead of using phrases directly in NMT, in this chapter, we address the issues above with the following contributions:

- To address the large phrase table issue, we propose an attentive feature extraction model and generate phrase representation based on token representations "on the fly". Our model first summarizes the representation of a given token sequence with mean or max-over-time pooling, then computes the attention weight of each token based on the token representation and the summarized representation, and generates the phrase representation by a weighted combination of token representations.
- To help the Transformer translation model better model long-distance dependencies, we let both encoder layers and decoder layers of the Transformer attend the phrase representation sequence which is shorter than the token sequence, in addition to the original token representation. Since the phrase representations are produced and attended at each encoder layer, the encoding of each layer is also enhanced with phrase-level attention computation.
- To the best of our knowledge, our work is the first to model phrase representations and incorporating them into the Transformer.

Our approach empirically brings about significant and consistent improvements over the strong Transformer model (both Base and Big settings). We conduct experiments on the WMT 14 English-German and English-French news translation task, and obtain +1.29 and +1.37 BLEU improvements respectively on top of the strong Transformer Base baseline, which demonstrates the effectiveness of our approach. Our approach helps Transformer Base models perform at the level of Transformer Big models, and even significantly better for long sentences, but with substantially fewer parameters and training steps. It also shows effectiveness with the Transformer Big setting. We also conduct length analysis with our approach, and the results show how our approach improves long-distance dependency capturing, which supports our conjecture that phrase representation sequences

can help the model capture long-distance relations better, given that in translating long sentences, we shall encounter more long-distance dependencies than translating short sentences. In the linguistically-informed subject-verb agreement analysis on the *Lingeval97* dataset (Sennrich, 2017) following Tang et al. (2018), our approach improves the accuracy of long-distance subject-verb dependencies, especially for cases where there are more than 10 tokens between the verb and the related subject.

3.2 Background and Related Work

In this section, we first review previous work which utilizes phrases in recurrent sequence-to-sequence models, then give a brief introduction to the stronger Transformer translation model that our work is based on.

3.2.1 Utilizing Phrases in RNN-based NMT

Most previous work focuses on utilizing phrases from SMT in NMT to address its coverage (Tu et al., 2016) problem.

Dahlmann et al. (2017) suggest that SMT usually performs better in translating rare words and using phrasal translations, though NMT reaches better translation quality. To benefit from SMT features, including phrase-level translation probabilities and a target language model, they introduce a hybrid search algorithm for attention-based NMT which extends the beam search of NMT with phrase translations from SMT. When to use phrase translations is decided based on attention weights of the NMT decoder which provides a soft coverage of the source sentence words, and a log-linear model is applied to combine the NMT translation score with phrase-based scores and n-gram target language model scores.

Wang et al. (2017d) propose that while NMT generally produces fluent but often inadequate translations, SMT yields adequate translations though less fluent. They incorporate SMT into NMT by utilizing recommendations from SMT in each decoding step of NMT to address the coverage issue and the unknown word issue of NMT. They employ an auxiliary classifier and a gating mechanism to score and combine SMT outputs with NMT decoding. With the proposed architecture, they integrate SMT into the training of NMT

in an end-to-end manner. As for the unknown word issue, they select a proper SMT candidate to replace an unknown target word conditioned on both attention weights of the NMT model and the coverage information from the SMT model.

Wang et al. (2017e) suggest that phrases play a vital role in machine translation, and propose to translate phrases in NMT by integrating target phrases from an SMT system with a phrase memory given that it is hard to integrate phrases into NMT which reads and generates sentences in a token-by-token way. The phrase memory is provided by the SMT model which dynamically picks relevant phrases with the partial translation from the NMT decoder in each decoding step. Alignment information derived from the attention mechanism of the NMT is introduced to the ranking of phrases of the SMT model, which leads to better interaction between SMT and NMT. The NMT decoder then decides to generate a word or to select an appropriate phrase from the phrase memory with a neural balancer. If a phrase was selected, the NMT decoder would perform phrase translation and update its decoding state by force decoding all words in that phrase.

3.2.2 The Transformer Translation Model

Our research is based on the Transformer translation model (Vaswani et al., 2017) shown in Figure 3.1, which significantly outperforms the previous recurrent sequence-to-sequence approach and can be efficiently computed in parallel.

The Transformer includes an encoder and a decoder. Both encoder and decoder are a stack of 6 layers. Besides the embedding matrix and positional embedding matrix in both encoder and decoder, the decoder also has a softmax classifier layer to produce translated tokens. The weights of the softmax classifier are normally tied to the target embedding matrix.

Each encoder layer consists of a self-attention network attending the whole input sequence to build contextual representations, and a feed-forward neural network to process the collected information. A decoder layer has an additional cross-attention layer between the self-attention sub-layer and the feed-forward neural network sub-layer to attend to the encoder's outputs to provide information from the encoded representation of the given source sentence. To stabilize training, a residual connection (He et al., 2016) is employed around each sub-layer, followed by layer normalization (Ba et al., 2016).

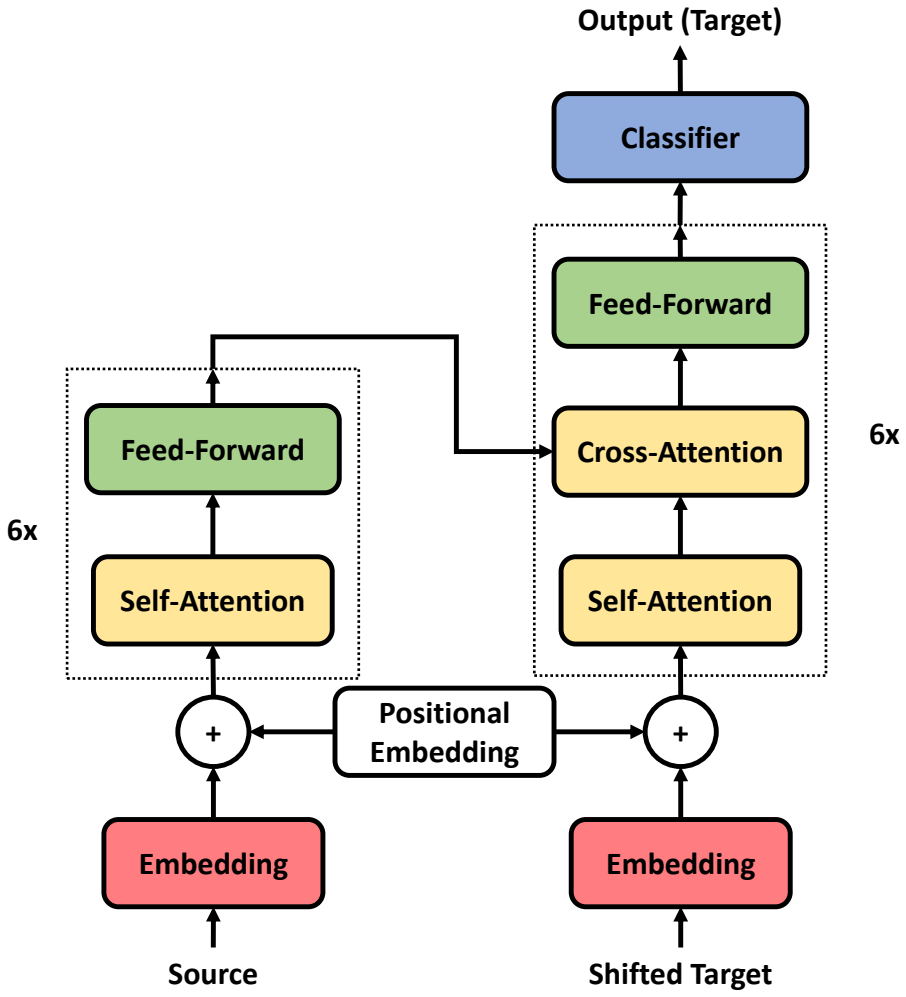


FIGURE 3.1: The Transformer translation model. Residual connection and layer normalization are omitted for simplicity.

Both encoder layers and decoder layers make use of the multi-head attention mechanism. The multi-head attention mechanism calculates the attention results of given queries on corresponding keys and values. It first projects queries, keys and values with 3 independent linear transformations, then splits the transformed key, query and value embeddings into several chunks of d_k dimension vectors, each chunk is called a head, [1] and scaled dot-product attention is independently applied in each head:

$$Attn(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V \tag{3.1}$$

where Q, K and V stand for the query vectors, key vectors and value vectors. Finally, the

[1] d_k is 64 for both the Transformer Base and the Transformer Big, and the numbers of heads for them are 8 and 16 respectively.

network concatenates the outputs of all heads and transforms the concatenation into the target space with another linear layer. The self-attention network uses the query sequence also as the key sequence and the value sequence in computation, while the cross-attention feeds another vector sequence to attend as queries and values.

Comparing the computation of the attentional network with RNNs, it is obvious that the attention computation connects distant words with a shorter network path, and intuitively it should perform better in capturing long-distance dependencies. However, empirical results show that its ability in modeling long-range dependencies does not significantly outperform RNNs. Instead, it leads to good performance on word sense disambiguation and semantic feature extraction.

3.2.3 Comparison with Previous Works

Compared to previous works using RNN-based NMT (He et al., 2016b; Wang et al., 2017d,e; Dahlmann et al., 2017), our proposed approach is based on the Transformer model, with the following further important differences:

- Our approach aims to improve the long-distance dependency modeling ability of NMT instead of coverage (Tu et al., 2016).
- Our approach does not require to train an SMT system or to extract aligned phrase translation from the training corpus, which makes it efficient and avoids suffering from potential error propagation from the SMT system. Instead, we directly generate and utilize our phrase representations during encoding rather than using recommended phrase translation pairs from SMT to aid decoding. There is no requirement to interact with the part of the algorithm which generates phrases. Our proposed neural phrase representation learning model is deeply integrated into the translation model, and the whole neural model can be trained in an end-to-end manner simply through backpropagation.
- We iteratively and dynamically generate phrase representations with token vectors. Previous work does not use SMT phrases in this way. Benefiting from the powerful multi-head attention mechanism (Vaswani et al., 2017) and the proposed attentive phrase representation generation algorithm, dense phrase embeddings should be

more powerful and informative than discrete phrase sequences recommended by SMT in previous approaches.

In more recent work, Wang et al. (2019f) propose to augment SANs with structural position representations to model the latent structure of the input sentence complementing standard sequential positional representations. They use dependency trees to represent the grammatical structure of a sentence, and propose two strategies to encode the positional relationships among words in the dependency tree. Hao et al. (2019a) also suggest to improve NMT performance from explicit modeling of phrases, as prior work on SMT has shown that extending the basic translation unit from words to phrases has produced substantial improvements. But they propose the multi-granularity self-attention mechanism which performs phrase-level (either n-grams or syntactic phrases) attention with several attention heads, rather than learning source phrase representations as in our work.

3.3 Transformer with Phrase Representation

In this section, we introduce our approach to generating phrase representations and integrating phrase representations into the Transformer model. In contrast to phrases in SMT which are bilingual pairs, in our approach they are segments on the monolingual source side.

Unlike SMT which learns phrase translations from alignment information, NMT translates in a token-by-token manner, and does not have a procedure to do phrase translation, since it cannot use phrases directly due to the large sparse phrase table. Fortunately, neural models have been proven powerful in combining representations even without explicitly modeling linguistic structures (Cho et al., 2014; Hochreiter and Schmidhuber, 1997; Vaswani et al., 2017; Devlin et al., 2019). To avoid a large phrase table while at the same time availing of its advantages, we generate phrase representations out of the token sequence vectors with an attentional network. Then we let the Transformer model attend the shorter phrase representation sequence before attending the original longer token representation sequence, with the expectation that it can perform token-level attention better with the previous information gained from phrase-level attention, especially when capturing long-distance dependencies.

For the segmentation of phrases, given that n-gram phrases with a fixed n are very effective for tensor libraries, we first try to cut a token sequence into a phrase sequence with a fixed phrase length which varies with the sequence length. Specifically, for long sentences, each phrase contains at most 8 tokens. We do not use a larger value because this setting can already significantly reduce the length of sequences, i.e., it turns the longest sequence of 256 tokens in the training set into a sequence of 32 phrases, and a reasonably small maximum phrase length value helps reduce inevitable information loss in merging several token vectors into one. Since short sentences suffer fewer problems in modeling long-distance dependencies and merging more token vectors into one fixed dimension vector will result in a loss of information, we try to cut a short sentence into at least 6 phrases while ensuring that there are at least 3 tokens inside a phrase. In practice, we implement this as:

$$ntok = max(min(8, seql/6), 3) \tag{3.2}$$

where $ntok$ and $seql$ stand for the number of tokens in each phrase and the length of a sentence respectively.

We pad the last phrase in case it does not have sufficient tokens. Thus we can transform the whole sequence into a tensor.

The n-gram phrase segmentation is efficient and simple, and we suggest the drawbacks of such "casual" segmentation boundaries can be alleviated with self-attention computation across the whole sequence and the attention mechanism applied in the generation of phrase representations which values tokens differently to a large extent, given that neural models have been proven good at learning competitively effective representations with the gate or attention mechanism even without modeling linguistic structures (Cho et al., 2014; Hochreiter and Schmidhuber, 1997; Vaswani et al., 2017; Devlin et al., 2019).

In our experiments, we also explore phrases extracted from the Stanford Parser (Socher et al., 2013) as an alternative to our simple segmentation strategy. The maximum number of tokens allowed is consistent with the simple segmentation approach, and we try to use the tokens from the largest sub-tree that complies with the maximum token limitation or from several adjacent sub-trees of the same depth as a phrase for efficiency. Our algorithm to extract phrases from parse trees is shown in Algorithm 1.

Algorithm 1 Extracting phrases from a parse tree. Input: A parse tree T, maximum tokens allowed in a phrase n; Output: Extracted phrase sequence S.

while T is not empty **do**
  Initialize a phrase sequence $p = []$, maximum tokens allowed in this phrase $mt = n$;
  Find the largest sub-tree ST with nst tokens ($nst < n$) and depth dst from the right side of T;
  Add the token sequence in ST into p;
  Remove ST from T;
  while $mt > 0$ **do**
    Find the adjacent sub-tree STA of depth dst with $nsta$ tokens from the right side of T;
    if STA exists and $nsta \leq mt$ **then**
      Insert the token sequence of STA to the beginning of p;
      Remove STA from T;
      $mt = mt - nsta$;
    else
      Break;
    end if
  end while
  Append p to S;
end while
Reverse S;
return S

To efficiently parallelize parser-based phrases of various lengths in a batch of data, we pad short phrases to the same length as the longest phrases in the batch of sentences. Thus a batch of sequences of phrases can be saved into a tensor. However, significantly more "<pad>" tokens will be introduced, and the syntax-informed model is slightly slower than the simple approach. An example of the two phrase segmentation approaches is shown in Figure 3.2.

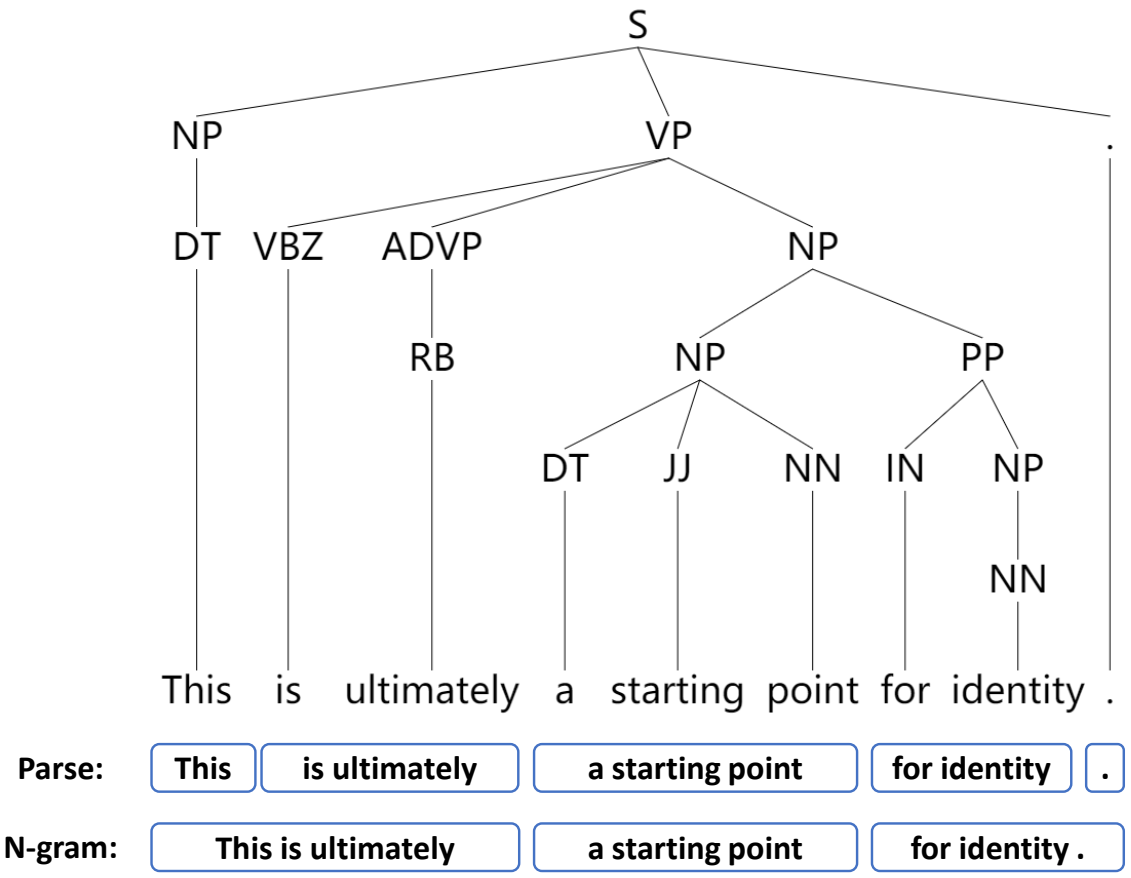


FIGURE 3.2: Example of parse phrases and n-gram phrases.

3.3.1 Attentive Phrase Representation Generation

Merging several token vectors into one is very likely to incur information loss, and introducing an importance evaluation mechanism is better than treating tokens equally. To highlight the most important features in a segmented phrase chunk, we introduce an attentive phrase representation generation model to value tokens differently according to their importance in the phrase. The model first roughly extracts features from all tokens in a phrase into a vector, then assigns a score to each token by comparing each token vector with the extracted phrase feature vector, and produces the weighted accumulation of all token vectors according to their scores as the final representation of the phrase.

Phrase representations are generated in every encoder layer. For the k_{th} encoder layer, we generate phrase representation $R^k_{e_{phrase}}$ from its input representation. Assume the phrase contains m tokens $\{t_1, ..., t_m\}$, and $\{R^k_{e_{t_1}}, R^k_{e_{t_2}}, ..., R^k_{e_{t_m}}\}$ are the corresponding input vectors to the encoder layer, we first generate a summary representation by:

$$R^k_{e_{all}} = F_{glance}(R^k_{e_{t_1}}, ..., R^k_{e_{t_m}}) \tag{3.3}$$

where F_{glance} is a function to extract features of the vector sequence into a fixed-dimension vector. We explore both element-wise mean operation and max-over-time pooling operation in our work.

After the summarized representation is produced, we calculate a score for each token in the phrase. The score of the i_{th} token s^k_i is calculated as:

$$\mathrm{s}^k_i = W^k_2 \sigma(W^k_1 [R^k_{e_{t_i}} | R^k_{e_{all}}] + b^k_1) + b^k_2 \tag{3.4}$$

where σ is the sigmoid activation function, and "|" means concatenation of vectors. The rationale for designing this approach is further explained below.

Then we normalize the score vector to weights with the softmax function, and the probability of the i_{th} token p^k_i is:

$$\mathrm{p}^k_i = \frac{e^{\mathrm{s}^k_i}}{\sum_{i=1}^{m} e^{\mathrm{s}^k_i}} \tag{3.5}$$

Finally, the representation of the phrase in the k_{th} encoder layer $R^k_{e_{phrase}}$ is generated by a weighted combination of all token vectors:

$$R^k_{e_{phrase}} = \sum_{i=1}^{m} p^k_i R^k_{e_{t_i}} \tag{3.6}$$

The representation of the phrase sequence can be computed efficiently in parallel. Each encoder layer will produce a vector sequence as the phrase representation. We do not use the multi-head attention in the computation of the phrase-representation attention because of two reasons:

- The multi-head attention calculates weights through dot-product. We suggest that a 2-layer neural network might be more powerful at semantic level feature extraction,

and it is less likely to be affected by positional embeddings which are likely to vote up adjacent vectors.

- Though we employ a 2-layer neural network, it only has one linear transformation and a vector to calculate attention weights, which contains fewer parameters than the multi-head attention model that has 4 linear transformations.

Recent studies show that different encoder layers capture linguistic properties of different levels (Peters et al., 2018), and aggregating layers is of profound value to better fuse semantic information (Shen et al., 2018b; Dou et al., 2018; Wang et al., 2018b; Dou et al., 2019). We assume that different decoder layers may value different levels of information, i.e., the representations of different encoder layers, differently. Thus we weighted combined phrase representations from every encoder layer for each decoder layer with the Transparent Attention (TA) mechanism (Bapna et al., 2018). For the decoder layer j, the phrase representation $R^j_{d_{phrase}}$ fed into that layer is calculated by:

$$R^j_{d_{phrase}} = \sum_{i=0}^{d} w_i^j R^i_{e_{phrase}} \tag{3.7}$$

where w_i^j are softmax normalized parameters trained jointly with the full model to learn the importance of encoder layers for the j_{th} decoder layer. d is the number of encoder layers, and 0 corresponds to the embedding layer.

3.3.2 Incorporating Phrase Representation into NMT

After the phrase representation sequence for each encoder layer and decoder layer is calculated with the approach described above, we propose an attentive combination network to incorporate the phrase representation for each layer into the Transformer translation model to aid it modeling long-distance dependencies. The attentive combination network is inserted in each encoder layer and each decoder layer to bring in information from the phrase representation. The architectures of the encoder layer and the decoder layer of the Transformer model with phrase representation are shown in Figure 3.3.

For an encoder layer, the new computation order is cross-attention from source tokens to source phrases → self-attention over tokens → feed-forward neural network to process

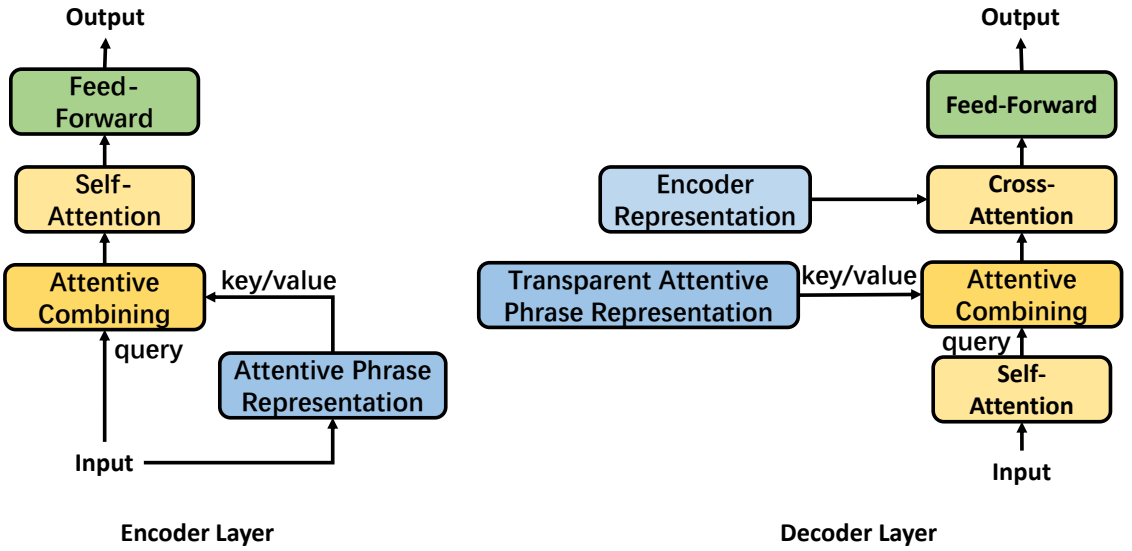


FIGURE 3.3: The encoder/decoder layer of the Transformer model with phrase representation. Residual connection and layer normalization are omitted for simplicity.

collected features, while for a decoder layer, it is: self-attention over decoded tokens → cross-attention to source phrases → cross-attention to source tokens → feed-forward neural network to process collected features. Compared to the computation order of the standard Transformer, the new computation order performs additional attending at phrase level before attending source token representations at the token level. We conjecture that attending at phrase level should be easier than at token level, and attention results at phrase level may aid the attention computation at the token level.

For a given input token representation sequence x and a phrase vector sequence R_{phrase}, the attentive combination network inserted into encoder and decoder layers first attends the phrase representation sequence and computes the attention output out_{phrase} as follows:

$$out_{phrase} = \text{Attn}_{\text{MH}}(x, R_{phrase}) \tag{3.8}$$

where Attn_{MH} is a multi-head cross-attention network with x as keys and R_{phrase} as corresponding queries and values.

The attention result is then combined again with the original input sequence x with a 2-layer neural network which aims to make up for potential information loss in the phrase representation with the original token representation:

$$out = W_4\sigma(W_3[x|out_{phrase}] + b_3) + b_4 \tag{3.9}$$

We also employ a residual connection around the attentive combination layer, followed by layer normalization to stabilize training.

Since the phrase representation is produced inside the Transformer model and utilized as the input of layers, and all related computations are differentiable, the attentive phrase representation model is simply trained as part of the whole model through backpropagation.

3.4 Experiments

In this section, we report our experiment settings and results, along with discussions and analysis. To compare with Vaswani et al. (2017), we conducted our experiments on the WMT 14 English to German and English to French news translation tasks.

3.4.1 Settings

We implemented our approaches based on our Neutron implementation (Xu and Liu, 2019) of the Transformer translation model described in Chapter 6. We applied joint Byte-Pair Encoding (BPE) (Sennrich et al., 2016a) with $32k$ merge operations on both data sets to address the unknown word problem. We only kept sentences with a maximum of 256 subword tokens for training. Training sets were randomly shuffled in every training epoch. The concatenation of newstest 2012 and newstest 2013 was used for validation and newstest 2014 as test sets for both tasks.

The number of warmup steps was set to $8k$, [2] and each training batch contained at least $25k$ target tokens. Our experiments ran on 2 GTX 1080 Ti GPUs, and a large batch size was achieved through gradient accumulation. We used a dropout of 0.1 for all experiments except for the Transformer Big on the En-De task which was 0.3. The training steps for Transformer Base and Transformer Big were $100k$ and $300k$ respectively following Vaswani et al. (2017). We employed a label smoothing value of 0.1 (Szegedy

[2] https://github.com/tensorflow/tensor2tensor/blob/v1.15.6/tensor2tensor/models/transformer.py#L1850

Models	En-De	En-Fr
Transformer Base	27.38	39.34
+PR	**28.67**†	**40.71**†
Transformer Big	28.49	41.36
+PR	**29.60**†	**42.45**†

TABLE 3.1: Results on WMT 14 En-De and En-Fr.

et al., 2016). We used the Adam (Kingma and Ba, 2015) optimizer with 0.9, 0.98 and 10^{-9} as β_1, β_2 and ϵ. The other settings were the same as Vaswani et al. (2017) except that we did not bind the embedding between the encoder and the decoder for efficiency, because if the source and target embeddings were bound, the number of classes of the classifier and the corresponding computation would increase due to the fact that there are some tokens which only appear in the source side.

We used a beam size of 4 for decoding, and evaluated tokenized case-sensitive BLEU [3] with the averaged model of the last 5 checkpoints for Transformer Base and 20 checkpoints for Transformer Big saved with an interval of $1,500$ training steps (Vaswani et al., 2017). We also conducted significance tests (Koehn, 2004).

3.4.2 Main Results

We applied our approach to both the Transformer Base setting and the Transformer Big setting, and conducted experiments on both tasks to validate the effectiveness of our approach. Since parsing a large training set (specifically, the WMT 14 En-Fr dataset) is slow, we did not use phrases from linguistic parse results in this experiment (reported in Table 3.1). [4] Results are shown in Table 3.1. † indicates $p < 0.01$ compared to the baseline for the significance test.

Table 3.1 shows that modeling phrase representation can bring consistent and significant improvements on both tasks, and benefit both the Transformer Base model and the stronger Transformer Big model. "+PR" is the Transformer with Phrase Representation, corresponding to the "+Max+Attn+TA" setting in Table 3.2.

[3] `https://github.com/moses-smt/mosesdecoder/blob/master/scripts/generic/multi-bleu.perl`

[4] We show results for Transformer Base with linguistic parse driven phrases in the ablation study in Table 3.2.

Models	BLEU	Δ	Para. (M)	Time	
				Train	Decode
Transformer Base	27.38		88.1	1.00x	1.00x
+Mean	27.99	0.61	129.0	1.64x	1.45x
+Max	28.13	0.75		1.60x	1.40x
+Max+Attn	28.52	1.14	173.0	1.74x	1.52x
+Max+Attn+TA	28.67	1.29		1.75x	1.53x
+Max+Attn+TA+Parsing Phrase	**28.76**	**1.38**		1.83x	1.60x
Transformer Big	28.49	1.11	264.1	7.73x	2.68x

TABLE 3.2: Ablation study on the WMT 14 En-De task. Δ indicates the BLEU improvements compared to the Transformer Base. Time represents the time consumption compared to the Transformer Base (in training and decoding). The Transformer Big consumes 3 times the training steps of the Transformer Base.

The En-Fr task used a larger dataset ($\sim 36M$ sentence pairs) and achieved a higher baseline BLEU than the En-De task. We suggest that the significant improvements obtained by our approach on the En-Fr task with the Transformer Big supports the effectiveness of our approach in challenging settings in the sense that our approach also produces improvements in large data settings.

We did not compare our methods with previous RNN-based approaches (Wang et al., 2017d,e) which use phrases recommended by an SMT system in the decoding to address the coverage issue because of the following reasons:

- Our approach is to learn phrase representations and to incorporate them into Transformers, which set higher baselines than RNN-based NMT, to enhance its long-distance dependency modeling ability. This is quite different from previous work.
- Introducing SMT phrases to the Transformer requires its decoder to do step-by-step decoding, which prevents the training of the Transformer decoder from availing of efficient parallelization.
- The unknown target word problem which was part of the focus and motivation in the previous work has already been addressed by BPE.

3.4.3 Ablation Study

We also conducted a Transformer Base based ablation study on the WMT 14 En-De task to assess the influence of phrase representation, attention mechanism in phrase representation generation, transparent attention and phrases from parser output on performance. Results are shown in Table 3.2.

"+Mean" and "+Max" are only using element-wise mean operation and max-over-time pooling to generate an initial rough phrase representation of a given token sequence. "+Attn" indicates generating phrase representations with our attentive approach, on top of the max-over-time pooling as F_{glance} in Equation 3.3. "+TA" indicates the use of the Transparent Attention mechanism to fuse information generated from every encoder layer for different decoder layers, [5] otherwise only outputs of the last encoder layer are fed into all decoder layers. "+Parse" means using phrases extracted from parse results with Algorithm 1.

Table 3.2 shows that introducing phrase representation can significantly improve the strong Transformer Base baseline, even just a simple element-wise mean operation over token representations brings about a +0.61 BLEU improvement ($p < 0.01$). Summarizing representations with max-over-time pooling performs slightly better than with the element-wise mean operation. Our attentive phrase representation generation approach can bring further improvements over the max-over-time pooling approach. Though utilizing phrases from the parser can make use of linguistic knowledge and obtain most improvements, our simple and effective segmenting approach performs competitively, and we interpret these comparisons to show the positive effects of collapsing token sequences into shorter phrase sequences on the modeling of long-distance dependencies.

Though a significant amount of parameters are introduced for incorporating phrase representation into the Transformer model, our approach ("+Max+Attn+TA") improved the performance of the Transformer Base model by +1.29 BLEU on the WMT 14 En-De news translation task, and our proposed Transformer model with phrase representation still performs competitively compared to the Transformer Big model with only about half

[5] This only introduces an additional $7 * 6$ parameter matrix, which does not show significant influence in view of the number of parameters.

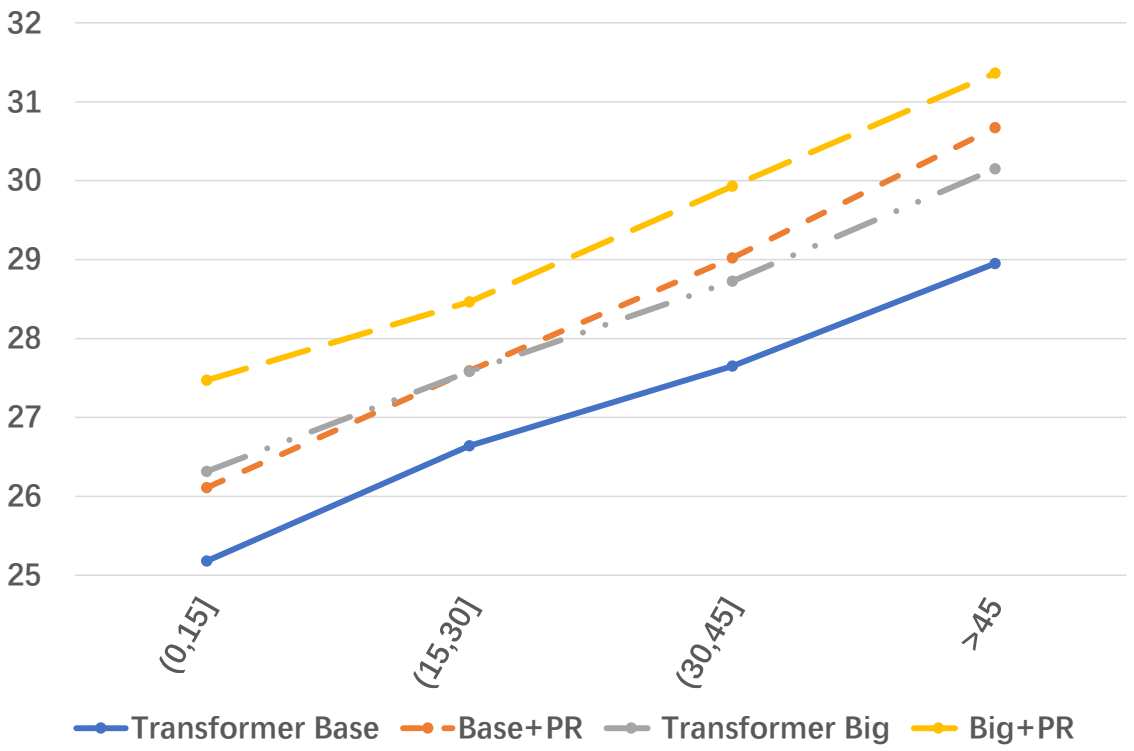


FIGURE 3.4: BLEU scores with respect to various input sentence lengths.

the number of parameters and 1/3 of the training steps. Thus, we suggest our improvements are not only because of introducing parameters, but also due to the modeling and utilization of phrase representation.

3.4.4 Length Analysis

To analyze the effects of our phrase representation approach on performance with increasing input length, we conducted a length analysis on the news test set of the WMT 14 En-De task. Following Bahdanau et al. (2014) and Tu et al. (2016), we grouped sentences of similar lengths together and computed BLEU scores of standard Transformers and Transformers with phrase representations for each group. Results are shown in Figure 3.4.

Figure 3.4 shows that our approach incorporating phrase representation into the Transformer significantly improves its performance in all length groups, and longer sentences show significantly more improvements than shorter sentences. In the Transformer Base setting, our approach improved the group with sentences of more than 45 tokens by +1.72

BLEU, almost twice of the improvements for sentences with less than 15 tokens which was +0.93 BLEU.

The effects of incorporating phrase representations into the Transformer are more significant, especially when compared to the Transformer Big which has about twice the number of parameters as our approach and consumes 3 times the training steps. According to Tang et al. (2018), the number of attention heads in Transformers impacts their ability to capture long-distance dependencies, and specifically, many-headed multi-head attention is essential for modeling long-distance phenomena with only self-attention. The Transformer Big model, with twice the number of heads in the multi-head attention network compared to those in the Transformer Base model, should be better at capturing long-distance dependencies. However, comparing with the Transformer Base, the improvement of the Transformer Big on long sentences (+1.20 BLEU for sentences with more than 45 tokens) was similar to that on short sentences (+1.14 BLEU for sentences with no more than 15 tokens), while our approach to model phrases in the Transformer model even brings significantly ($p < 0.01$) more improvements (+1.72 BLEU) on the performance of long sentences with the Transformer Base setting (8 heads) than the Transformer Big with 16 heads (+1.20 BLEU).

The length analysis result is consistent with our conjecture, given that there are likely to be more long-distance dependencies in longer source sentences. We suggest that phrase sequences which are shorter than the corresponding token sequences can help the model capture long-distance dependencies better, and modeling phrase representations for the Transformer can enhance its performance on long sequences.

3.4.5 Subject-Verb Agreement Analysis

Intuitively, in translating longer sentences, we expect to encounter more long-distance dependencies than in short sentences. To verify whether our method can improve the capability of the NMT model to capture long-distance dependencies, we also conducted a linguistically-informed subject-verb agreement analysis on the *Lingeval97* dataset (Sennrich, 2017) following Tang et al. (2018).

In German, subjects and verbs must agree with one another in grammatical number and person. In *Lingeval97*, each contrastive translation pair consists of a correct reference

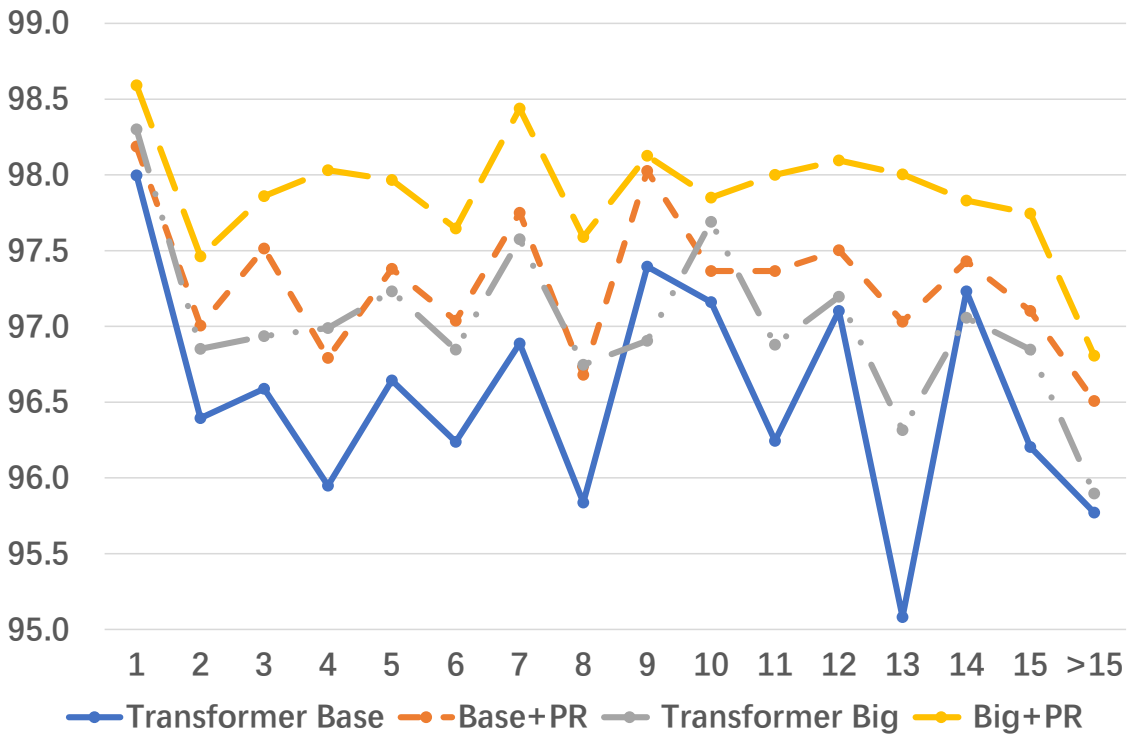


FIGURE 3.5: Subject-verb agreement analysis. X-axis and y-axis represent subject-verb distance in words and the accuracy respectively.

translation, and a contrastive example that has been minimally modified to introduce one translation error. Specifically, the grammatical number of a verb is modified to introduce an agreement error. The accuracy of a model is the number of times it assigns a higher score to the reference translation than to the contrastive one, relative to the total number of predictions. Results are shown in Figure 3.5.

Figure 3.5 shows that both increasing the number of heads (i.e., the Transformer Big) and incorporating phrase representations can improve the accuracy of subject-verb dependencies in almost all distances, especially for long distances. However, the accuracy improvements brought by our approach to integrating phrase representations into the Transformer Base are larger than that by the Transformer Big, especially for cases where there are more than 10 tokens between the verb and the corresponding subject.

3.5 Conclusion

Considering that the strong Transformer translation model still has difficulty in fully capturing long-distance dependencies (Tang et al., 2018), and that using a shorter phrase

sequence (in addition to the original longer token sequence) is an intuitive approach to help the model capture long-distance features, in this chapter, we first propose an attention mechanism to generate phrase representations by merging corresponding token representations. In addition, we incorporate the generated phrase representations into the Transformer translation model to help it better capture long-distance relationships.

We obtain statistically significant improvements on the WMT 14 English-German and English-French tasks over the strong Transformer baseline, which demonstrates the effectiveness of our approach. Our further analyses (on both source input length and subject-verb agreement) show that integrating phrase representation learning into the Transformer empirically improves its performance, especially in handling long-distance dependencies.

Chapter 4

Lipschitz Constrained Parameter Initialization for Deep Transformers

The Transformer translation model employs residual connection and layer normalization to ease the optimization difficulties caused by its multi-layer encoder/decoder structure. Previous research shows that even with residual connection and layer normalization, Transformers with deep encoders still have difficulty in training, and particularly Transformer models with more than 12 encoder layers fail to converge according to Bapna et al. (2018).

In this chapter, we address **RQ4:** *Why do Transformers, specifically deep Transformers, have difficulty in converging even with layer normalization and residual connections?* and **RQ5:** *How to prevent layer normalization from shrinking residual connections?*

We first empirically demonstrate that a simple modification made in the official implementation, which changes the computation order of residual connection and layer normalization, can significantly ease the optimization of deep Transformers.

We then compare the subtle differences in computation order in considerable detail, and present a parameter initialization method that leverages the Lipschitz constraint on the initialization of Transformer parameters that effectively ensures training convergence.

In contrast to findings in previous research, we further demonstrate that with Lipschitz parameter initialization, deep Transformers with the *original* computation order can converge, and obtain significant BLEU improvements with up to 24 layers. In contrast to previous research which focuses on deep encoders, our approach additionally enables Transformers to also benefit from deep decoders. Large parts of the research presented in this chapter are based on Xu et al. (2020a).

4.1 Introduction

Neural machine translation has achieved great success in the last few years (Bahdanau et al., 2014; Gehring et al., 2017; Vaswani et al., 2017). The Transformer (Vaswani et al., 2017), which has outperformed previous RNN/CNN based translation models (Bahdanau et al., 2014; Gehring et al., 2017), is based on multi-layer self-attention networks and can be trained very efficiently.

The multi-layer structure allows the Transformer to model complicated functions. Increasing the depth of models can increase their capacity but may also cause optimization difficulties (Mhaskar et al., 2017; Telgarsky, 2016; Eldan and Shamir, 2016; He et al., 2016; Bapna et al., 2018). In order to ease optimization, the Transformer employs residual connection and layer normalization techniques which have been proven useful in reducing optimization difficulties of deep neural networks for various tasks (He et al., 2016; Ba et al., 2016).

However, even with residual connections and layer normalization, deep Transformers are still hard to train: the original Transformer (Vaswani et al., 2017) only contains 6 encoder layers and 6 decoder layers. Bapna et al. (2018) show that Transformer models with more than 12 encoder layers fail to converge, and propose the Transparent Attention (TA) mechanism which combines outputs of all encoder layers into a weighted encoded representation. Wang et al. (2019c) find that deep Transformers with proper use of layer normalization are able to converge and propose to aggregate previous layers' outputs for each layer. Wu et al. (2019c) explore incrementally increasing the depth of the Transformer Big by freezing pre-trained shallow layers. Concurrent work closest to ours is Zhang et al. (2019a). They address the same issue, but propose a different layer-wise initialization approach to reduce the standard deviation.

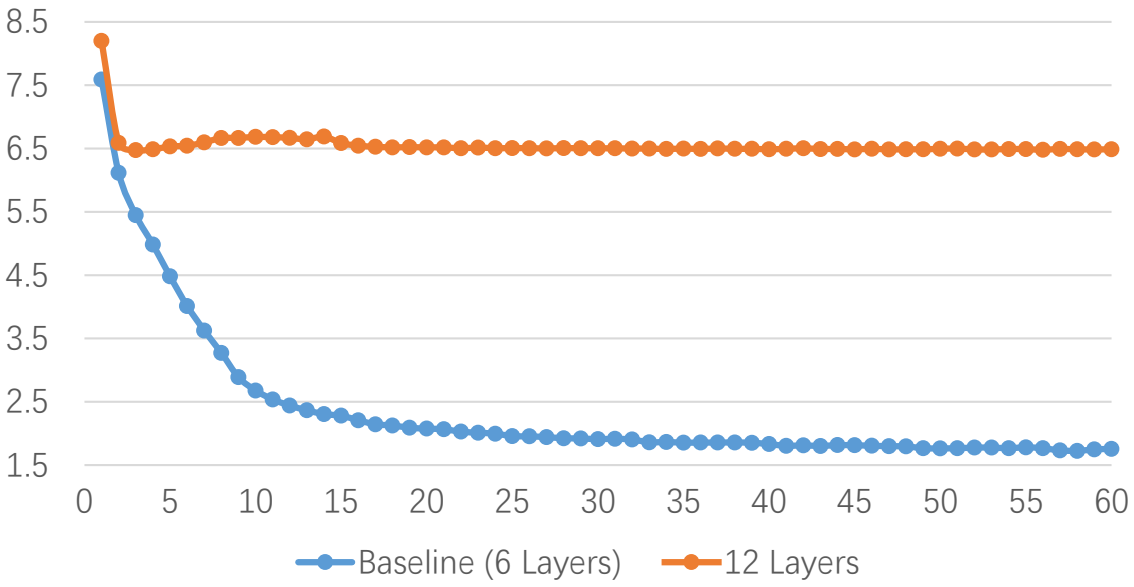


FIGURE 4.1: Training loss.

Our contributions are as follows:

- We empirically demonstrate that a simple modification made in the Transformer's official implementation (Vaswani et al., 2018) which changes the computation order of residual connection and layer normalization can effectively ease its optimization.
- We deeply analyze how the subtle difference of computation order affects convergence in deep Transformers, and propose to initialize deep Transformers under the Lipschitz constraint.
- In contrast to previous work, we empirically show that with proper parameter initialization, deep Transformers with the original computation order can converge.
- Our simple approach effectively ensures the convergence of deep Transformers with up to 24 layers, and achieves +1.50 and +0.92 BLEU improvements over the baseline on the WMT 14 English to German task and the WMT 15 Czech to English task.
- We further investigate deep decoders for the Transformer in addition to the deep encoders studied in previous work, and show that deep decoders can also benefit the Transformer.

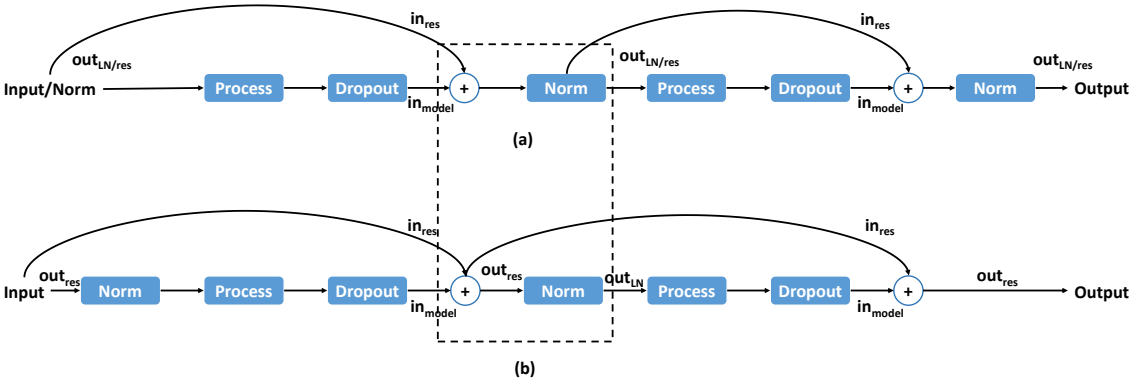


FIGURE 4.2: Two computation sequences of Transformer translation models: (a) the one used in the original paper, (b) the official implementation. We suggest to regard the output of layer normalization ($out_{LN/res}$) as the output of residual connection rather than the addition of in_{res} and in_{model} for (a), because it ($out_{LN/res}$) is the input (in_{res}) of the next residual connection computation.

4.2 Convergence of Different Computation Orders

We plot the loss averaged over every 500 training steps of the 6-layer Transformer model (Vaswani et al., 2017) and of the corresponding 12-layer model on the WMT 14 English-German training set in Figure 4.1.

Figure 4.1 shows that the deeper model is not over-fitting on the training set, it in fact performs worse than its shallower counterpart even on the training set.

In this paper, we focus on the convergence of the training of deep Transformers and the factors that prevent them from convergence (as opposed to other important issues such as over-fitting on the training set). To alleviate the training problem for the standard Transformer model, Layer normalization (Ba et al., 2016) and residual connection (He et al., 2016) are adopted.

4.2.1 Empirical Study of the Convergence Issue

The official implementation (Vaswani et al., 2018) of the Transformer uses a different computation order (Figure 4.2 b) compared to the published version (Vaswani et al., 2017) (Figure 4.2 a), since it (Figure 4.2 b) seems better for harder-to-learn models. [1] Even though several studies (Chen et al., 2018b; Domhan, 2018) have mentioned this change and

[1] `https://github.com/tensorflow/tensor2tensor/blob/v1.6.5/tensor2tensor/layers/common_hparams.py#L110-L112`.

although Wang et al. (2019c) analyze the difference between the two computation orders during backpropagation, and Zhang et al. (2019a) point out the effects of normalization in their work, how this modification impacts on the performance of the Transformer, especially for deep Transformers, has not been deeply studied before. Here we present both empirical convergence experiments (Table 4.1) and a theoretical analysis of the effect of the interaction between layer normalization and residual connection (Table 4.2).

In order to compare with Bapna et al. (2018), we used the same datasets from the WMT 14 English to German task and the WMT 15 Czech to English task for our experiments. Duplicate data in the training set were removed. [2] All data were tokenized and truecased with Moses (Koehn et al., 2007). The concatenation of newstest 2012 and newstest 2013 was used for validation and newstest 2014 as the test set for the English to German task, and newstest 2013 as the validation set and newstest 2015 as the test set for the Czech to English task.

Parameters were initialized with Glorot Initialization (Glorot and Bengio, 2010) like in many other Transformer implementations (Klein et al., 2017; Hieber et al., 2017; Vaswani et al., 2018):

$$Uniform(-\sqrt{\frac{6}{(isize + osize)}}, +\sqrt{\frac{6}{(isize + osize)}}) \tag{4.1}$$

where $isize$ and $osize$ are the two dimensions of the matrix.

We conducted experiments based on the Neutron implementation (Xu and Liu, 2019) of the Transformer translation model described in Chapter 6.

We applied joint Byte-Pair Encoding (BPE) (Sennrich et al., 2016a) with $32k$ merge operations to address the unknown word issue. We only kept sentences with a maximum of 256 sub-word tokens for training. The training set was randomly shuffled in every training epoch.

The number of warmup steps was set to $8k$, [3] and each training batch contained at least $25k$ target tokens. We used a dropout of 0.1. We used the Transformer Base setting (Vaswani et al., 2017) where the embedding dimension and the hidden dimension of the

[2] We only kept those most frequent source sentences for the same translation, and vice versa.

[3] `https://github.com/tensorflow/tensor2tensor/blob/v1.15.4/tensor2tensor/models/transformer.py#L1818`.

Models	Layers		En-De		Cs-En	
	Encoder	Decoder	v1	v2	v1	v2
Bapna et al. (2018)*	16	6	28.39	None	29.36	None
Wang et al. (2019c)	30	6	29.3			
Wu et al. (2019c)	8		29.92	None		
Zhang et al. (2019a)	20		28.67			
Transformer*	6		27.77‡	27.31	28.62	28.40
	12		¬	28.12	¬	29.38
	18		¬	28.60	¬	29.61
	24		¬	**29.02**	¬	**29.73**

TABLE 4.1: Results of different computation orders. "¬" means fail to converge, "None" means not reported in original works, "*" indicates our implementation of their approach. † and ‡ mean $p < 0.01$ and $p < 0.05$ while comparing between v1 (the official publication) and v2 (the official implementation) with the same number of layers in the significance test. Wu et al. (2019c) use the Transformer Big setting, while the others are based on the Transformer Base Setting. Zhang et al. (2019a) use merged attention decoder layers with a $50k$ batch size.

position-wise feed-forward neural network were 512 and $2,048$ respectively. We employed a label smoothing (Szegedy et al., 2016) value of 0.1. We used the Adam optimizer (Kingma and Ba, 2015) with 0.9, 0.98 and 10^{-9} as β_1, β_2 and ϵ. We followed Vaswani et al. (2017) for the other settings.

We trained the model on 2 GTX 1080 Ti GPUs, and performed decoding on 1 of them. We used a batch size of $25k$ target tokens which was achieved through gradient accumulation of small batches, and the model was trained for $100k$ training steps.

We used a beam size of 4 for decoding, and evaluated tokenized case-sensitive BLEU with the averaged model of the last 5 checkpoints saved with an interval of $1,500$ training steps.

Results of the two different computation orders are shown in Table 4.1, which shows that deep Transformers with the computation order of the official implementation (v2) have no convergence issue.

v1	v2
$\mu = mean(in_{model} + in_{res})$	
$\sigma = std(in_{model} + in_{res})$	
$out_{LN} = \frac{(in_{model}+in_{res}-\mu)}{\sigma} * w + b$	
$out^{v1}_{res} = out_{LN} = \frac{w}{\sigma} * out^{v2}_{res} - \frac{\mathrm{w}}{\sigma} * \mu + b$	$out^{v2}_{res} = in_{res} + in_{model}$

TABLE 4.2: Computation with layer normalization and residual connection. v1 and v2 stand for the computation order of the original Transformer paper and that of the official implementation respectively. "mean" and "std" are the computation of mean value and standard deviation. in_{model} and in_{res} stand for output of current layer and accumulated outputs from previous layers respectively. w and b are the trainable weight and bias of layer normalization which are initialized with a vector full of 1s and another vector full of 0s. out_{LN} is the computation result of the layer normalization. out^{v1}_{res} and out^{v2}_{res} are results of residual connections of v1 and v2.

4.2.2 Theoretical Analysis

Since the subtle change of computation order results in large differences in convergence, we further analyze the differences between the computation orders to investigate how they affect convergence.

We conjecture that the convergence issue of deep Transformers is perhaps due to the fact that layer normalization over residual connections in Figure 4.2 (a) effectively reduces the impact of residual connections due to the subsequent layer normalization, in order to avoid a potential explosion of combined layer outputs (Chen et al., 2018b), which is also studied by Wang et al. (2019c); Zhang et al. (2019a). We therefore investigate how the layer normalization and the residual connection are computed in the two computation orders, shown in Table 4.2.

Table 4.2 shows that the computation of residual connection in v1 is weighted by $\frac{w}{\sigma}$ compared to v2, and the residual connection of previous layers will be shrunk if $\frac{w}{\sigma} < 1.0$, which makes it difficult for deep Transformers to converge.

We suggest that Bapna et al. (2018) introduce the TA mechanism to compensate normalized residual connections through combining outputs of shallow layers to the final encoder output for the published Transformer, to further obtain significant improvements

with Transformers with deep encoders. Wang et al. (2019c) additionally aggregate outputs of all preceding encoder layers for each encoder layer instead of only aggregating for decoder layers.

The layer aggregation approach (Yu et al., 2018) may also help alleviate training problems in a way similar to the transparent attention approach, but significantly more parameters will be introduced and Dou et al. (2018, 2019) only focused on benefiting from combining shallow layers' representations through aggregating layers of the 6-layer baseline Transformer.

4.3 Lipschitz Constrained Parameter Initialization

Since the diminished residual connections (Table 4.2) may cause the convergence issue of deep v1 Transformers, is it possible to constrain $\frac{w}{\sigma} \geq 1$? Given that w is initialized with 1, we suggest that the standard deviation of $in_{model} + in_{res}$ should be constrained as follows:

$$0 < \sigma = std(in_{model} + in_{res}) \leq 1 \tag{4.2}$$

in which case $\frac{w}{\sigma}$ will be greater than or at least equal to 1, and the residual connection of v1 will not be shrunk anymore. To achieve this goal, we can constrain elements of $in_{model} + in_{res}$ to be in $[a, b]$ and ensure that their standard deviation is smaller than 1.

Let's define $P(x)$ as any probability distribution of x between $[a, b]$:

$$\int_a^b P(x)dx = 1 \tag{4.3}$$

then the standard deviation of x is:

$$\sigma(P(x), x) = \sqrt{\int_a^b P(x)\Big(x - \int_a^b P(x)xdx\Big)^2 dx} \tag{4.4}$$

Given that $(x - \int_a^{\mathrm{b}} P(x)xdx) < (b-a)$ for x $\in [a, b]$ as $P(x)$ is constrained by Equation 4.3, we reformulate Equation 4.4 as follows:

$$\sigma(P(x), x) < \sqrt{\int_a^{\mathrm{b}} P(x)(b-a)^2 dx} \tag{4.5}$$

From Equation 4.5 we obtain:

$$\sigma(P(x), x) < (b-a)\sqrt{\int_a^{\mathrm{b}} P(x)dx} \tag{4.6}$$

After applying Equation 4.3 in Equation 4.6, we find that:

$$\sigma(P(x), x) < b - a \tag{4.7}$$

Thus, as long as $b-a \leq 1$ (the range of elements of the representation x), the requirements for the corresponding σ described in Equation 1 can be satisfied.

To achieve this goal, we can simply constrain the range of elements of x to be smaller than 1 and initialize the sub-model before layer normalization to be a k-Lipschitz function, where $k \leq 1$. Because if the function F of the sub-layer is a k-Lipschitz function, for inputs x, y $\in$ [a, b], $|F(x) - F(y)| < k|x - y|$ holds. Given that $|x - y| \leq b - a$, we can get $|F(x) - F(y)| < k(b - a)$, the range of the output of that sub-layer is constrained by making it a k-Lipschitz function with constrained input.

The k-Lipschitz constraint can be satisfied effectively through weight clipping, [4] but we empirically find that deep Transformers are only hard to train at the beginning and only applying a constraint to parameter initialization is sufficient, which is more efficient and can avoid a potential risk of weight clipping on performance. Zhang et al. (2019a) also show that decreasing parameter variance at the initialization stage is sufficient for ensuring the convergence of deep Transformers, which is consistent with our observation.

[4]Note that the weight of the layer normalization cannot be clipped. Otherwise, residual connections will be more heavily shrunk.

Layers	En-De		Cs-En	
	v1-L	v2-L	v1-L	v2-L
6	27.96†	27.38	28.78‡	28.39
12	28.67†	28.13	29.17	29.45
18	29.05‡	28.67	29.55	29.63
24	**29.46**	29.20	29.70	**29.88**

TABLE 4.3: Results with Lipschitz constrained parameter initialization.

4.4 Experiments

We use the training data described in Section 4.2.1 to examine the effectiveness of the proposed Lipschitz constrained parameter initialization approach.

In practice, we initialize embedding matrices and weights of linear transformations with uniform distributions of $[-e, +e]$ and $[-l, +l]$ respectively. We use $\sqrt{\frac{2}{esize+vsize}}$ as e and $\sqrt{\frac{1}{isize}}$ as l where $esize$, $vsize$ and $isize$ stand for the size of embedding, vocabulary size and the input dimension of the linear transformation respectively. [5]

Results for the two computation orders with the new parameter initialization method are shown in Table 4.3. v1-L indicates v1 with Lipschitz constrained parameter initialization, the same for v2-L.

Table 4.3 shows that deep v1-L models do not suffer from convergence problems anymore with our new parameter initialization approach. It is also worth noting that unlike Zhang et al. (2019a), our parameter initialization approach does not degrade the translation quality of the 6-layer Transformer, and the 12-layer Transformer with our approach already achieves performance comparable to the 20-layer Transformer in Zhang et al. (2019a) (shown in Table 4.1).

While previous approaches (Bapna et al., 2018; Wang et al., 2019c) only increase the depth of the encoder, we suggest that deep decoders should also be helpful. We analyzed the influence of deep encoders and decoders separately with the original paper (Vaswani et al., 2017)'s computation order, and results are shown in Table 4.4.

[5] To preserve the magnitude of the variance of the weights in the forward pass.

Encoder	Decoder	En-De	Cs-En
6		27.96	28.78
24	6	28.76	29.20
6	24	28.63	29.36
24		**29.46**	**29.70**

TABLE 4.4: Effects of encoder and decoder depth with Lipschitz constrained parameter initialization (v1-L).

Table 4.4 shows that the deep decoder can indeed benefit performance in addition to the deep encoder, especially on the Czech to English task.

4.5 Related Work

4.5.1 Deep NMT

Zhou et al. (2016) introduce the fast-forward connection, based on deep LSTM networks, and an interleaved bi-directional architecture for stacking the LSTM layers. Fast-forward connections play an essential role in propagating the gradients and building a deep topology of depth 16.

Given that NMT with deep architecture in its encoder or decoder RNNs often suffer from severe gradient diffusion due to the non-linear recurrent activations, optimization of such networks is often very difficult. Wang et al. (2017b) propose a novel Linear Associative Unit (LAU) to reduce the gradient propagation path inside the recurrent unit. Different from conventional approaches (LSTM unit and GRU), LAU uses linear associative connections between input and output of the recurrent unit, thus allowing unimpeded information flow through both space and time.

4.5.2 Deep Transformers

Bapna et al. (2018) attempt to train significantly (2-3x) deeper Transformer and Bi-RNN encoders for machine translation. They show that Transformer models with more than 12 encoder layers fail to converge, and propose the Transparent Attention (TA) mechanism

which improves gradient flow during backpropagation by allowing each decoder layer to attend weighted combinations of all encoder layer outputs, instead of just the top encoder layer.

Wang et al. (2019c) find that by relocating the layer normalization unit, Transformers with deep encoders can be optimized smoothly. They propose the Dynamic Linear Combination of Layers (DLCL) approach which additionally aggregates previous layers' outputs for each encoder layer to memorize the features extracted from all preceding layers, suggesting it overcomes the problem with the standard residual network where a residual connection just relies on the output of one-layer below and may forget the earlier layers. They successfully train a 30-layer encoder, surpassing the 16-layer encoder of Bapna et al. (2018).

Wu et al. (2019c) propose an effective two-stage approach with three specially designed components to construct deeper NMT models, which incrementally increases the depth of the encoder and the decoder of the Transformer Big model by freezing both parameters and the encoder-decoder attention computation of pre-trained shallow layers, and stacking 2 new encoder and decoder layers upon frozen layers.

Zhang et al. (2019a) perform empirical analysis which suggests that the convergence of deep Transformers is poor due to gradient vanishing caused by the interaction between residual connection and layer normalization, and propose the layer-wise Depth-Scaled Initialization (DS-Init) approach, which decreases parameter variance at the initialization stage, and reduces output variance of residual connections so as to ease gradient backpropagation through normalization layers. To reduce the computational cost of decoder layers, they additionally propose the Merged Attention sub-layer (MAtt) which combines a simplified average-based self-attention sub-layer and the encoder-decoder attention sub-layer on the decoder side.

4.6 Conclusion

In contrast to previous studies (Bapna et al., 2018; Wang et al., 2019c; Wu et al., 2019c) which show that deep Transformers with the computation order as in Vaswani et al.

(2017) have difficulty in convergence, we show that deep Transformers with the original computation order can converge as long as proper parameter initialization is performed.

We first investigate convergence differences between the published Transformer (Vaswani et al., 2017) and its official implementation (Vaswani et al., 2018), and compare the differences of computation orders between them. We conjecture that the convergence issue of deep Transformers is because layer normalization sometimes shrinks residual connections. We support our conjecture with a theoretical analysis (Table 4.2), and propose a Lipschitz constrained parameter initialization approach for solving this problem.

Our experiments show the effectiveness of our simple approach on the convergence of deep Transformers, which achieves significant improvements on the WMT 14 English to German and the WMT 15 Czech to English news translation tasks. We also study the effects of deep decoders in addition to deep encoders extending previous studies.

Chapter 5

Dynamically Adjusting Transformer Batch Size by Monitoring Gradient Direction Change

The choice of hyperparameters affects the performance of neural models. While much previous research (Sutskever et al., 2013; Duchi et al., 2011; Kingma and Ba, 2015) focuses on accelerating convergence and reducing the effects of the learning rate, comparatively few papers concentrate on the effect of batch size.

In this chapter, we address **RQ6:** *How to dynamically and automatically find proper and efficient batch sizes during training?* and **RQ7:** *How to efficiently monitor gradient direction change?*

Specifically, we analyze how increasing batch size affects gradient direction, and propose to evaluate the stability of gradients with their angle change. Based on our observations, the angle change of gradient direction first tends to stabilize (i.e., gradually decrease) while accumulating mini-batches, and then starts to fluctuate.

We propose to automatically and dynamically determine batch sizes by accumulating gradients of mini-batches and performing an optimization step at just the time when the direction of gradients starts to fluctuate.

To improve the efficiency of our approach for large models, we propose a sampling approach to select gradients of parameters sensitive to the batch size. Our approach dynamically determines proper and efficient batch sizes during training.

In our experiments on the WMT 14 English to German and English to French tasks, our approach improves over the Transformer with a fixed $25k$ batch size by $+0.73$ and $+0.82$ BLEU respectively.

5.1 Introduction

The performance of neural models is likely to be affected by the choice of hyperparameters. While many previous studies (Sutskever et al., 2013; Duchi et al., 2011; Kingma and Ba, 2015) focus on accelerating convergence and reducing the effects of the learning rate, comparatively few papers concentrate on the effect of batch size.

However, the batch size is also an important hyperparameter, and some batch sizes empirically lead to better performance than others.

Specifically, it has been shown that the performance of the Transformer model (Vaswani et al., 2017) for Neural Machine Translation (Bahdanau et al., 2014; Gehring et al., 2017) relies heavily on the batch size (Popel and Bojar, 2018; Ott et al., 2018; Abdou et al., 2017; Zhang et al., 2019a).

The influence of batch size on performance raises the question, how to dynamically find proper and efficient batch sizes during training? In this chapter, we investigate the relationship between the batch size and gradients, and propose a dynamic batch size approach by monitoring gradient direction changes. Our contributions are as follows:

- We observe the effects on gradients with increasing batch size, and find that a large batch size stabilizes the direction of gradients.
- We propose to automatically determine dynamic batch sizes in training by monitoring the gradient direction change while accumulating gradients of small batches.
- To measure gradient direction change efficiently with large models, we propose an approach to dynamically select those gradients of parameters/layers which are sensitive to the batch size.

- In machine translation experiments, our approach improves the training efficiency and the performance of the Transformer model.

5.2 Gradient Direction Change and Automated Batch Size

Gradients indicate the direction and size of parameter updates to minimize the loss function in training. To reveal the effects of the batch size in optimization, we evaluate its influence on the direction change of gradients.

5.2.1 Gradient Direction Change with Increasing Batch Size

To investigate the influence of batch size on gradient direction, we gradually accumulate gradients of small mini-batches as the gradients of a large batch that consists of those mini-batches, and observe how the direction of gradients varies.

Let $d_i^j : (x_i^j, y_i^j)$ stand for the large batch concatenated from the ith mini-batch to the jth mini-batch, where x_i^j and y_i^j are inputs and targets. Then the gradients g_i^j of model parameters θ on d_i^j are:

$$g_i^j = \frac{\partial L(\theta, x_i^j, y_i^j)}{\partial \theta} \tag{5.1}$$

In gradient accumulation, the gradients g_0^k are the sum of g_0^{k-1} and g_k^k:

$$g_0^k = g_0^{k-1} + g_k^k \tag{5.2}$$

To measure the change of gradient direction during accumulation, we regard the two gradients g_0^{k-1} and g_0^k as 2 vectors, and compute the angle $a(g_0^{k-1}, g_0^k)$ between them:

$$a(g_0^{k-1}, g_0^k) = \arccos\left(\frac{g_0^{k-1} \bullet g_0^k}{|g_0^{k-1}||g_0^k|}\right) \tag{5.3}$$

where "$\bullet$" indicates the inner-product of vectors.

We use the angle of 2 vectors rather than cosine similarity because:

k	1	2	3	4	5	6	7	8	9	10
Size	4064	8994	12768	17105	21265	25571	29411	33947	38429	43412
$a(g_0^{k-1}, g_0^k)$		51.52	30.37	27.42	22.61	20.87	19.80	19.59	18.92	19.23
$a(g_0^{k-3}, g_0^k)$				59.53	44.20	41.77	35.34	32.19	32.10	34.29

TABLE 5.1: The direction change of gradients while accumulating mini-batches.

- The angle indicates the change between gradient directions.
- When the angle is small, a significant change in the angle only results in a subtle difference in cosine similarity. [1]

We observe the gradient direction varying during accumulating gradients of a Transformer model training on the WMT 14 English-German task following the settings of Vaswani et al. (2017) except for a batch size of around $50k$ target tokens. To achieve the gradient of the large batch size, we gradually accumulate gradients of mini-batches with around $4k$ target tokens.

Table 5.1 shows a typical example: (i) gradient change is big at the beginning, (ii) gradient change reduces with increasing batch size, and (iii) eventually it will start fluctuating (here at k=10). [2]

Intuitively, the less the direction of accumulated gradients is moved by the gradients of a new mini-batch, the more certainty there is about the gradient direction. Thus we propose that the magnitude of the angle fluctuation relates to the certainty of the model parameter optimization direction, and may therefore serve as a measure of optimization difficulty.

5.2.2 Automated Batch Size with Gradient Direction Change

Table 5.1 shows that the optimization direction is less stable with a small batch than with a large batch. But after the direction of gradients has stabilized, accumulating more mini-batches seems useless as the gradient direction starts to fluctuate.

[1] $cos(5°) \approx 0.9961$, $cos(10°) \approx 0.9848$.

[2] By comparing $\sum_{i=0}^{n} a(g_0^{k-i-1}, g_0^{k-i})$ with $a(g_0^{k-n-1}, g_0^k)$, we can find the direction changes from g_0^{k-i-1} to g_0^k are inconsistent. Otherwise, $\sum_{i=0}^{n} a(g_0^{k-i-1}, g_0^{k-i}) \approx a(g_0^{k-n-1}, g_0^k)$.

Thus, we suggest to compute dynamic and efficient batch sizes by accumulating gradients of mini-batches, while evaluating the gradient direction change with each new mini-batch, and stop accumulating more mini-batches and perform an optimization step when the gradient direction fluctuates.

In practice, we only monitor $a(g_0^{k-1}, g_0^k)$ for efficiency. We record the minimum angle change a_{min} while accumulating gradients, and suppose that the gradient direction starts to fluctuate and stop accumulating more mini-batches when $a(g_0^{k-1}, g_0^k) > a_{min} * \alpha$. In this way, we can achieve a dynamic batch size (the size of d_0^k), where α is a pre-specified hyperparameter.

5.2.3 Efficiently Monitoring Gradient Direction Change

In practice, a model may have a large number of parameters, and the cost for computing the cosine similarity between two corresponding gradient vectors is relatively high. To tackle this issue, we propose to divide model parameters into groups, and monitor gradient direction change only on a selected group in each optimization step. For a multi-layer model, i.e., the Transformer, a group may consist of parameters of 1 layer or several layers.

To select the parameter group which is sensitive to the batch size, we record the angles of gradient direction change $a(g_0^0, g_0^1), ..., a(g_0^{k-1}, g_0^k)$ during the gradient accumulation, and define a_{max} and a_{min} as the maximum and minimum direction change:

$$a_{max} = max(a(g_0^0, g_0^1), ..., a(g_0^{k-1}, g_0^k)) \tag{5.4}$$

$$a_{min} = min(a(g_0^0, g_0^1), ..., a(g_0^{k-1}, g_0^k)) \tag{5.5}$$

We then use Δa to measure the uncertainty reduction in the optimization direction:

$$\Delta a = a_{max} - a_{min} \tag{5.6}$$

Intuitively, the optimization direction of the parameter group which results in a larger Δa profits more from the batch size, and the group with a larger Δa should be more frequently sampled.

We average the recent history of Δa_k of the kth parameter group into $\overline{\Delta a_k}$. Inspired by Gumbel (1954); Maddison et al. (2014); Zhang et al. (2019d), we first add Gumble noise to each $\overline{\Delta a_k}$ to prevent the selection falling into a fixed group:

$$\Delta a_k^* = \overline{\Delta a_k} - \log(-\log u) \tag{5.7}$$

where $u \in (0, 1)$ is a uniform distribution.

Then we zero negative values [3] in Δa_1^*, ..., Δa_n^* and normalize them into a probability distribution:

$$p_k = \frac{\Delta a_k^{*\beta}}{\sum_{i=1}^{n} \Delta a_i^{*\beta}} \tag{5.8}$$

We use p_k as the probability to sample the kth group, and β is a hyperparameter to sharpen the probability distribution. We do not use softmax because it would heavily sharpen the distribution when the gap between values is large, and makes it almost impossible to select and evaluate the other groups in addition to the one with the highest Δa_k^*. [4]

5.3 Experiments

We implemented our approaches based on the Neutron implementation (Xu and Liu, 2019) of the Transformer translation model. We applied our approach to the training of the Transformer, and to compare with Vaswani et al. (2017), we conducted our experiments on the WMT 14 English to German and English to French news translation tasks on 2 GTX 1080Ti GPUs. The concatenation of newstest 2012 and newstest 2013 was used for validation and newstest 2014 as test sets for both tasks.

[3] Δa_k is positive, but after adding Gumble noise, there is a small possibility that it turns negative. In our case, negative values only occur very few times.

[4] For example, the result of softmax over [22, 31, 60] is [3.13e-17, 2.54e-13, 1.00], the last element takes almost all possibility mass. But we later find that if Δa is normalized ($\Delta a = (a_{max} - a_{min})/a_{max}$) in Equation 5.6, the softmax works comparably well, which avoids using the hyperparameter β in Equation 5.8.

Batch Size	En-De	En-Fr	Time
25k	27.38	39.34	35h21m
50k	27.93	39.97	60h38m
dyn	**28.11**†	**40.16**†	**33h37m**

TABLE 5.2: Performance. Time is the training time on the WMT 14 En-De task for $100k$ training steps. † indicates $p < 0.01$ in the significance test.

We applied joint Byte-Pair Encoding (BPE) (Sennrich et al., 2016a) with $32k$ merge operations to address the unknown word issue. We only kept sentences with a maximum of 256 sub-word tokens for training. All models were trained for $100k$ steps. The training set was randomly shuffled in every training epoch.

The number of warmup steps was set to $8k$. [5] We used a dropout of 0.1. We used the Transformer Base setting (Vaswani et al., 2017) of which the embedding dimension and the hidden dimension of the position-wise feed-forward neural network were 512 and $2,048$ respectively. We employed a label smoothing (Szegedy et al., 2016) value of 0.1. We used the Adam optimizer (Kingma and Ba, 2015) with 0.9, 0.98 and 10^{-9} as β_1, β_2 and ϵ. We followed all settings of Vaswani et al. (2017) except for the batch size.

We used a beam size of 4 for decoding, and evaluated case-sensitive tokenized BLEU [6] of the averaged model of the last 5 checkpoints with significance test (Koehn, 2004).

We used an α of 1.1 to determine the fluctuation of gradient direction by default. We regarded each encoder/decoder layer as a parameter group, and used a β of 3 for the parameter group selection. Hyperparameters were tuned on the development set.

5.3.1 Performance

We compared the results of our dynamic batch size approach to two fixed batch size baselines. The $25k$ batch size is the empirical value of Vaswani et al. (2017), while Zhang et al. (2019a) investigate $50k$ batch size. Results are shown in Table 5.2 with the statistics of batch sizes of our approach shown in Table 5.3 and the detailed distribution of batch sizes for the En-De task shown in Figure 5.1.

[5] https://github.com/tensorflow/tensor2tensor/blob/v1.15.6/tensor2tensor/models/transformer.py#L1850
[6] `https://github.com/moses-smt/mosesdecoder/blob/master/scripts/generic/multi-bleu.perl`

	En-De	En-Fr
min	7069	8025
avg	26264.19	30248.90
max	102165	103352

TABLE 5.3: Statistics of batch sizes.

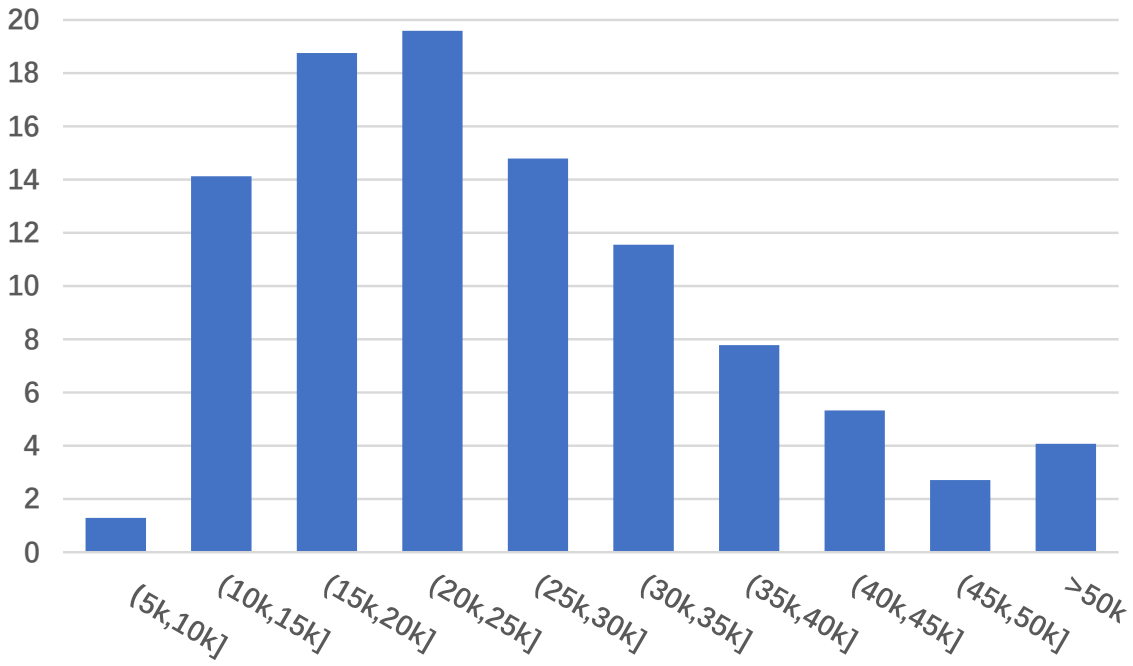


FIGURE 5.1: Distribution of dynamic batch sizes. Values on y-axis are percentages.

Table 5.2 and 5.3 show that our approach outperforms both the fixed $25k$ and $50k$ batch size settings with an average batch size of around $26k$, and our approach is slightly faster than the $25k$ setting despite the additional cost for monitoring gradient direction change. [7]

Figure 5.1 shows an interesting fact that the most frequently used automated batch sizes were close to the fixed value ($25k$) of Vaswani et al. (2017).

5.3.2 Analysis of Minimum Gradient Direction Change

In order to observe the varying minimum gradient direction change during training, we averaged the minimum angle for every $2.5k$ training steps. Results are shown in Figure

[7] It is hard to accumulate an accurate $25k$ target tokens in a batch, and in fact, the fixed $25k$ setting results in an average batch size of 26729.79.

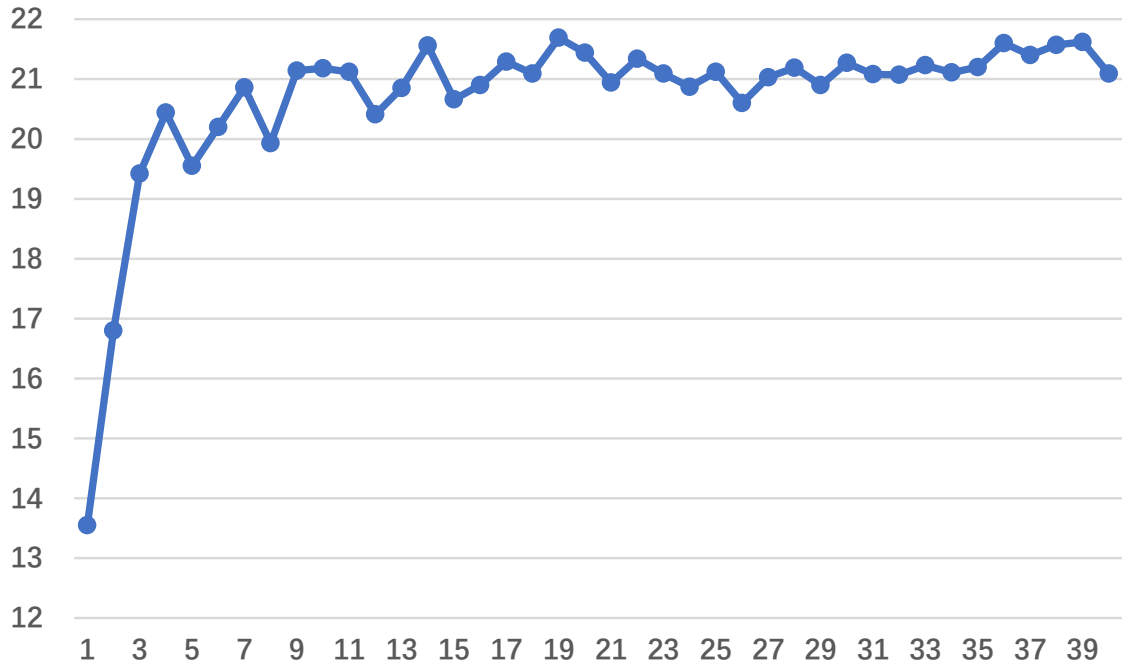


FIGURE 5.2: Minimum gradient direction change during training. X-axis 2.5k training steps, y averaged a_{min} (Equation 5.5).

α	Batch Size avg	Batch Size max	BLEU	Time
1.0	19367.76	60945	27.90	**24h50m**
1.1	26264.19	102165	28.11	33h37m
1.2	36208.47	164908	**28.39**	46h04m
1.3	51470.34	205210	28.37	63h56m

TABLE 5.4: Effects of different α.

5.2.

Figure 5.2 shows that the minimum direction change of gradients was small at the beginning, and gradually increased with training. Given that a small angle change indicates that there is more certainty in the gradient direction, this observation is consistent with the fact that finding the optimization direction is harder and harder with training.

5.3.3 Effects of α

We studied the effects of different α values on the En-De task, and results are shown in Table 5.4. [8]

Table 5.4 shows that with increasing α, the average batch size and the time cost increases along with the performance. A wide range of values works relatively well, indicating that its selection is robust, and 1.1 seems to be a good trade-off between the cost and the performance in our experiments. [9] It is also worth noting that $\alpha = 1$ outperforms the $25k$ baseline while being 1.42 times faster (Table 5.2).

5.4 Related Work

Popel and Bojar (2018) demonstrate that the batch size affects the performance of the Transformer, and a large batch size tends to benefit performance, but they use fixed batch sizes during training. Abdou et al. (2017) propose to use a linearly increasing batch size from 65 to 100 which slightly outperforms their baseline. Smith et al. (2018) show that the same learning curve on both training and test sets can be obtained by increasing the batch size during training instead of decaying the learning rate.

For fast convergence, Balles et al. (2017) propose to approximately estimate the mean value of the batch size for the next batch by maximizing the expected gain with a sample gradient variance ($||g||^2$) computed on the current batch, while our approach compares the gradient direction of change ($a(g_0^{k-1}, g_0^k)$) during accumulation of mini-batches in the assembling of a large batch.

We suggest our approach is complementary to Sutskever et al. (2013); Duchi et al. (2011); Kingma and Ba (2015), as their approaches decide the magnitude of the movement in the optimization direction, while our approach provides reliable gradient direction.

[8] We observed that the minimum batch size does not change significantly with increasing α, so we omit it for space.

[9] For $\alpha = 1.2$ on the En-Fr task, the corresponding values are: 44294.16, 185972, **40.35** and 54h12m.

5.5 Conclusion

In this chapter, we analyze the effects of accumulated batches on the gradient direction, and propose to achieve efficient automated batch sizes by monitoring change in gradient accumulation and performing an optimization step when the accumulated gradient direction is almost stable. To improve the efficiency of our approach with large models, we propose a sampling approach to select gradients of parameters sensitive to the batch size.

Our approach improves the Transformer with a fixed $25k$ batch size by +0.73 and +0.82 BLEU on the WMT 14 English to German and English to French tasks respectively while preserving efficiency.

Chapter 6

Neutron: an Implementation of the Transformer Translation Model and its Variants

The Transformer translation model is easier to parallelize and provides better performance compared to recurrent seq2seq models, which makes it popular among industry and the research community. We introduce our Neutron implementation [1] in this Chapter, providing the Transformer model and a wide range of recent variants targeting, amongst others, decoding speed, convergence, sentence and document context, advanced optimizers, unlimited batch size and high-performance parallelization modules. Neutron is highly optimized, easy to modify, and provides competitive performance with interesting features while keeping code readability. The code follows the design pattern of PyTorch and can be easily adapted to other projects.

6.1 Introduction

Vaswani et al. (2017) propose the Transformer architecture which contains only the attention mechanism and standard feed-forward neural networks augmented by residual connection, dropout, layer normalization and label smoothing loss for its training.

[1] We open-source our implementation at `https://github.com/anoidgit/transformer`.

The Transformer parallelizes better and outperforms previous RNN-based sequence-to-sequence models in many cases. It is widely applied in industry and attracts wide attention from researchers. As a result, many recent studies have been conducted based on the Transformer, and enhanced architectures have been proposed. Good code-bases are important to best support rapidly developing research.

Our implementation supports popular features provided in most Machine Translation (MT) libraries, including beam search, ensemble, length penalty and averaging of models.

In addition, we implement influential variants of the Transformer from recent research along with the standard implementation of the Transformer in Neutron, such as the average attention network (to accelerate the decoding of the Transformer) (Zhang et al., 2018a), the hierarchical layer aggregation (to reuse outputs of shallow encoder layers) (Dou et al., 2018), recurrent decoder (Chen et al., 2018b) and modeling sentential context (Wang et al., 2019e) for improving the MT quality, transparent attention (Bapna et al., 2018) to ensure the convergence of deep encoders and the document-level Transformer (Zhang et al., 2018c). We also support the more efficient training scheduler (dynamic sampling and review mechanism) proposed by Wang et al. (2018c), and other features, like some advanced optimizers (e.g., Lookahead, RAdam).

Besides, we support unlimited batch size with limited memory available on a single GPU by gradient accumulation which accumulates gradients of small mini-batches as the gradient of a large batch consisting of these small batches, which is important for the Transformer given that it has been shown that the performance of the Transformer model (Vaswani et al., 2017) relies heavily on the batch size (Popel and Bojar, 2018; Ott et al., 2018; Abdou et al., 2017; Zhang et al., 2019a). In addition to the large batch size support on a single GPU, we provide almost-from-scratch designed high-performance multi-GPU parallelization modules, which reduces unnecessary communication between GPUs and makes the parallelization over multiple GPUs more effective.

We will introduce features of the Neutron implementation in the next Section (6.2), its design in Section 6.3, performance in Section 6.4, followed by related work (in Section 6.5) and conclusion (in Section 6.6).

6.2 Features

We support a wide range of features for research purposes in the Neutron implementation, and we introduce them in this section.

6.2.1 Fundamental Features Supported

6.2.1.1 Basic Features

We support a wide range of fundamental features which are usually used in MT, including beam search with length penalty, ensemble of models, etc.

Beam search records the top-k translations during decoding and selects the translation with the highest overall probability rather than simply taking the token with the highest probability in each decoding step in greedy decoding. Since beam search helps keep some translations with higher overall scores even though their predicted probabilities in some decoding steps are not the highest, beam search normally results in better translation performance. Given that beam search is widely employed in many studies and supported by most MT toolkits, we also support beam search in the Neutron implementation. Our beam decoding is implemented at batch level, which performs beam search for a batch of source sentences rather than translating source sentence one-by-one, and is computationally efficient and friendly for GPUs.

Wu et al. (2016) suggest that beam search has to compare hypotheses of different length, and without some form of length normalization regular beam search will favor shorter results over longer ones on average since a negative log-probability is added at each step, yielding lower (more negative) scores for longer sentences. Thus they propose the length penalty, as shown in Equation 6.1.

$$\mathrm{lp}(Y) = \frac{5 + |Y|^{\alpha}}{(5+1)^{\alpha}} \tag{6.1}$$

where Y stands for the hypothesis decoded, and $|Y|$ indicates its length. α is a hyperparameter selected on the development set.

The score of each hypothesis of the beam results is normalized by dividing it by its length penalty factor. We support length penalty in the beam search implementation.

Compared to predicting with only one model, an ensemble of several models constructed by averaging their prediction probabilities in each decoding step usually leads to further improvements. It is widely employed in shared task submissions. We also support the ensemble of models.

6.2.1.2 Gradient Accumulation

As studied by Popel and Bojar (2018); Ott et al. (2018); Abdou et al. (2017); Zhang et al. (2019a), the batch size of the Transformer impacts its optimization and performance, Vaswani et al. (2017) use a batch size of approximately $25k$ source tokens and $25k$ target tokens which is distributed to 8 NVIDIA P100 GPUs. However, using many GPUs for one experiment is not viable in many cases, especially when training large models, which requires more GPU memory than a small model with the same batch size.

To achieve theoretically unlimited batch sizes with even only one GPU, we implement gradient accumulation which computes the gradients of several mini-batches one-by-one, and accumulates their gradients as the corresponding gradients of the large batch consisting of these small mini-batches.

6.2.1.3 Training Support

Label Smoothing Loss. We support the label smoothing loss (Szegedy et al., 2016) applied in the Transformer (Vaswani et al., 2017) which optimizes the KL-divergence rather than the perplexity and improves accuracy and BLEU score.

Learning Rate Scheduler. Vaswani et al. (2017) use a learning rate scheduler which increases the learning rate linearly for the first warmup training steps, and then decreases it after that proportionally to the inverse square root of the step number, as shown in Equation 6.2. We implement their learning rate scheduler for training.

$$\text{lrate}(step) = dmodel^{-0.5} * \min(step^{-0.5}, step * warmup\_steps^{-1.5}) \tag{6.2}$$

where *dmodel* and *warmup_step* are the input dimension of the Transformer and the number of warmup steps respectively.

6.2.1.4 Data Storage and Retrieval

In our NMT implementation, we introduce a data processing procedure to convert parallel corpora into tensors loadable for training and testing scripts. The processing includes the following steps:

1. sorting the bilingual corpora according to their length (number of source and target tokens);
2. building separate vocabularies or a shared vocabulary with the training data;
3. segmenting the training set into mini-batches;
4. padding and mapping mini-batches into tensors with corresponding vocabularies and saving them into an HDF5 format file on the hard drive.

We use the HDF5 format for the storage of tensors because it has the following advantages:

On-Disk Shuffling of the Whole Training Set. The training set is normally shuffled in each training epoch to provide a data distribution close to that of the whole training set in each optimization step, and to avoid that the particular part of the whole training period overfits the distribution represented in localized batches of data.

For NMT, data are usually sorted to gather translation pairs of similar lengths together to reduce the number of the special padding tokens introduced and corresponding computation waste. Shuffling is particularly important in this case. Otherwise, the model is likely to overfit into batches of similar lengths.

In many other NMT toolkits, the shuffling is normally addressed in two ways: 1) loading the whole training set into memory. Thus any parts of the data can be retrieved efficiently. 2) shuffling the whole training set, loading part of the training set into memory as the cache, and sorting the cache to reduce padding tokens.

While it is challenging for the first solution to load a large dataset (e.g., the training set for the WMT 14 English-French task which consists of $\sim 36M$ sentence pairs) with limited memory resources. For the second solution, there might be a gap between the distribution of the sorted cache and that of the full training set.

In our implementation, we first sort the full training set, then convert them into tensors, but we save tensors into an HDF5 format [2] file on the hard drive. The HDF5 file format and library [3] integrate a rich set of performance features that allow for access time and storage space optimizations, enabling us to retrieve any part of the training set from the hard drive like accessing the memory. Thus, we can shuffle very large datasets under low memory costs by loading batches of random indexes from HDF5 files.

Compression. HDF5 library supports gzip, lzf and szip compression algorithms. Gzip is one of these most popular compression algorithms which can provide a good compression ratio with moderate speed. Lzf provides a low to moderate compression rate, but it is very fast. Szip has both high speed and compression, but due to a patent-encumbered filter used in the NASA community, szip is not available with all installations of HDF5 for legal reasons.

Tensors converted from parallel corpora for machine translation are normally highly compressible for the following reasons:

1. Natural languages are normally compressible, as words and their co-occurrences are usually not uniformly distributed;

2. Tensors are an array of 32-bit integers, while in most cases, the vocabulary size for NLP is smaller than 65536 (2^{16}), which means that 16-bit is usually sufficient for saving vocabulary indexes, and the 32-bit integer word indexes are highly compressible.

We compress data sets to reduce I/O. Given that the dataset is normally saved once and will be read many times, we select the gzip algorithm available for almost all platforms for tensor compression by default as it provides a good compression ratio even though with

[2] `https://www.hdfgroup.org/`.
[3] `http://www.h5py.org/`.

only moderate compression speed, but it is configurable to select the other algorithms supported by the HDF5 library.

6.2.2 Multi-GPU Parallelization

The default multi-GPU parallelization model of PyTorch (Paszke et al., 2019) automatically synchronizes parameters during the forward pass and collects gradients after back-propagation. Specifically, it will automatically send model parameters to all utilized GPUs before the forward propagation and collect gradients from these GPUs after the backward propagation. However, when working with gradient accumulation, this results in two kinds of redundant communications between GPUs:

1. Even though small mini-batches of a large batch are computed one-by-one, parameters of the model do not change in the gradient accumulation before the optimization step is performed. Thus sending model parameters before the forward propagation to all involved GPUs is redundant for all mini-batches following the first mini-batch.

2. Collecting and accumulating gradients of all involved GPUs after the backward propagation of each mini-batch is unnecessary. Gradients computed on one involved GPU of a mini-batch can be accumulated only on that device without accumulating gradients from the other GPUs until the optimization step after backward propagating the last mini-batch.

We provide a new implementation of the parallelization model which requires explicit calls to distribute parameters and to accumulate gradients across GPUs but avoids the redundant communication and significantly accelerates the multi-GPU gradient accumulation case. Specifically, we distribute parameters only once after an optimization step, then perform independent forward propagation and backward propagation of mini-batches composing the large batch on all involved GPUs, finally we collect and accumulate gradients from all involved GPUs before performing the next optimization step. Our multi-GPU parallelization model also supports multi-GPU decoding which is normally not implemented in the other libraries.

In addition to the multi-GPU model parallelization feature, we also support the multi-GPU parallelization of the optimizer for the model's training. Thus, in our toolkit, all

computations for the model training are parallelized across multiple GPUs, not only the forward/backward propagation, but also optimization steps to update model parameters, while in many other toolkits, only the forward propagation and backward propagation of the model can be parallelized. In detail, we divide model parameters into n groups, where n is the number of GPUs utilized, and employ n optimizers respectively for parameter groups on corresponding GPU devices. Before each optimization step, the gradients of the group's parameters are accumulated from the other GPUs, and after the optimization step, the new parameters of the group are sent back to the other GPUs. This is particularly helpful for big models (e.g., Transformer Big and deep Transformers) and costly optimizers (e.g., Adam).

6.2.3 Models

Two Computation Orders. The official implementation of the Transformer (Vaswani et al., 2018) uses a different computation order than the Transformer described in the paper (Vaswani et al., 2017), which leads to differences in performance and convergence (Xu et al., 2020a). Specifically, in the original paper (Vaswani et al., 2017), for each multi-head attention sub-layer or the position-wise feed-forward neural network sub-layer, the computation of that a sub-layer is *processing* → *dropout* → *residual connection* → *layer normalization*, where processing stands for either the multi-head attention machinery or the feed-forward neural network. The official implementation (Vaswani et al., 2018) suggests that the computation order of *layer normalization* → *processing* → *dropout* → *residual connection*, seems better for harder-to-learn models. [4] While some other NMT implementations only implement one of the two computation orders, we support both computation orders in our implementation with empirical results and theoretical analysis described in Chapter 4.

Self-Attention with Relative Position. In contrast to recurrent and convolutional neural networks, the Transformer does not explicitly model relative or absolute position information in its architecture. Instead, it requires adding representations of absolute positions to its inputs. Shaw et al. (2018) present an alternative approach, extending the self-attention mechanism to efficiently consider representations of relative positions,

[4] `https://github.com/tensorflow/tensor2tensor/blob/v1.6.5/tensor2tensor/layers/common_hparams.py#L110-L112`.

or distances between sequence elements. They describe an efficient implementation of the method and cast it as an instance of relation-aware self-attention mechanisms that can generalize to arbitrary graph-labeled inputs. We implement their approach in our self-attention implementation, supporting both encoder and decoder.

Average Attention. With parallelizable attention networks, the neural Transformer is very fast to train. However, due to the self-attention in the decoder, Zhang et al. (2018a) suggest that the decoding procedure becomes slow. To alleviate this issue, they propose an average attention network as an alternative to the self-attention network in the decoder of the neural Transformer. The average attention network consists of two layers, with an average layer that models dependencies on previous positions and a gating layer that is stacked over the average layer to enhance the expressiveness of the proposed attention network. They apply this network on the decoder part of the neural Transformer to replace the original target-side self-attention model to accelerate decoding with almost no loss in training time and translation performance. We implement their approach in our work to provide support for potential practical applications.

Transparent Attention. Bapna et al. (2018) take the first step towards training extremely deep models for translation, by training deep encoders for Transformer and LSTM-based models. They find that the vanilla Transformer models completely fail to train when increasing the encoder depth. To ease optimization, they propose the transparent attention mechanism, which allows them to train deeper models which results in consistent gains on the WMT 14 English to German and WMT 15 Czech to English tasks. We implement their approach (Bapna et al., 2018) which ensures the convergence of deep Transformer encoders for our study on the convergence of deep Transformers (in Chapter 4).

Hierarchical Layer Aggregation. Advanced NMT models generally implement encoder and decoder as multiple layers, which allows systems to model complex functions and capture complicated linguistic structures. However, Dou et al. (2018) suggest that only the top layers of encoder and decoder are leveraged in the subsequent process, which misses the opportunity to exploit useful information embedded in other layers. They propose to simultaneously expose all of these signals with layer aggregation and multi-layer

attention mechanisms. In addition, they introduce an auxiliary regularization term to encourage different layers to capture diverse information. We implement their hierarchical layer aggregation model which leads to the best performance in their paper while investigating the effects of shallow layer's outputs in machine translation.

RNMT Decoder. Chen et al. (2018b) quantify the effect of several modeling improvements (including multi-head attention and layer normalization) as well as optimization techniques (such as synchronous replica training and label smoothing), which are used in recent architectures. They demonstrate that these techniques are applicable across different model architectures. Inspired by their findings of the relative strengths and weaknesses of individual model architectures, they propose new model architectures that combine the RNMT decoder with the Transformer encoder. We implement the recurrent decoder in their work which may lead to improved performance compared to the Transformer decoder. We also support other recurrent models like ATR (Zhang et al., 2018b).

Sentential Context. Wang et al. (2019e) show that a shallow sentential context extracted from the top encoder layer only can improve translation performance via contextualizing the encoding representations of individual words. Next, they introduce a deep sentential context, which aggregates the sentential context representations from all of the internal layers of the encoder to form a more comprehensive context representation. We implement their approach using the transparent attention model to aggregate sentence representations in our work.

Learning Source Phrase Representations. In Chapter 3, we (Xu et al., 2020c) improve the long-distance dependency capturing ability of the Transformer by incorporating source phrase representations. Specifically, we propose an attentive feature extraction model and generate phrase representations based on token representations, and incorporate phrase representation learning into the Transformer to improve its long-distance relation capturing ability. We implement our approach, which is also naturally integrated as part of the Neutron implementation.

Document-level Transformer. Zhang et al. (2018c) extend the Transformer model with a new context encoder to represent document-level context, which is then incorporated into the original encoder and decoder by inserting corresponding cross-attention network(s) into each layer. As large-scale document-level parallel corpora are usually not available, they introduce a two-step training method to take full advantage of abundant sentence-level parallel corpora and limited document-level parallel corpora. We implement their approach as a baseline for context-aware NMT.

6.2.4 Advanced Features

Lipschitz Constrained Parameter Initialization. In Chapter 4 and in Xu et al. (2020a), after the theoretical analysis of the convergence issue of deep Transformers of the original computation order, we propose to ensure the convergence of deep Transformers by the Lipschitz constrained parameter initialization approach. Unlike in previous work (Zhang et al., 2019a) our approach does not degrade the performance of the 6-layer Transformer while ensuring the convergence of deep Transformers with significant BLEU improvements. We integrate this parameter initialization approach as the default in our implementation.

Dynamic Batch Sizes. In Chapter 5 and in Xu et al. (2020b), we first analyze how increasing batch size affects gradient direction, and propose to evaluate the stability of gradients with their angle change. Based on our observations, the angle change of gradient direction first tends to stabilize (i.e., gradually decrease) while accumulating mini-batches, and then starts to fluctuate. Thus, we propose to automatically and dynamically determine batch sizes by accumulating gradients of mini-batches and performing an optimization step at just the time when the direction of gradients starts to fluctuate. In addition, we propose a sampling approach to select gradients of parameters sensitive to the batch size to improve the efficiency of our approach for large models. We support our dynamic batch size approach in our Neutron implementation.

Reducing Optimization Difficulty. We suggest that the biases of linear transformations for projection between representation spaces and before the layer normalization are redundant, as the layer normalization (Ba et al., 2016) will add another bias. Thus,

we suggest that removing redundant biases may reduce the computation costs along with these bias parameters, will go hand-in-hand with a small acceleration, and may ease the optimization of models.

Dynamic Sentence Sampling. Traditional NMT involves a fixed training procedure where each sentence is sampled once during each epoch. Wang et al. (2018c) suggest that in reality, some sentences are well-learned during the initial few epochs, and the well-learned sentences would continue to be used in training along with those sentences that were not well learned for 10-30 epochs, which results in a waste of time and effort. They propose an efficient method to dynamically sample the sentences in order to accelerate NMT training. In their approach, a weight is assigned to each sentence based on the measured difference between the training costs of two iterations. Further, in each epoch, a certain percentage of sentences are dynamically sampled according to their weights. We implement their approaches, including a dynamic sampling and review mechanism, in the training script.

Activation Functions. Though Devlin et al. (2019) use the Transformer encoder for BERT, they replace the Rectified Linear Unit (ReLU) activation function by the Gaussian Error Linear Unit (GELU) activation function (Hendrycks and Gimpel, 2016). Ramachandran et al. (2017) propose to leverage automatic search techniques to discover new activation functions. Using a combination of exhaustive and reinforcement learning-based search, they discover the Swish activation function. Although the Transformer uses the ReLU activation function in Vaswani et al. (2017), we provide a variety of activation functions, including GELU and Swish. Simply alternating the ReLU with these activation functions may provide further improvements.

Optimizers. The Transformer uses the Adam optimizer (Kingma and Ba, 2015) for its training by default. Given that the learning rate warmup heuristic achieves remarkable success in stabilizing training, accelerating convergence and improving generalization for adaptive stochastic optimization algorithms like RMSprop and Adam, Liu et al. (2020) study its mechanism in details. Pursuing the theory behind warmup, they identify a problem of the adaptive learning rate (i.e., it has a problematically large variance in the early

stage), suggest warmup works as a variance reduction technique, and provide both empirical and theoretical evidence to verify the hypothesis. They further propose RAdam, a new variant of Adam, by introducing a term to rectify the variance of the adaptive learning rate. Zhang et al. (2019c) propose the Lookahead optimization algorithm, which is orthogonal to previous approaches and iteratively updates two sets of weights. Intuitively, the algorithm chooses a search direction by looking ahead at the sequence of "fast weights" generated by another optimizer. They show that Lookahead improves the learning stability and lowers the variance of its inner optimizer with negligible computation and memory cost. We support both RAdam (Liu et al., 2020) and Lookahead (Zhang et al., 2019c) and their combination.

6.2.5 Data Cleaning

Normally, parallel corpora for training MT systems are not collected directly from translators at the sentence level. Some of the corpus data are crawled from the internet, and automatic sentence alignment tools are utilized to extract sentence-level translation pairs. As a result, there might be some incorrect translations (sentence pairs) in the parallel data. We provide some tools to clean data sets to alleviate this issue. Removing this noisy data will reduce the size of data and the corresponding vocabulary size, and usually leads to faster training with better performance than using the raw data in practice.

6.2.5.1 Max Keeper

Parallel data are usually the combination of several corpora, and these corpora may contain some common sentence pairs, which will be redundant after concatenation. Redundancy may also introduce biases into the dataset. A sentence pair that appears k times as frequently as another does not mean that it is also k times as important or correct as another. In addition to that, alignment tools may wrongly align the same source sentence to several translations in different parallel documents, especially for short sentences.

To remove redundant data, we implement a script that counts all sentences and their translations in the corpus, and only saves those translations with the highest frequency for each source sentence. We also replace potentially repeated blanks or tabulars into a single blank during cleaning to normalize the dataset.

6.2.5.2 Cleaning with Vocabulary

In the bilingual training data crawled from the Internet, some sentence pairs are meaningless or even do not belong to the language pairs.

We implement a vocabulary-based cleaning approach for this case. Specifically, we first collect the vocabulary and count frequencies of tokens on the training set. Then we filter the training set with a hyperparameter named *vratio*. We regard *vratio* tokens of the full vocabulary with lowest frequency counts as rare words, and if the percentage of rare tokens in a sentence is higher than another hyperparameter *vkratio*, we suggest that the sentence is unlikely to be part of the language pair, and we will remove it from the data.

6.2.5.3 Cleaning with Length Ratios

Length ratio is widely employed in the data pre-processing procedure for MT, since there are some incorrectly aligned sentence pairs in the training data which have abnormally large length ratios. We provide enhanced support at the sub-word level (Sennrich et al., 2016a), in addition to filtering tokenized texts.

Assume a sentence contains *nsub* tokens after BPE processing, *nsubadd* tokens are additionally produced by BPE separation, *nsep* tokens of the original tokenized sentence which has *ntok* tokens are segmented into sub-word units. The following ratios are defined at the monolingual level:

$$cratio = nsubadd/nsub \tag{6.3}$$

$$bratio = nsub/ntok \tag{6.4}$$

$$sratio = nsep/ntok \tag{6.5}$$

Assume a source sentence contains *nsubsrc* sub-word tokens and *nsrc* tokens before applying BPE, *nsubtgt* and *ntgt* correspondingly for its translation, we define two bilingual ratios:

$$uratio = \frac{\max(nsubsrc, nsubtgt)}{\min(nsubsrc, nsubtgt)} \tag{6.6}$$

$$oratio = \frac{\max(nsrc, ntgt)}{\min(nsrc, ntgt)} \tag{6.7}$$

Thus, we have introduced another four ratios in addition to the original length ratio *oratio*. The reason why we clean the training set with ratios related to sub-word units is that rare words in noisy sentence pairs are likely to be segmented into many sub-word units, especially for URLs, which will significantly increase those ratios. Filtering with sub-word level ratios and ratios between token-level and sub-word level may help address these cases.

We provide tools to calculate the above ratios on the validation set, which are good and safe choices for data cleaning.

6.2.6 Additional Tools

Averaging Models. The training of the Transformer periodically saves checkpoints (specifically, for every $1.5k$ training steps). Vaswani et al. (2017) average parameters of several checkpoints and evaluate the performance of the averaged model, which normally leads to more stable and better performance than evaluating the last model. We support this function which loads parameters of given checkpoints and saves the averaged parameters into a new model file.

Ranking. A ranking tool is provided to rank data sets with a pre-trained model, where either per-token loss or the loss of a sentence pair can be measured efficiently. This tool can be employed for data cleaning, data selection for domain adaptation or in the evaluation of linguistic phenomena, e.g., the subject-verb agreement analysis in Chapter 3.

Web Server. We provide a simple translation web server with REST API support in addition to the translation scripts, which we think may be helpful for integrating trained models into the other MT-based applications and platforms. It can also run as a demo to provide a friendly User Interface (UI) when performing human evaluation of some particularly designed examples.

Conversion to C Libraries. We implement a conversion tool based on Cython [5] which can convert Python implementations of core modules and functions into C code and compile them into loadable C libraries. This may bring about a slight acceleration and make it easier to integrate our toolkit into MT services or the other applications depending on MT.

Forbidden Indexes for the Shared Vocabulary. In practical application scenarios, there might be some tokens that only appear on the source side when a shared vocabulary is adopted, and these tokens which never appear in the target side training data will still get a small smoothing probability in the loss function. We provide a tool to extract those indexes and save them into a file that can be loaded to prevent the effects of those tokens on the decoder classifier bias and the label smoothing loss.

6.3 Design

In this section, we present the design of our implementation, i.e., the way we organize the code.

Scripts. We provide scripts for processing training and test data under "scripts/". The training data processing script utilizes several implemented tools to 1) sort the training set and the development set. As a result, sentences with similar lengths will be gathered together, and the number of padding tokens can be reduced. 2) collect vocabularies given the training data. 3) convert parallel sentences to tensors. The test script processes similarly to the training script as regards the first three steps except that it handles only monolingual data and uses the vocabulary produced by the training step rather than rebuilding a new one. After these procedures, it then: 4) uses the translation implementation to translate. 5) restores the order of the sorted test set and corresponding translations. 6) merges sub-word units into tokens.

[5] https://cython.org/

Basic Modules. We provide our implementations of basic modules under "modules/", including the multi-head attention network, the positional embedding, the position-wise feed-forward network, the average attention network, RNNs, etc.

Loss. We implement the label smoothing loss (Szegedy et al., 2017) which minimizes the Kullback-Leibler divergence instead of the cross-entropy under "loss/".

Learning Rate Scheduler. We implement the learning rate of the Transformer (Vaswani et al., 2017) in Equation 6.2 in "lrsch.py".

Parallelization. Our multi-GPU parallelization model which avoids the redundant communication between GPUs described in Section 6.2.2 is implemented under "parallel/".

Support Functions. We implement basic functions for data processing, training and decoding such as conversion from texts to tensors, HDF5 serialization, parameter initialization approaches, freezing/unfreezing parameters of models, padding list of tensors to the same size on the assigned dimension under "utils/".

Transformer and its Variants. We gather the implementation of the Transformer model and its variants, including transparent attention, RNMT decoder, etc., under "transformer/".

Optimizers. The implementation of optional optimizers (RAdam, Lookahead) can be loaded from "optm/".

Tools. All tools to support data processing, averaging checkpoints, etc., are implemented under "tools/".

6.4 Performance

To compare with Vaswani et al. (2017), we tested our Neutron implementation on the WMT 14 English to German news translation task following Vaswani et al. (2017) in this Chapter. The performance on some other datasets can be found in the other chapters (Chapter 3, 4 and 5).

To address the unknown word issue, we applied joint Byte-Pair Encoding (BPE) (Sennrich et al., 2016a) with $32k$ merge operations and 8 as the vocabulary threshold for the BPE to reduce the vocabulary size (Sennrich and Zhang, 2019).

We only kept sentences with a maximum of 256 sub-word tokens for training. The training set was randomly shuffled in every training epoch. The concatenation of newstest 2012 and newstest 2013 was used for validation and newstest 2014 as the test set.

The number of warmup steps was set to $8k$, [6] and each training batch contained at least $25k$ target tokens. The large batch size was achieved through gradient accumulation of small batches. We trained the model on 2 GTX 1080 Ti GPUs, and performed decoding on 1 of them.

We employed the pre-norm computation order (Vaswani et al., 2018). We used a dropout of 0.1. We used the Transformer Base setting (Vaswani et al., 2017) with embedding dimension and hidden dimension of the position-wise feed-forward neural network 512 and $2,048$ respectively, and the model was trained for $100k$ training steps. We employed a label smoothing (Szegedy et al., 2016) value of 0.1. We used the Adam optimizer (Kingma and Ba, 2015) with 0.9, 0.98 and 10^{-9} as β_1, β_2 and ϵ. We followed Vaswani et al. (2017) for the other settings.

We used a beam size of 4 without length penalty for decoding, and evaluated tokenized case-sensitive BLEU [7] with the averaged model of the last 5 checkpoints saved with an interval of $1,500$ training steps (Vaswani et al., 2017). Results are shown in Table 6.1.

Table 6.1 shows that our Neutron implementation of the Transformer surpasses Vaswani et al. (2017) in terms of BLEU while being extremely fast in both training and decoding.

[6] `https://github.com/tensorflow/tensor2tensor/blob/v1.15.4/tensor2tensor/models/transformer.py#L1818`.

[7] `https://github.com/moses-smt/mosesdecoder/blob/master/scripts/generic/multi-bleu.perl`.

	BLEU		Speed	
	En-De	En-Fr	Training	Decoding
Vaswani et al. (2017)	27.3	38.1		
Neutron	27.38	39.34	23213.65	150.15

TABLE 6.1: Performance and speed. Training speed and decoding speed are measured on the En-De task by the number of target tokens per second and the number of sentences per second.

Our implementation of many other approaches also performs competitively on publicly available datasets.

6.5 Related Work

6.5.1 Baseline Models

Sutskever et al. (2014); Bahdanau et al. (2014); Luong et al. (2015); Gehring et al. (2017); Vaswani et al. (2017) and many other researchers proposed various kinds of NMT models, shifting the MT technology from RNN-based approaches to CNN-based models, and further to Transformer-based methods.

Sutskever et al. (2014) propose to employ two LSTMs as encoder and decoder respectively for sequence-to-sequence MT, the encoder encodes the source to a vector, and the decoder auto-regressively generates the corresponding translation in a token-by-token manner. Their simple approach solely based on neural models performs comparably to previous PBSMT systems carefully tuned with substantial engineering effort.

Bahdanau et al. (2014) integrate the attention mechanism into the NMT decoder to jointly learn to translate and align. The attention mechanism attends the source encoding in every decoding step, which brings information from the source side and improves translation quality, especially for long sentences, and alleviates the information loss issue of representing the information of the source sentence with only a fixed-dimension vector.

Gehring et al. (2017) propose to use CNNs that evolve token representations independently rather than RNNs which compute in a token-by-token manner for NMT. Their approach improves parallelization on GPUs.

Vaswani et al. (2017) propose the Transformer based on the multi-head attention mechanism which is able to model over the whole sequence rather than contexts in a fixed window with CNNs for NMT while keeping the advantage of parallelization. The Transformer establishes the new state-of-the-art.

6.5.2 Open-Source Toolkits

Often coming from research work, there are now many open-source implementations, including:

fairseq. fairseq (Ott et al., 2019) is an open-source sequence modeling toolkit that allows researchers and developers to train custom models for translation, summarization, language modeling, and other text generation tasks. The toolkit supports distributed training across multiple GPUs and machines and fast mixed-precision training and inference on modern GPUs.

OpenNMT. Klein et al. (2017) develop OpenNMT. The toolkit prioritizes efficiency, modularity, and extensibility with the goal of supporting NMT research into model architectures, feature representations, and source modalities, while maintaining competitive performance and reasonable training requirements. OpenNMT consists of modeling and translation support, as well as detailed pedagogical documentation about the underlying techniques.

Tensor2Tensor. Tensor2tensor (Vaswani et al., 2018) is the official implementation of the Transformer (Vaswani et al., 2017). It is a library based on TensorFlow (Abadi et al., 2016) for deep learning models that is well-suited for neural machine translation.

Sockeye. Hieber et al. (2017) present Sockeye. Sockeye is a production-ready framework for training and applying models as well as an experimental platform for researchers. Written in Python and built on MXNet (Chen et al., 2015), the toolkit offers scalable training and inference for the three most prominent encoder-decoder architectures: attentional recurrent neural networks, self-attentional Transformers, and fully convolutional

networks. Sockeye also supports a wide range of optimizers, normalization and regularization techniques, and inference improvements from the current NMT literature.

Marian. Marian (Junczys-Dowmunt et al., 2018) is an efficient and self-contained NMT framework with an integrated automatic differentiation engine based on dynamic computation graphs. It is written entirely in C++, and can achieve high training and translation speed.

THUMT. THUMT (Zhang et al., 2017a) implements the standard attention-based encoder-decoder framework and supports three training criteria: maximum likelihood estimation, minimum risk training, and semi-supervised training. It features a visualization tool for displaying the relevance between hidden states in neural networks and contextual words, which helps to analyze the internal workings of NMT.

Lingvo. Lingvo (Shen et al., 2019) is a Tensorflow framework (Abadi et al., 2016) offering a complete solution for collaborative deep learning research, with a particular focus towards sequence-to-sequence models. Models are composed of modular building blocks that are flexible and easily extensible, and experiment configurations are centralized and highly customizable. Lingvo supports distributed training and quantized inference directly within the framework, and it contains existing implementations of a large number of utilities, helper functions, and the newest research ideas.

These implementations play a vital role in both NMT research and applications. They also provide valuable starting points for us to implement our NMT models. This granted, we suggest that our Neutron implementation has its own specialization and focus:

1. It mainly concentrates on cutting-edge research and provides implementations of many approaches that are not included in other libraries.

2. It is very carefully optimized to run efficiently (as shown in Table 6.1) on cheap hardware. When using GPUs, Neutron utilizes at most n CPU cores, where n is the number of GPUs used, takes less than 4GB memory, and can effectively train on old GPUs with even no more than 8GB GPU memory.

6.6 Conclusion

In this chapter, we introduce our Neutron implementation (Xu and Liu, 2019) of the Transformer, including its supported features, advantages, design and performance, etc. with a particular focus on the Transformer and its recent state-of-the-art variants for NMT, implemented as Neutron based on the PyTorch.

Our NMT implementation is able to obtain competitive performance as a baseline and supports our research in this thesis.

Chapter 7

Declaration of Contribution

I hereby declare that for each of the three main papers on which the thesis is build:

- **Hongfei Xu**, Josef van Genabith, Deyi Xiong, Qiuhui Liu, and Jingyi Zhang. Learning Source Phrase Representations for Neural Machine Translation. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 386–396, Online, July 2020. Association for Computational Linguistics.: I am the first author of the paper, I developed the basic idea, I developed the implementation, I carried out the experiments and evaluations reported in the paper, I wrote each of the main drafts of the paper, I discussed and refined the ideas and the writeup of the papers with my co-authors.

- **Hongfei Xu**, Qiuhui Liu, Josef van Genabith, Deyi Xiong, and Jingyi Zhang. Lipschitz Constrained Parameter Initialization for Deep Transformers. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 397–402, Online, July 2020. Association for Computational Linguistics.: I am the first author of the paper, I developed the basic idea, I developed the implementation, I carried out the experiments and evaluations reported in the paper, I wrote each of the main drafts of the paper, I discussed and refined the ideas and the writeup of the papers with my co-authors.

- **Hongfei Xu**, Josef van Genabith, Deyi Xiong, and Qiuhui Liu. Dynamically Adjusting Transformer Batch Size by Monitoring Gradient Direction Change. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 3519–3524, Online, July 2020. Association for Computational Linguistics.: I am the first author of the paper, I developed the basic idea, I developed the implementation, I carried out the experiments and evaluations reported in the paper, I wrote each of the main drafts of the paper, I discussed and refined the ideas and the writeup of the papers with my co-authors.

Chapter 8

Conclusion and Future Work

8.1 Research Contributions and Questions Answered

In this thesis, we have posed seven research questions (RQ1 through RQ7, listed below) and proposed approaches to address each one of them based on our Neutron implementation of the Transformer (described in Chapter 6).

RQ1: *How to improve the ability of the Transformer in long-distance relation capturing?*

In Chapter 3, we propose to additionally model NMT at phrase level to help the Transformer capture long-distance relationships, given that modeling phrases instead of words has significantly improved the Statistical Machine Translation (SMT) approach through the use of larger translation blocks ("phrases") and with this its reordering ability. In our experiments, we obtain significant improvements on the WMT 14 English-German and English-French tasks on top of the strong Transformer baseline, which shows the effectiveness of our approach. Our approach helps Transformer Base models perform at the level of Transformer Big models on the En-De task, and even significantly better for long sentences, but with substantially fewer parameters and training steps. The fact that phrase representations help even in the Big setting further supports our conjecture that they make a valuable contribution to long-distance relations and scales to large data sets. We also conduct length analysis with our approach, and the results show how our approach

improves long-distance dependency capturing, which supports our conjecture that phrase representation sequences can help the model capture long-distance relations better, given that in translating long sentences, we expect to encounter more long-distance dependencies than translating short sentences. In the linguistically-informed subject-verb agreement analysis on the *Lingeval97* dataset (Sennrich, 2017) following Tang et al. (2018), our approach improves the accuracy of long-distance subject-verb dependencies, especially for cases where there are more than 10 tokens between the verb and the corresponding subject.

Contributions for RQ1:

- To help the Transformer translation model better model long-distance dependencies, we let both encoder layers and decoder layers of the Transformer attend the source phrase representation sequence which is shorter than the token sequence, in addition to the original token representation.
- To the best of our knowledge, our work is the first to model source phrase representations and incorporating them into the Transformer.
- Our approach empirically brings about significant and consistent improvements over the strong Transformer model (both Base and Big settings) on the WMT 14 English-German and English-French news translation tasks, and the results on the subject-verb agreement task show how our approach improves long-distance dependency capturing.

RQ2: *How to avoid the potentially large phrase table while benefiting from phrase representations?*

Since the phrase table is much larger than the word vocabulary, which is not affordable for NMT, and distribution over phrases is much sparser than that over words, which may lead to data sparsity and hurt the performance of NMT, we generate phrase representations based on token representations "on-the-fly" rather than using phrase embeddings directly. Considering that merging several token vectors into one is very likely to incur information loss, we suggest that introducing an importance evaluation mechanism is better than treating tokens equally. Our research reported in Chapter 3 proposes an attentive feature

extraction model and generates phrase representation based on token representations. Specifically, our model first summarizes the representation of a given token sequence with mean or max-over-time pooling, then uses a 2-layer neural network to compute scores of token representations by comparing the extracted feature representation with each token representation, and generates the phrase representation by a weighted combination of token representations with normalized scores.

Contributions for RQ2:

- To address the large phrase table issue, we introduce an attentive phrase representation generation model to value tokens differently according to their importance in the phrase, which is able to highlight the important features in a phrase.

RQ3: *How to learn and utilize phrase representation in the Transformer translation model?*

In Chapter 3, we propose an attentive combination network to incorporate the source phrase representation for each layer into the Transformer translation model to aid it modeling long-distance dependencies. Specifically, we insert the attentive combination layer into each encoder and decoder layer to attend to the source phrase representation sequence before attending to the token-level source representations. We do this under the expectation that attending at phrase level might be easier than at token level, and attention results at phrase level may aid the attention computation at the token level. The phrase representation sequence of each encoder layer is generated by the attentive phrase representation algorithm with the input representation sequence to this layer. For the source phrase representation used by decoder layers, we weighted combine phrase representations generated by all encoder layers for each decoder layer with the transparent attention mechanism.

Contributions for RQ3:

- We integrate the learning of phrase representations into the Transformer in an end-to-end manner with an attentive combination network to let the model additionally condition on the source phrase level, which is differentiable and can be efficiently trained through backpropagation.

RQ4: *Why do Transformers, specifically deep Transformers, have difficulty in converging even with layer normalization and residual connections?*

Chapter 4 first presents both empirical convergence experiments (Table 4.1) comparing the computation order of the official implementation (Vaswani et al., 2018) and that of the published version (Vaswani et al., 2017). Then we perform a theoretical analysis (Table 4.2) of the effect of the interaction between layer normalization and residual connection. Our analysis shows that the convergence issue of deep Transformers is likely due to the fact that layer normalization over residual connections may effectively reduce the impact of residual connections due to subsequent layer normalization, in case the standard deviation of the input to the layer normalization is larger than 1, in order to avoid a potential explosion of combined layer outputs (Chen et al., 2018b), which will shrink the corresponding gradients in backpropagation.

Contributions for RQ4:

- We provide empirical results of two different computation orders of deep Transformers which show the different convergence patterns for the different computation orders between layer normalization and residual connection.
- We theoretically analyze the impact of layer normalization over the residual connection on convergence.

RQ5: *How to prevent layer normalization from shrinking residual connections?*

In Chapter 4, we demonstrate that the goal to constrain the standard deviation of the input to layer normalization can be achieved through proper parameter initialization. Specifically, we propose to initialize the sub-model of deep Transformers before layer normalization under the k-Lipschitz constraint (where $k \leq 1$), which can simply and effectively ensure convergence. Our empirical results with the Lipschitz constrained parameter initialization show that deep Transformers with the original computation order (Vaswani et al., 2017) can converge with significant improvements as long as they are initialized with our parameter initialization approach. It is also worth noting that our parameter initialization approach does not degrade the performance of the 6-layer Transformer, in contrast to previous work (Zhang et al., 2019a).

Contributions for RQ5:

- We propose to initialize deep Transformers under the Lipschitz constraint, and empirically show that the convergence of deep Transformers can be ensured with proper parameter initialization.
- Our parameter initialization approach brings about significant BLEU improvements with up to 24 layers. While ensuring the convergence of deep Transformers, our approach does not degrade the translation quality of the 6-layer Transformer, and the 12-layer Transformer with our approach already achieves performance comparable to the 20-layer Transformer in the previous work (Zhang et al., 2019a) using merged attention decoder layers with a $50k$ batch size.

RQ6: *How to dynamically and automatically find proper and efficient batch sizes during training?*

Gradient accumulation accumulates gradients of small mini-batches as the gradient of a large batch consisting of these small batches. Chapter 5 analyzes how increasing batch size affects gradient direction during gradient accumulation, and proposes to evaluate the stability of gradients with their angle change. We observe that:

1. The gradient direction change is big at the beginning.
2. With gradient accumulation of more and more batches, gradient direction change reduces with increasing batch size.
3. Eventually, gradient direction change will start fluctuating.

We propose to automatically and dynamically determine batch sizes by accumulating gradients of mini-batches, while evaluating the gradient direction change with each new mini-batch, and stop accumulating more mini-batches at just the time when the direction of gradients starts to fluctuate.

In our experiments on the WMT 14 English-German and English-French news translation tasks, our approach significantly outperforms the baseline with a fixed 25 thousand

batch size by +0.73 and +0.82 BLEU respectively while being more efficient than the 50 thousand batch size setting.

Contributions for RQ6:

- We observe the effects on gradients with increasing batch size, and find that a large batch size stabilizes the direction of gradients.
- We propose to automatically determine dynamic batch sizes in training by monitoring the gradient direction change while accumulating gradients of small batches.
- In machine translation experiments, our approach improves the training efficiency and the performance of the Transformer model.

RQ7: *How to efficiently monitor gradient direction change?*

The Transformer contains a large number of parameters, and the costs for computing the cosine similarity of corresponding gradients in each accumulation step are relatively high. In Chapter 5, we propose to divide model parameters into groups, and monitor gradient direction change only on a selected group in each optimization step rather than estimating the gradient direction change of all parameters. To select the parameter group sensitive to the batch size frequently, we record the minimum and maximum angle change during gradient accumulation, and normalize the angle reductions of parameter groups after adding Gumble noise as their sample probabilities.

Contributions for RQ7:

- We propose to dynamically select those gradients of parameters/layers which are sensitive to the batch size.
- The sampling approach ensures the efficiency of our approach for monitoring gradient direction change, especially for large models.

8.2 Future Work

Based on the work presented in the thesis, below we outline several research directions which can be explored in the future.

NMT Modeling with Phrase Representations. Using phrases in NMT models involves 3 parts: phrase segmentation, phrase representation learning and integrating phrase representations into NMT models. Future studies may research approaches for each part. For example, we could try to find more efficient approaches to learn and to integrate phrase representations in NMT models, as the cost of additionally introducing phrase representations to the Transformer of our current approach (described in Chapter 3) which focuses on verifying the impacts of phrase representations on long-distance relation capturing is relatively high. Performing phrase-level attention with some redundant heads (Voita et al., 2019d) in the multi-head attention network seems to be a good choice (Hao et al., 2019a), but how to determine the number of heads kept for token representations remains an unsolved problem. Is there any way to let the model learn the number of heads for phrase-level or even multi-level attention?

Thus, in terms of phrase representations for NMT research, there are several options to be explored, such as:

1. To explore better segmentation approaches of phrases for NMT. We examine both n-gram phrases and linguistically motivated phrases from a parser in Chapter 3, with our attentive phrase representation generation approach. The performance gap between these two kinds of phrase segmentation approaches in terms of BLEU scores in our experiments is small. However, is it possible to design an efficient phrase segmentation approach (without involving a parser) which produces linguistically reasonable phrases that lead to comparable performance with only a simple phrase representation learning approach (without the use of attention or gate mechanisms)? For example, segmenting phrases based on bi-gram or n-gram co-occurrences or using function words to chunk sentences into more linguistically motivated phrases.

2. To design more powerful and efficient algorithms for the generation of phrase representations based on token representations. Our attentive phrase representation generation approach in Chapter 3 contains 3 steps: 1) summarizing all token representations into a vector. 2) comparing each token representation with the summarized representation with a 2-layer neural network. 3) weighted combined token representations with normalized scores. Future work may study efficient approaches

for phrase representation learning to simplify the procedure. Is it possible to generate the phrase representation of corresponding token representations in only one step?

3. To find a better way for integrating phrase representation learning into NMT. Our approach (in Chapter 3) inserts the attentive combination network into each encoder layer and decoder layer to attend the source phrase representation sequence before attending the source token representation sequence. Is it necessary to perform phrase-level attention in all layers, especially for deep Transformers which have more layers? Are there any more efficient approaches to integrate the phrase representation learning into NMT models?

4. To explore the use of phrases on the target side in addition to the source. We only investigate the effects of source phrase representations in Chapter 3. However, long-distance dependency learning ability is likely to be also important for the target language model, and future work may research the effects of target phrase representations in addition to source phrase representations.

5. To integrate our approach into the other models for tasks in which the capability of capturing long-distance relations is crucial, e.g., document-level MT.

Future work in this direction will result in more efficient architectures with improved performance in both translation and long-distance dependency learning.

Efficient Deep Transformers. In Chapter 4, we analyze and study the convergence issue of deep Transformers. However, to the best of our knowledge, almost all related research (Bapna et al., 2018; Wang et al., 2019c; Wu et al., 2019c; Zhang et al., 2019a) including ours is following the motivation of residual connections to highlight outputs of shallow layers in the forward propagation to ease the flow of gradients during backpropagation and further ensure convergence of the models.

However, the motivation of using deep neural networks is to increase the complexity of the model function. But the residual connection, which adds the input to the output of the layer, somehow hampers the non-linearity, i.e., the complexity of the model function. As a result, the improvements in performances are smaller and smaller with increasing

depth, and deep models seem to have difficulty in using parameters as efficiently as their shallow counterparts. Future research may study:

1. Training deep models without residual connection. Is it possible to ensure the convergence of deep models after residual connections are removed? E.g., via parameter initialization, using some other non-linear activation functions, non-linear cross-layer connection, etc.

2. Why do deep models without residual connection suffer from the convergence issue? He et al. (2016) suggest that the non-linear activation function makes the layer without the residual connection have difficulty in learning the identity function, and thus the model without residual connections suffers from a severer convergence problem than the model with residual connections. However, we suggest that the identity function is not what we want the layer to learn, and what is the real problem behind the training issue of non-linear layers is worth exploring.

Parameter Initialization. Chapter 4 studies the impact of parameter initialization under the Lipschitz constraint on the convergence of deep Transformers. However, convergence is not the only thing that parameter initialization affects. It can also affect the after-training performance of models.

For example, the Lottery Ticket (LT) hypothesis (Frankle and Carbin, 2019; Frankle et al., 2019; Zhou et al., 2019; Dettmers and Zettlemoyer, 2019) suggests that there is a sparse sub-network in a dense network that outperforms the fully connected original network with a significant reduction in parameters and corresponding computation in Computer Vision (CV) tasks, and its connections have initial weights that make training particularly effective.

Future work may carry out research on improved parameter initialization of the Transformer which can accelerate its convergence, and result in better performance after training, or reduce the number of parameters required. More specifically:

1. To validate the LT hypothesis in the MT task with NMT models. Specifically with the Transformer translation model, which parameters can be heavily pruned and why?

2. To analyze the reason w.r.t. parameter initialization behind the LT hypothesis. Why can the random parameter initialization of the sub-network take the job of the whole model with comparative performance? Why is the initialization good?

3. To design improved parameter initialization approaches that can lead to improved after-training performance of the full model based on findings from the LT hypothesis analysis.

Hyperparameter Selection/Neural Architecture Search of Translation Models.

The dynamic batch size in Chapter 4 studies how to select the proper batch size during training based on tracking gradient direction change. However, the selection of the other hyperparameters, such as hidden units, encoder/decoder depth, or even more aggressively, the architecture of neural machine translation models has not been studied. Future work may include:

1. Exploration of pruning-based (Gomez et al., 2019) or searching-based approaches (Radosavovic et al., 2020) to find proper hidden dimensions and the corresponding depth for MT models.

2. Exploration of Neural Architecture Search (NAS) approaches (So et al., 2019; Gaier and Ha, 2019; Luo et al., 2018; Wong et al., 2018), especially Efficient Neural Architecture Search (ENAS) approaches (Pham et al., 2018; Liu et al., 2019b; Xie et al., 2019; Cai et al., 2019; Bender et al., 2018; Pasunuru and Bansal, 2019; Dong and Yang, 2019b; Wu et al., 2019a; Fedorov et al., 2019; Nayman et al., 2019; Dong and Yang, 2019a; Peng et al., 2019; Hu et al., 2019; Chen et al., 2019c; Geifman and El-Yaniv, 2019) for NMT.

Bibliography

Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, Manjunath Kudlur, Josh Levenberg, Rajat Monga, Sherry Moore, Derek G. Murray, Benoit Steiner, Paul Tucker, Vijay Vasudevan, Pete Warden, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. Tensorflow: A system for large-scale machine learning. In *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, pages 265–283, Savannah, GA, November 2016. USENIX Association. ISBN 978-1-931971-33-1. URL `https://www.usenix.org/conference/osdi16/technical-sessions/presentation/abadi`.

Mostafa Abdou, Vladan Glončák, and Ondřej Bojar. Variable mini-batch sizing and pre-trained embeddings. In *Proceedings of the Second Conference on Machine Translation*, pages 680–686, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/W17-4780. URL `https://www.aclweb.org/anthology/W17-4780`.

Philip Arthur, Graham Neubig, and Satoshi Nakamura. Incorporating discrete translation lexicons into neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1557–1567, Austin, Texas, November 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1162. URL `https://www.aclweb.org/anthology/D16-1162`.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. URL `https://arxiv.org/abs/1607.06450`.

Parnia Bahar, Christopher Brix, and Hermann Ney. Towards two-dimensional sequence to sequence model in neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3009–3015, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1335. URL `https://www.aclweb.org/anthology/D18-1335`.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014. URL `https://arxiv.org/abs/1409.0473`.

Shaojie Bai, J. Zico Kolter, and Vladlen Koltun. Deep equilibrium models. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 690–701. Curran Associates, Inc., 2019. URL `http://papers.nips.cc/paper/8358-deep-equilibrium-models.pdf`.

Lukas Balles, Javier Romero, and Philipp Hennig. Coupling adaptive batch sizes with learning rates. In Gal Elidan, Kristian Kersting, and Alexander T. Ihler, editors, *Proceedings of the Thirty-Third Conference on Uncertainty in Artificial Intelligence, UAI 2017, Sydney, Australia, August 11-15, 2017*. AUAI Press, 2017. URL `http://auai.org/uai2017/proceedings/papers/141.pdf`.

Satanjeev Banerjee and Alon Lavie. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic*

and Extrinsic Evaluation Measures for Machine Translation and/or Summarization, pages 65–72, Ann Arbor, Michigan, June 2005. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/W05-0909.

Ankur Bapna, Mia Chen, Orhan Firat, Yuan Cao, and Yonghui Wu. Training deeper neural machine translation models with transparent attention. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3028–3033. Association for Computational Linguistics, 2018. URL http://aclweb.org/anthology/D18-1338.

Rachel Bawden, Rico Sennrich, Alexandra Birch, and Barry Haddow. Evaluating discourse phenomena in neural machine translation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1304–1313, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1118. URL https://www.aclweb.org/anthology/N18-1118.

Yonatan Belinkov and Yonatan Bisk. Synthetic and natural noise both break neural machine translation. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=BJ8vJebC-.

Gabriel Bender, Pieter-Jan Kindermans, Barret Zoph, Vijay Vasudevan, and Quoc Le. Understanding and simplifying one-shot architecture search. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 550–559, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018. PMLR. URL http://proceedings.mlr.press/v80/bender18a.html.

Arianna Bisazza and Clara Tump. The lazy encoder: A fine-grained analysis of the role of morphology in neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2871–2876, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1313. URL https://www.aclweb.org/anthology/D18-1313.

Denny Britz, Anna Goldie, Minh-Thang Luong, and Quoc Le. Massive exploration of neural machine translation architectures. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1442–1451, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1151. URL https://www.aclweb.org/anthology/D17-1151.

Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311, 1993. URL https://www.aclweb.org/anthology/J93-2003.

Han Cai, Ligeng Zhu, and Song Han. ProxylessNAS: Direct neural architecture search on target task and hardware. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=HylVB3AqYm.

Qian Cao and Deyi Xiong. Encoding gated translation memory into neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3042–3047, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1340. URL https://www.aclweb.org/anthology/D18-1340.

Huadong Chen, Shujian Huang, David Chiang, and Jiajun Chen. Improved neural machine translation with a syntax-aware encoder and decoder. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1936–1945, Vancouver, Canada, July 2017a. Association for Computational Linguistics. doi: 10.18653/v1/P17-1177. URL https://www.aclweb.org/anthology/P17-1177.

Huadong Chen, Shujian Huang, David Chiang, Xinyu Dai, and Jiajun Chen. Combining character and word information in neural machine translation using a multi-level attention. In

Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), pages 1284–1293, New Orleans, Louisiana, June 2018a. Association for Computational Linguistics. doi: 10.18653/v1/N18-1116. URL https://www.aclweb.org/anthology/N18-1116.

Kehai Chen, Rui Wang, Masao Utiyama, Lemao Liu, Akihiro Tamura, Eiichiro Sumita, and Tiejun Zhao. Neural machine translation with source dependency representation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2846–2852, Copenhagen, Denmark, September 2017b. Association for Computational Linguistics. doi: 10.18653/v1/D17-1304. URL https://www.aclweb.org/anthology/D17-1304.

Kehai Chen, Rui Wang, Masao Utiyama, and Eiichiro Sumita. Neural machine translation with reordering embeddings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1787–1799, Florence, Italy, July 2019a. Association for Computational Linguistics. doi: 10.18653/v1/P19-1174. URL https://www.aclweb.org/anthology/P19-1174.

Kehai Chen, Rui Wang, Masao Utiyama, and Eiichiro Sumita. Recurrent positional embedding for neural machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1361–1367, Hong Kong, China, November 2019b. Association for Computational Linguistics. doi: 10.18653/v1/D19-1139. URL https://www.aclweb.org/anthology/D19-1139.

Mia Xu Chen, Orhan Firat, Ankur Bapna, Melvin Johnson, Wolfgang Macherey, George Foster, Llion Jones, Mike Schuster, Noam Shazeer, Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Lukasz Kaiser, Zhifeng Chen, Yonghui Wu, and Macduff Hughes. The best of both worlds: Combining recent advances in neural machine translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 76–86, Melbourne, Australia, July 2018b. Association for Computational Linguistics. doi: 10.18653/v1/P18-1008. URL https://www.aclweb.org/anthology/P18-1008.

Tianqi Chen, Mu Li, Yutian Li, Min Lin, Naiyan Wang, Minjie Wang, Tianjun Xiao, Bing Xu, Chiyuan Zhang, and Zheng Zhang. Mxnet: A flexible and efficient machine learning library for heterogeneous distributed systems. *CoRR*, abs/1512.01274, 2015. URL http://arxiv.org/abs/1512.01274.

Yukang Chen, Tong Yang, Xiangyu Zhang, GAOFENG MENG, Xinyu Xiao, and Jian Sun. Detnas: Backbone search for object detection. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 6642–6652. Curran Associates, Inc., 2019c. URL http://papers.nips.cc/paper/8890-detnas-backbone-search-for-object-detection.pdf.

Yong Cheng, Zhaopeng Tu, Fandong Meng, Junjie Zhai, and Yang Liu. Towards robust neural machine translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1756–1766. Association for Computational Linguistics, 2018. URL http://aclweb.org/anthology/P18-1163.

Yong Cheng, Lu Jiang, and Wolfgang Macherey. Robust neural machine translation with doubly adversarial inputs. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4324–4333, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1425. URL https://www.aclweb.org/anthology/P19-1425.

David Chiang. Hierarchical phrase-based translation. *Computational Linguistics*, 33(2):201–228, 2007. doi: 10.1162/coli.2007.33.2.201. URL https://www.aclweb.org/anthology/J07-2003.

David Chiang. Learning to translate with source and target syntax. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1443–1452, Uppsala, Sweden, July 2010. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/P10-1146.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar, October 2014. Association for Computational Linguistics. doi: 10.3115/v1/D14-1179. URL https://www.aclweb.org/anthology/D14-1179.

Reuben Cohn-Gordon and Noah Goodman. Lost in machine translation: A method to reduce meaning loss. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 437–441, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1042. URL https://www.aclweb.org/anthology/N19-1042.

Gonçalo M. Correia, Vlad Niculae, and André F. T. Martins. Adaptively sparse transformers. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2174–2184, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1223. URL https://www.aclweb.org/anthology/D19-1223.

Anna Currey and Kenneth Heafield. Multi-source syntactic neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2961–2966, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1327. URL https://www.aclweb.org/anthology/D18-1327.

Leonard Dahlmann, Evgeny Matusov, Pavel Petrushkov, and Shahram Khadivi. Neural machine translation leveraging phrase-based models in a hybrid search. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1411–1420, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1148. URL https://www.aclweb.org/anthology/D17-1148.

Mostafa Dehghani, Stephan Gouws, Oriol Vinyals, Jakob Uszkoreit, and Lukasz Kaiser. Universal transformers. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=HyzdRiR9Y7.

Tim Dettmers and Luke Zettlemoyer. Sparse networks from scratch: Faster training without losing performance. *CoRR*, abs/1907.04840, 2019. URL http://arxiv.org/abs/1907.04840.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://www.aclweb.org/anthology/N19-1423.

Tobias Domhan. How much attention do you need? a granular analysis of neural machine translation architectures. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1799–1808. Association for Computational Linguistics, 2018. URL http://aclweb.org/anthology/P18-1167.

Xuanyi Dong and Yi Yang. Network pruning via transformable architecture search. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 760–771. Curran Associates, Inc., 2019a. URL http://papers.nips.cc/paper/8364-network-pruning-via-transformable-architecture-search.pdf.

Xuanyi Dong and Yi Yang. Searching for a robust neural architecture in four gpu hours. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019b. URL http://openaccess.thecvf.com/content_CVPR_2019/papers/Dong_Searching_for_a_Robust_Neural_Architecture_in_Four_GPU_Hours_CVPR_2019_paper.pdf.

B. Dorr, E. Hovy, and L. Levin. Machine translation: Interlingual methods. In Keith Brown, editor, *Encyclopedia of Language & Linguistics (Second Edition)*, pages 383–394. Elsevier, Oxford, second edition, 2006. ISBN 978-0-08-044854-1. doi: https://doi.org/10.1016/B0-08-044854-2/00939-1. URL https://www.sciencedirect.com/science/article/pii/B0080448542009391.

Zi-Yi Dou, Zhaopeng Tu, Xing Wang, Shuming Shi, and Tong Zhang. Exploiting deep representations for neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4253–4262, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1457. URL https://www.aclweb.org/anthology/D18-1457.

Zi-Yi Dou, Zhaopeng Tu, Xing Wang, Longyue Wang, Shuming Shi, and Tong Zhang. Dynamic layer aggregation for neural machine translation with routing-by-agreement. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence*, pages 86–93, 2019. URL https://aaai.org/ojs/index.php/AAAI/article/view/3772/3650.

John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12:2121–2159, July 2011. ISSN 1532-4435. URL http://dl.acm.org/citation.cfm?id=1953048.2021068.

Chris Dyer, Victor Chahuneau, and Noah A. Smith. A simple, fast, and effective reparameterization of IBM model 2. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 644–648, Atlanta, Georgia, June 2013. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/N13-1073.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. Understanding back-translation at scale. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 489–500. Association for Computational Linguistics, 2018. URL http://aclweb.org/anthology/D18-1045.

Maha Elbayad, Jiatao Gu, Edouard Grave, and Michael Auli. Depth-adaptive transformer. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=SJg7KhVKPH.

Ronen Eldan and Ohad Shamir. The power of depth for feedforward neural networks. In Vitaly Feldman, Alexander Rakhlin, and Ohad Shamir, editors, *29th Annual Conference on Learning Theory*, volume 49 of *Proceedings of Machine Learning Research*, pages 907–940, Columbia University, New York, New York, USA, 23–26 Jun 2016. PMLR. URL http://proceedings.mlr.press/v49/eldan16.html.

Akiko Eriguchi, Yoshimasa Tsuruoka, and Kyunghyun Cho. Learning to parse and translate improves neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 72–78, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-2012. URL https://www.aclweb.org/anthology/P17-2012.

Marzieh Fadaee and Christof Monz. Back-translation sampling by targeting difficult words in neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 436–446, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1040. URL https://www.aclweb.org/anthology/D18-1040.

Igor Fedorov, Ryan P Adams, Matthew Mattina, and Paul Whatmough. Sparse: Sparse architecture search for cnns on resource-constrained microcontrollers. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 4977–4989. Curran Associates, Inc., 2019. URL http://papers.nips.cc/paper/8743-sparse-sparse-architecture-search-for-cnns-on-resource-constrained-microcontrollers.pdf.

Yang Feng, Shiyue Zhang, Andi Zhang, Dong Wang, and Andrew Abel. Memory-augmented neural machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1390–1399, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1146. URL https://www.aclweb.org/anthology/D17-1146.

Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=rJl-b3RcF7.

Jonathan Frankle, Gintare Karolina Dziugaite, Daniel M. Roy, and Michael Carbin. The lottery ticket hypothesis at scale. *CoRR*, abs/1903.01611, 2019. URL http://arxiv.org/abs/1903.01611.

Adam Gaier and David Ha. Weight agnostic neural networks. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 5364–5378. Curran Associates, Inc., 2019. URL http://papers.nips.cc/paper/8777-weight-agnostic-neural-networks.pdf.

Michel Galley and Christopher D. Manning. A simple and effective hierarchical phrase reordering model. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 848–856, Honolulu, Hawaii, October 2008. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/D08-1089.

Michel Galley, Jonathan Graehl, Kevin Knight, Daniel Marcu, Steve DeNeefe, Wei Wang, and Ignacio Thayer. Scalable inference and training of context-rich syntactic translation models. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 961–968, Sydney, Australia, July 2006. Association for Computational Linguistics. doi: 10.3115/1220175.1220296. URL https://www.aclweb.org/anthology/P06-1121.

Sarthak Garg, Stephan Peitz, Udhyakumar Nallasamy, and Matthias Paulik. Jointly learning to align and translate with transformer models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4453–4462, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1453. URL https://www.aclweb.org/anthology/D19-1453.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. Convolutional sequence to sequence learning. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1243–1252, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR. URL http://proceedings.mlr.press/v70/gehring17a.html.

Yonatan Geifman and Ran El-Yaniv. Deep active learning with a neural architecture search. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 5976–5986. Curran Associates, Inc., 2019. URL http://papers.nips.cc/paper/8831-deep-active-learning-with-a-neural-architecture-search.pdf.

Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In Yee Whye Teh and Mike Titterington, editors, *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 249–256, Chia Laguna Resort, Sardinia, Italy, 13–15 May 2010. PMLR. URL http://proceedings.mlr.press/v9/glorot10a.html.

Aidan N. Gomez, Ivan Zhang, Kevin Swersky, Yarin Gal, and Geoffrey E. Hinton. Learning sparse networks using targeted dropout. *CoRR*, abs/1905.13678, 2019. URL http://arxiv.org/abs/1905.13678.

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor O.K. Li. Incorporating copying mechanism in sequence-to-sequence learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1631–1640, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-1154. URL https://www.aclweb.org/anthology/P16-1154.

Jiatao Gu, James Bradbury, Caiming Xiong, Victor O.K. Li, and Richard Socher. Non-autoregressive neural machine translation. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=B1l8BtlCb.

Jiatao Gu, Changhan Wang, and Junbo Zhao. Levenshtein transformer. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 11181–11191. Curran Associates, Inc., 2019. URL http://papers.nips.cc/paper/9297-levenshtein-transformer.pdf.

Emil Julius Gumbel. Statistical theory of extreme values and some practical applications. *NBS Applied Mathematics Series*, 33, 1954.

Junliang Guo, Xu Tan, Di He, Tao Qin, Linli Xu, and Tie-Yan Liu. Non-autoregressive neural machine translation with enhanced decoder input. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 3723–3730. AAAI Press, 2019a. doi: 10.1609/aaai.v33i01.33013723. URL https://doi.org/10.1609/aaai.v33i01.33013723.

Qipeng Guo, Xipeng Qiu, Pengfei Liu, Yunfan Shao, Xiangyang Xue, and Zheng Zhang. Star-transformer. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1315–1325, Minneapolis, Minnesota, June 2019b. Association for Computational Linguistics. doi: 10.18653/v1/N19-1133. URL https://www.aclweb.org/anthology/N19-1133.

Zhijiang Guo, Yan Zhang, Zhiyang Teng, and Wei Lu. Densely connected graph convolutional networks for graph-to-sequence learning. *Transactions of the Association for Computational Linguistics*, 7:297–312, March 2019c. doi: 10.1162/tacl_a_00269. URL https://www.aclweb.org/anthology/Q19-1019.

Jie Hao, Xing Wang, Shuming Shi, Jinfeng Zhang, and Zhaopeng Tu. Multi-granularity self-attention for neural machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 887–897, Hong Kong, China, November 2019a. Association for Computational Linguistics. doi: 10.18653/v1/D19-1082. URL https://www.aclweb.org/anthology/D19-1082.

Jie Hao, Xing Wang, Shuming Shi, Jinfeng Zhang, and Zhaopeng Tu. Towards better modeling hierarchical structure for self-attention with ordered neurons. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1336–1341, Hong Kong, China, November 2019b. Association for Computational Linguistics. doi: 10.18653/v1/D19-1135. URL https://www.aclweb.org/anthology/D19-1135.

Jie Hao, Xing Wang, Baosong Yang, Longyue Wang, Jinfeng Zhang, and Zhaopeng Tu. Modeling recurrence for transformer. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1198–1207, Minneapolis, Minnesota, June 2019c. Association for Computational Linguistics. doi: 10.18653/v1/N19-1122. URL https://www.aclweb.org/anthology/N19-1122.

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, June 2016. doi: 10.1109/CVPR.2016.90.

Shilin He, Zhaopeng Tu, Xing Wang, Longyue Wang, Michael Lyu, and Shuming Shi. Towards understanding neural machine translation with word importance. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 953–962, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1088. URL https://www.aclweb.org/anthology/D19-1088.

Tianyu He, Xu Tan, Yingce Xia, Di He, Tao Qin, Zhibo Chen, and Tie-Yan Liu. Layer-wise coordination between encoder and decoder for neural machine translation. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 7944–7954. Curran Associates, Inc., 2018. URL http://papers.nips.cc/paper/8019-layer-wise-coordination-between-encoder-and-decoder-for-neural-machine-translation.pdf.

Wei He, Zhongjun He, Hua Wu, and Haifeng Wang. Improved neural machine translation with SMT features. In Dale Schuurmans and Michael P. Wellman, editors, *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA*, pages 151–157. AAAI Press, 2016a. URL http://www.aaai.org/ocs/index.php/AAAI/AAAI16/paper/view/12189.

Wei He, Zhongjun He, Hua Wu, and Haifeng Wang. Improved neural machine translation with SMT features. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, pages 151–157, 2016b. URL https://www.aaai.org/ocs/index.php/AAAI/AAAI16/paper/view/12189/11577.

Georg Heigold, Stalin Varanasi, Günter Neumann, and Josef van Genabith. How robust are character-based word embeddings in tagging and MT against wrod scramlbing or randdm nouse? In *Proceedings of the 13th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Track)*, pages 68–80, Boston, MA, March 2018. Association for Machine Translation in the Americas. URL https://www.aclweb.org/anthology/W18-1807.

Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *CoRR*, abs/1606.08415, 2016. URL http://arxiv.org/abs/1606.08415.

Felix Hieber, Tobias Domhan, Michael Denkowski, David Vilar, Artem Sokolov, Ann Clifton, and Matt Post. Sockeye: A toolkit for neural machine translation. *arXiv preprint arXiv:1712.05690*, 2017. URL https://arxiv.org/abs/1712.05690.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8): 1735–1780, November 1997. ISSN 0899-7667. doi: 10.1162/neco.1997.9.8.1735. URL http://dx.doi.org/10.1162/neco.1997.9.8.1735.

Chris Hokamp and Qun Liu. Lexically constrained decoding for sequence generation using grid beam search. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1535–1546, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1141. URL https://www.aclweb.org/anthology/P17-1141.

Hanzhang Hu, John Langford, Rich Caruana, Saurajit Mukherjee, Eric J Horvitz, and Debadeepta Dey. Efficient forward architecture search. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 10122–10131. Curran Associates, Inc., 2019. URL http://papers.nips.cc/paper/9202-efficient-forward-architecture-search.pdf.

Liang Huang and David Chiang. Forest rescoring: Faster decoding with integrated language models. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 144–151, Prague, Czech Republic, June 2007. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/P07-1019.

W. John Hutchins. Machine translation: A brief history. In E.F.K. KOERNER and R.E. ASHER, editors, *Concise History of the Language Sciences*, pages 431–445. Pergamon, Amsterdam, 1995. ISBN 978-0-08-042580-1. doi: https://doi.org/10.1016/B978-0-08-042580-1.50066-0. URL https://www.sciencedirect.com/science/article/pii/B9780080425801500660.

Sathish Reddy Indurthi, Insoo Chung, and Sangha Kim. Look harder: A neural machine translation model with hard attention. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3037–3043, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1290. URL https://www.aclweb.org/anthology/P19-1290.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. Marian: Fast neural machine translation in C++. In *Proceedings of ACL 2018, System Demonstrations*, pages 116–121, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-4020. URL https://www.aclweb.org/anthology/P18-4020.

Lukasz Kaiser, Aidan N. Gomez, and Francois Chollet. Depthwise separable convolutions for neural machine translation. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=S1jBcueAb.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1412.6980.

Eliyahu Kiperwasser and Miguel Ballesteros. Scheduled multi-task learning: From syntax to translation. *Transactions of the Association for Computational Linguistics*, 6:225–240, 2018. doi: 10.1162/tacl_a_00017. URL https://www.aclweb.org/anthology/Q18-1017.

Nikita Kitaev, Lukasz Kaiser, and Anselm Levskaya. Reformer: The efficient transformer. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=rkgNKkHtvB.

G. Klein, Y. Kim, Y. Deng, J. Senellart, and A. M. Rush. OpenNMT: Open-Source Toolkit for Neural Machine Translation. *ArXiv e-prints*, 2017. URL https://arxiv.org/abs/1701.02810.

Philipp Koehn. Statistical significance tests for machine translation evaluation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, 2004. URL http://aclweb.org/anthology/W04-3250.

Philipp Koehn, Franz J. Och, and Daniel Marcu. Statistical phrase-based translation. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 127–133, 2003. URL https://www.aclweb.org/anthology/N03-1017.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic, June 2007. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/P07-2045.

Shaohui Kuang and Deyi Xiong. Fusing recency into neural machine translation with an inter-sentence gate model. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 607–617, Santa Fe, New Mexico, USA, August 2018. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/C18-1051.

Shaohui Kuang, Junhui Li, António Branco, Weihua Luo, and Deyi Xiong. Attention focusing for neural machine translation by bridging source and target embeddings. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1767–1776. Association for Computational Linguistics, 2018a. URL http://aclweb.org/anthology/P18-1164.

Shaohui Kuang, Deyi Xiong, Weihua Luo, and Guodong Zhou. Modeling coherence for neural machine translation with dynamic and topic caches. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 596–606, Santa Fe, New Mexico, USA, August 2018b. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/C18-1050.

Gaurav Kumar, George Foster, Colin Cherry, and Maxim Krikun. Reinforcement learning based curriculum optimization for neural machine translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2054–2061, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1208. URL https://www.aclweb.org/anthology/N19-1208.

Samuel Läubli, Rico Sennrich, and Martin Volk. Has machine translation achieved human parity? a case for document-level evaluation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4791–4796, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1512. URL https://www.aclweb.org/anthology/D18-1512.

Jason Lee, Elman Mansimov, and Kyunghyun Cho. Deterministic non-autoregressive neural sequence modeling by iterative refinement. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1173–1182, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1149. URL https://www.aclweb.org/anthology/D18-1149.

Jian Li, Zhaopeng Tu, Baosong Yang, Michael R. Lyu, and Tong Zhang. Multi-head attention with disagreement regularization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2897–2903, Brussels, Belgium, October-November 2018a. Association for Computational Linguistics. doi: 10.18653/v1/D18-1317. URL https://www.aclweb.org/anthology/D18-1317.

Jian Li, Baosong Yang, Zi-Yi Dou, Xing Wang, Michael R. Lyu, and Zhaopeng Tu. Information aggregation for multi-head attention with routing-by-agreement. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3566–3575, Minneapolis, Minnesota, June 2019a. Association for Computational Linguistics. doi: 10.18653/v1/N19-1359. URL https://www.aclweb.org/anthology/N19-1359.

Junhui Li, Deyi Xiong, Zhaopeng Tu, Muhua Zhu, Min Zhang, and Guodong Zhou. Modeling source syntax for neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 688–697, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1064. URL https://www.aclweb.org/anthology/P17-1064.

Xintong Li, Lemao Liu, Zhaopeng Tu, Shuming Shi, and Max Meng. Target foresight based attention for neural machine translation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1380–1390, New Orleans, Louisiana,

June 2018b. Association for Computational Linguistics. doi: 10.18653/v1/N18-1125. URL https://www.aclweb.org/anthology/N18-1125.

Xintong Li, Guanlin Li, Lemao Liu, Max Meng, and Shuming Shi. On the word alignment from neural machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1293–1303, Florence, Italy, July 2019b. Association for Computational Linguistics. doi: 10.18653/v1/P19-1124. URL https://www.aclweb.org/anthology/P19-1124.

Zhuohan Li, Zi Lin, Di He, Fei Tian, Tao Qin, Liwei Wang, and Tie-Yan Liu. Hint-based training for non-autoregressive machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5708–5713, Hong Kong, China, November 2019c. Association for Computational Linguistics. doi: 10.18653/v1/D19-1573. URL https://www.aclweb.org/anthology/D19-1573.

Percy Liang, Ben Taskar, and Dan Klein. Alignment by agreement. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 104–111, New York City, USA, June 2006. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/N06-1014.

Jindřich Libovický and Jindřich Helcl. End-to-end non-autoregressive neural machine translation with connectionist temporal classification. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3016–3021, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1336. URL https://www.aclweb.org/anthology/D18-1336.

Junyang Lin, Xu Sun, Xuancheng Ren, Muyu Li, and Qi Su. Learning when to concentrate or divert attention: Self-adaptive attention temperature for neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2985–2990, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1331. URL https://www.aclweb.org/anthology/D18-1331.

Tal Linzen, Emmanuel Dupoux, and Yoav Goldberg. Assessing the ability of LSTMs to learn syntax-sensitive dependencies. *Transactions of the Association for Computational Linguistics*, 4:521–535, 2016. doi: 10.1162/tacl_a_00115. URL https://www.aclweb.org/anthology/Q16-1037.

Hairong Liu, Mingbo Ma, Liang Huang, Hao Xiong, and Zhongjun He. Robust neural machine translation with joint textual and phonetic embedding. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3044–3049, Florence, Italy, July 2019a. Association for Computational Linguistics. doi: 10.18653/v1/P19-1291. URL https://www.aclweb.org/anthology/P19-1291.

Hanxiao Liu, Karen Simonyan, and Yiming Yang. DARTS: Differentiable architecture search. In *International Conference on Learning Representations*, 2019b. URL https://openreview.net/forum?id=S1eYHoC5FX.

Liyuan Liu, Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Jiawei Han. On the variance of the adaptive learning rate and beyond. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=rkgz2aEKDr.

Yang Liu, Qun Liu, and Shouxun Lin. Tree-to-string alignment template for statistical machine translation. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 609–616, Sydney, Australia, July 2006. Association for Computational Linguistics. doi: 10.3115/1220175.1220252. URL https://www.aclweb.org/anthology/P06-1077.

Yang Liu, Yajuan Lü, and Qun Liu. Improving tree-to-tree translation with packed forests. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 558–566, Suntec, Singapore, August 2009. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/P09-1063.

Renqian Luo, Fei Tian, Tao Qin, Enhong Chen, and Tie-Yan Liu. Neural architecture optimization. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 7816–7827. Curran Associates, Inc., 2018. URL http://papers.nips.cc/paper/8007-neural-architecture-optimization.pdf.

Thang Luong, Hieu Pham, and Christopher D. Manning. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal, September 2015. Association for Computational Linguistics. doi: 10.18653/v1/D15-1166. URL https://www.aclweb.org/anthology/D15-1166.

Xuezhe Ma, Chunting Zhou, Xian Li, Graham Neubig, and Eduard Hovy. FlowSeq: Non-autoregressive conditional sequence generation with generative flow. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4282–4292, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1437. URL https://www.aclweb.org/anthology/D19-1437.

Chris J Maddison, Daniel Tarlow, and Tom Minka. A∗ sampling. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 3086–3094. Curran Associates, Inc., 2014. URL http://papers.nips.cc/paper/5449-a-sampling.pdf.

Chaitanya Malaviya, Pedro Ferreira, and André F. T. Martins. Sparse and constrained attention for neural machine translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 370–376, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-2059. URL https://www.aclweb.org/anthology/P18-2059.

Diego Marcheggiani, Joost Bastings, and Ivan Titov. Exploiting semantics in neural machine translation with graph convolutional networks. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 486–492, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-2078. URL https://www.aclweb.org/anthology/N18-2078.

Sameen Maruf and Gholamreza Haffari. Document context neural machine translation with memory networks. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1275–1284, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1118. URL https://www.aclweb.org/anthology/P18-1118.

Sameen Maruf, André F. T. Martins, and Gholamreza Haffari. Selective attention for context-aware neural machine translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3092–3102, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1313. URL https://www.aclweb.org/anthology/N19-1313.

Hrushikesh Mhaskar, Qianli Liao, and Tomaso Poggio. When and why are deep networks better than shallow ones? In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, pages 2343–2348, 2017. URL https://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14849.

Haitao Mi and Liang Huang. Forest-based translation rule extraction. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 206–214, Honolulu, Hawaii, October 2008. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/D08-1022.

Haitao Mi, Zhiguo Wang, and Abe Ittycheriah. Supervised attentions for neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2283–2288, Austin, Texas, November 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1249. URL https://www.aclweb.org/anthology/D16-1249.

Paul Michel and Graham Neubig. MTNT: A testbed for machine translation of noisy text. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 543–553, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1050. URL https://www.aclweb.org/anthology/D18-1050.

Paul Michel, Xian Li, Graham Neubig, and Juan Pino. On evaluation of adversarial perturbations for sequence-to-sequence models. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3103–3114, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1314. URL https://www.aclweb.org/anthology/N19-1314.

Lesly Miculicich, Dhananjay Ram, Nikolaos Pappas, and James Henderson. Document-level neural machine translation with hierarchical attention networks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2947–2954, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1325. URL https://www.aclweb.org/anthology/D18-1325.

Lesly Miculicich Werlen, Nikolaos Pappas, Dhananjay Ram, and Andrei Popescu-Belis. Self-attentive residual decoder for neural machine translation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1366–1379, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1124. URL https://www.aclweb.org/anthology/N18-1124.

Niv Nayman, Asaf Noy, Tal Ridnik, Itamar Friedman, Rong Jin, and Lihi Zelnik. Xnas: Neural architecture search with expert advice. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 1977–1987. Curran Associates, Inc., 2019. URL http://papers.nips.cc/paper/8472-xnas-neural-architecture-search-with-expert-advice.pdf.

Franz Josef Och. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 160–167, Sapporo, Japan, July 2003. Association for Computational Linguistics. doi: 10.3115/1075096.1075117. URL https://www.aclweb.org/anthology/P03-1021.

Franz Josef Och and Hermann Ney. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51, 2003. doi: 10.1162/089120103321337421. URL https://www.aclweb.org/anthology/J03-1002.

Franz Josef Och and Hermann Ney. The alignment template approach to statistical machine translation. *Computational Linguistics*, 30(4):417–449, 2004. doi: 10.1162/0891201042544884. URL https://www.aclweb.org/anthology/J04-4002.

Myle Ott, Sergey Edunov, David Grangier, and Michael Auli. Scaling neural machine translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 1–9, Brussels, Belgium, October 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-6301. URL https://www.aclweb.org/anthology/W18-6301.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-4009. URL https://www.aclweb.org/anthology/N19-4009.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA, July 2002. Association for Computational Linguistics. doi: 10.3115/1073083.1073135. URL https://www.aclweb.org/anthology/P02-1040.

Nikolaos Pappas and James Henderson. Deep residual output layers for neural language generation. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 5000–5011, Long Beach, California, USA, 09–15 Jun 2019. PMLR. URL http://proceedings.mlr.press/v97/pappas19a.html.

Ramakanth Pasunuru and Mohit Bansal. Continual and multi-task architecture search. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1911–1922, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1185. URL https://www.aclweb.org/anthology/P19-1185.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. URL http://papers.nips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

Junran Peng, Ming Sun, ZHAO-XIANG ZHANG, Tieniu Tan, and Junjie Yan. Efficient neural architecture transformation search in channel-level for object detection. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 14313–14322. Curran Associates, Inc., 2019. URL http://papers.nips.cc/paper/9576-efficient-neural-architecture-transformation-search-in-channel-level-for-object-detection.pdf.

Ben Peters, Vlad Niculae, and André F. T. Martins. Sparse sequence-to-sequence models. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1504–1519, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1146. URL https://www.aclweb.org/anthology/P19-1146.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1202. URL https://www.aclweb.org/anthology/N18-1202.

Hieu Pham, Melody Guan, Barret Zoph, Quoc Le, and Jeff Dean. Efficient neural architecture search via parameters sharing. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 4095–4104, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018. PMLR. URL http://proceedings.mlr.press/v80/pham18a.html.

Emmanouil Antonios Platanios, Otilia Stretcu, Graham Neubig, Barnabas Poczos, and Tom Mitchell. Competence-based curriculum learning for neural machine translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1162–1172, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1119. URL https://www.aclweb.org/anthology/N19-1119.

Martin Popel and Ondřej Bojar. Training Tips for the Transformer Model. *The Prague Bulletin of Mathematical Linguistics*, 110:43–70, April 2018. ISSN 0032-6585. doi: 10.2478/pralin-2018-0002. URL https://ufal.mff.cuni.cz/pbml/110/art-popel-bojar.pdf.

Xiao Pu, Nikolaos Pappas, James Henderson, and Andrei Popescu-Belis. Integrating weakly supervised word sense disambiguation into neural machine translation. *Transactions of the Association for Computational Linguistics*, 6:635–649, 2018. doi: 10.1162/tacl_a_00242. URL https://www.aclweb.org/anthology/Q18-1044.

Ilija Radosavovic, Raj Prateek Kosaraju, Ross Girshick, Kaiming He, and Piotr Dollar. Designing network design spaces. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. URL http://openaccess.thecvf.com/content_CVPR_2020/papers/Radosavovic_Designing_Network_Design_Spaces_CVPR_2020_paper.pdf.

Prajit Ramachandran, Barret Zoph, and Quoc V. Le. Searching for activation functions, 2017. URL https://arxiv.org/abs/1710.05941.

Motoki Sato, Jun Suzuki, and Shun Kiyono. Effective adversarial regularization for neural machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 204–210, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1020. URL https://www.aclweb.org/anthology/P19-1020.

Rico Sennrich. How grammatical is character-level neural machine translation? assessing MT quality with contrastive translation pairs. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 376–382, Valencia, Spain, April 2017. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/E17-2060.

Rico Sennrich and Biao Zhang. Revisiting low-resource neural machine translation: A case study. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 211–221, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1021. URL https://www.aclweb.org/anthology/P19-1021.

Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725. Association for Computational Linguistics, 2016a. doi: 10.18653/v1/P16-1162. URL http://aclweb.org/anthology/P16-1162.

Rico Sennrich, Barry Haddow, and Alexandra Birch. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany, August 2016b. Association for Computational Linguistics. doi: 10.18653/v1/P16-1009. URL https://www.aclweb.org/anthology/P16-1009.

Harshil Shah and David Barber. Generative neural machine translation. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 1346–1355. Curran Associates, Inc., 2018. URL http://papers.nips.cc/paper/7409-generative-neural-machine-translation.pdf.

Shiv Shankar and Sunita Sarawagi. Posterior attention models for sequence to sequence learning. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=BkltNhC9FX.

Shiv Shankar, Siddhant Garg, and Sunita Sarawagi. Surprisingly easy hard-attention for sequence to sequence learning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 640–645, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1065. URL https://www.aclweb.org/anthology/D18-1065.

Chenze Shao, Yang Feng, Jinchao Zhang, Fandong Meng, Xilin Chen, and Jie Zhou. Retrieving sequential information for non-autoregressive neural machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3013–3024, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1288. URL https://www.aclweb.org/anthology/P19-1288.

Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. Self-attention with relative position representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 464–468, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-2074. URL https://www.aclweb.org/anthology/N18-2074.

Jonathan Shen, Patrick Nguyen, Yonghui Wu, Zhifeng Chen, et al. Lingvo: a modular and scalable framework for sequence-to-sequence modeling, 2019. URL https://arxiv.org/abs/1902.08295.

Libin Shen, Jinxi Xu, and Ralph Weischedel. A new string-to-dependency machine translation algorithm with a target dependency language model. In *Proceedings of ACL-08: HLT*, pages 577–585, Columbus, Ohio, June 2008. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/P08-1066.

Shiqi Shen, Yong Cheng, Zhongjun He, Wei He, Hua Wu, Maosong Sun, and Yang Liu. Minimum risk training for neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1683–1692, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-1159. URL https://www.aclweb.org/anthology/P16-1159.

Tao Shen, Tianyi Zhou, Guodong Long, Jing Jiang, Shirui Pan, and Chengqi Zhang. Disan: Directional self-attention network for rnn/cnn-free language understanding. In Sheila A. McIlraith and Kilian Q. Weinberger, editors, *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 5446–5455. AAAI Press, 2018a. URL https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16126.

Yanyao Shen, Xu Tan, Di He, Tao Qin, and Tie-Yan Liu. Dense information flow for neural machine translation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1294–1303, New Orleans, Louisiana, June 2018b. Association for Computational Linguistics. doi: 10.18653/v1/N18-1117. URL https://www.aclweb.org/anthology/N18-1117.

Samuel L. Smith, Pieter-Jan Kindermans, and Quoc V. Le. Don't decay the learning rate, increase the batch size. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=B1Yy1BxCZ.

Matthew Snover, Bonnie Dorr, Rich Schwartz, Linnea Micciulla, and John Makhoul. A study of translation edit rate with targeted human annotation. In *Proceedings of the 7th Conference of the Association for Machine Translation of the Americas*, pages 223–231, Cambridge, Massachusetts, USA, August 2006. The Association for Machine Translation in the Americas. URL http://mt-archive.info/AMTA-2006-Snover.pdf.

David So, Quoc Le, and Chen Liang. The evolved transformer. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*,

volume 97 of *Proceedings of Machine Learning Research*, pages 5877–5886, Long Beach, California, USA, 09–15 Jun 2019. PMLR. URL http://proceedings.mlr.press/v97/so19a.html.

Richard Socher, John Bauer, Christopher D. Manning, and Andrew Y. Ng. Parsing with compositional vector grammars. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 455–465, Sofia, Bulgaria, August 2013. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/P13-1045.

Linfeng Song, Daniel Gildea, Yue Zhang, Zhiguo Wang, and Jinsong Su. Semantic neural machine translation using AMR. *Transactions of the Association for Computational Linguistics*, 7:19–31, March 2019. doi: 10.1162/tacl_a_00252. URL https://www.aclweb.org/anthology/Q19-1002.

Matthias Sperber, Graham Neubig, Jan Niehues, and Alex Waibel. Neural lattice-to-sequence models for uncertain inputs. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1380–1389, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1145. URL https://www.aclweb.org/anthology/D17-1145.

Matthias Sperber, Graham Neubig, Ngoc-Quan Pham, and Alex Waibel. Self-attentional models for lattice inputs. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1185–1197, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1115. URL https://www.aclweb.org/anthology/P19-1115.

Felix Stahlberg, Eva Hasler, Aurelien Waite, and Bill Byrne. Syntactically guided neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 299–305, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-2049. URL https://www.aclweb.org/anthology/P16-2049.

Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the importance of initialization and momentum in deep learning. In Sanjoy Dasgupta and David McAllester, editors, *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 1139–1147, Atlanta, Georgia, USA, 17–19 Jun 2013. PMLR. URL http://proceedings.mlr.press/v28/sutskever13.html.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 3104–3112. Curran Associates, Inc., 2014. URL http://papers.nips.cc/paper/5346-sequence-to-sequence-learning-with-neural-networks.pdf.

C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2818–2826, June 2016. doi: 10.1109/CVPR.2016.308. URL https://ieeexplore.ieee.org/document/7780677.

Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *AAAI*, volume 4, page 12, 2017. URL http://www.aaai.org/ocs/index.php/AAAI/AAAI17/paper/download/14806/14311.

Kai Sheng Tai, Richard Socher, and Christopher D. Manning. Improved semantic representations from tree-structured long short-term memory networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1556–1566, Beijing, China, July 2015. Association for Computational Linguistics. doi: 10.3115/v1/P15-1150. URL https://www.aclweb.org/anthology/P15-1150.

Xin Tan, Longyin Zhang, Deyi Xiong, and Guodong Zhou. Hierarchical modeling of global context for document-level neural machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1576–1585, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1168. URL https://www.aclweb.org/anthology/D19-1168.

Gongbo Tang, Mathias Müller, Annette Rios, and Rico Sennrich. Why self-attention? A targeted evaluation of neural machine translation architectures. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4263–4272, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1458. URL https://www.aclweb.org/anthology/D18-1458.

Gongbo Tang, Rico Sennrich, and Joakim Nivre. Encoders help you disambiguate word senses in neural machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1429–1435, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1149. URL https://www.aclweb.org/anthology/D19-1149.

Yi Tay, Aston Zhang, Anh Tuan Luu, Jinfeng Rao, Shuai Zhang, Shuohang Wang, Jie Fu, and Siu Cheung Hui. Lightweight and efficient neural natural language processing with quaternion networks. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1494–1503, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1145. URL https://www.aclweb.org/anthology/P19-1145.

Matus Telgarsky. benefits of depth in neural networks. In Vitaly Feldman, Alexander Rakhlin, and Ohad Shamir, editors, *29th Annual Conference on Learning Theory*, volume 49 of *Proceedings of Machine Learning Research*, pages 1517–1539, Columbia University, New York, New York, USA, 23–26 Jun 2016. PMLR. URL http://proceedings.mlr.press/v49/telgarsky16.html.

Jörg Tiedemann and Yves Scherrer. Neural machine translation with extended context. In *Proceedings of the Third Workshop on Discourse in Machine Translation*, pages 82–92, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/W17-4811. URL https://www.aclweb.org/anthology/W17-4811.

Ke Tran, Arianna Bisazza, and Christof Monz. The importance of being recurrent for modeling hierarchical structure. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4731–4736, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1503. URL https://www.aclweb.org/anthology/D18-1503.

Yao-Hung Hubert Tsai, Shaojie Bai, Makoto Yamada, Louis-Philippe Morency, and Ruslan Salakhutdinov. Transformer dissection: An unified understanding for transformer's attention via the lens of kernel. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4344–4353, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1443. URL https://www.aclweb.org/anthology/D19-1443.

Zhaopeng Tu, Zhengdong Lu, Yang Liu, Xiaohua Liu, and Hang Li. Modeling coverage for neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 76–85, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-1008. URL https://www.aclweb.org/anthology/P16-1008.

Zhaopeng Tu, Yang Liu, Lifeng Shang, Xiaohua Liu, and Hang Li. Neural machine translation with reconstruction. In Satinder P. Singh and Shaul Markovitch, editors, *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9,*

2017, San Francisco, California, USA, pages 3097–3103. AAAI Press, 2017. URL http://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14161.

Zhaopeng Tu, Yang Liu, Shuming Shi, and Tong Zhang. Learning to remember translation history with a continuous cache. *Transactions of the Association for Computational Linguistics*, 6:407–420, 2018. doi: 10.1162/tacl_a_00029. URL https://www.aclweb.org/anthology/Q18-1029.

Arata Ugawa, Akihiro Tamura, Takashi Ninomiya, Hiroya Takamura, and Manabu Okumura. Neural machine translation incorporating named entity. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3240–3250, Santa Fe, New Mexico, USA, August 2018. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/C18-1274.

Vaibhav Vaibhav, Sumeet Singh, Craig Stewart, and Graham Neubig. Improving robustness of machine translation with synthetic noise. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1916–1920, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1190. URL https://www.aclweb.org/anthology/N19-1190.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc., 2017. URL http://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf.

Ashish Vaswani, Samy Bengio, Eugene Brevdo, Francois Chollet, Aidan N. Gomez, Stephan Gouws, Llion Jones, Łukasz Kaiser, Nal Kalchbrenner, Niki Parmar, Ryan Sepassi, Noam Shazeer, and Jakob Uszkoreit. Tensor2tensor for neural machine translation. *CoRR*, abs/1803.07416, 2018. URL http://arxiv.org/abs/1803.07416.

Stephan Vogel, Hermann Ney, and Christoph Tillmann. HMM-based word alignment in statistical translation. In *COLING 1996 Volume 2: The 16th International Conference on Computational Linguistics*, 1996. URL https://www.aclweb.org/anthology/C96-2141.

Elena Voita, Pavel Serdyukov, Rico Sennrich, and Ivan Titov. Context-aware neural machine translation learns anaphora resolution. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1264–1274, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1117. URL https://www.aclweb.org/anthology/P18-1117.

Elena Voita, Rico Sennrich, and Ivan Titov. The bottom-up evolution of representations in the transformer: A study with machine translation and language modeling objectives. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4395–4405, Hong Kong, China, November 2019a. Association for Computational Linguistics. doi: 10.18653/v1/D19-1448. URL https://www.aclweb.org/anthology/D19-1448.

Elena Voita, Rico Sennrich, and Ivan Titov. Context-aware monolingual repair for neural machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 876–885, Hong Kong, China, November 2019b. Association for Computational Linguistics. doi: 10.18653/v1/D19-1081. URL https://www.aclweb.org/anthology/D19-1081.

Elena Voita, Rico Sennrich, and Ivan Titov. When a good translation is wrong in context: Context-aware machine translation improves on deixis, ellipsis, and lexical cohesion. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1198–1212, Florence, Italy, July 2019c. Association for Computational Linguistics. doi: 10.18653/v1/P19-1116. URL https://www.aclweb.org/anthology/P19-1116.

Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5797–5808, Florence, Italy, July 2019d. Association for Computational Linguistics. doi: 10.18653/v1/P19-1580. URL https://www.aclweb.org/anthology/P19-1580.

Benyou Wang, Donghao Zhao, Christina Lioma, Qiuchi Li, Peng Zhang, and Jakob Grue Simonsen. Encoding word order in complex embeddings. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=Hke-WTVtwr.

Longyue Wang, Zhaopeng Tu, Andy Way, and Qun Liu. Exploiting cross-sentence context for neural machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2826–2831, Copenhagen, Denmark, September 2017a. Association for Computational Linguistics. doi: 10.18653/v1/D17-1301. URL https://www.aclweb.org/anthology/D17-1301.

Longyue Wang, Zhaopeng Tu, Andy Way, and Qun Liu. Learning to jointly translate and predict dropped pronouns with a shared reconstruction mechanism. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2997–3002, Brussels, Belgium, October-November 2018a. Association for Computational Linguistics. doi: 10.18653/v1/D18-1333. URL https://www.aclweb.org/anthology/D18-1333.

Longyue Wang, Zhaopeng Tu, Xing Wang, and Shuming Shi. One model to learn both: Zero pronoun prediction and translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 921–930, Hong Kong, China, November 2019a. Association for Computational Linguistics. doi: 10.18653/v1/D19-1085. URL https://www.aclweb.org/anthology/D19-1085.

Mingxuan Wang, Zhengdong Lu, Jie Zhou, and Qun Liu. Deep neural machine translation with linear associative unit. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 136–145, Vancouver, Canada, July 2017b. Association for Computational Linguistics. doi: 10.18653/v1/P17-1013. URL https://www.aclweb.org/anthology/P17-1013.

Mingxuan Wang, Jun Xie, Zhixing Tan, Jinsong Su, Deyi Xiong, and Lei Li. Towards linear time neural machine translation with capsule networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 803–812, Hong Kong, China, November 2019b. Association for Computational Linguistics. doi: 10.18653/v1/D19-1074. URL https://www.aclweb.org/anthology/D19-1074.

Qiang Wang, Fuxue Li, Tong Xiao, Yanyang Li, Yinqiao Li, and Jingbo Zhu. Multi-layer representation fusion for neural machine translation. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3015–3026, Santa Fe, New Mexico, USA, August 2018b. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/C18-1255.

Qiang Wang, Bei Li, Tong Xiao, Jingbo Zhu, Changliang Li, Derek F. Wong, and Lidia S. Chao. Learning deep transformer models for machine translation. In *Proceedings of the 57th Conference of the Association for Computational Linguistics*, pages 1810–1822, Florence, Italy, July 2019c. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/P19-1176.

Rui Wang, Masao Utiyama, and Eiichiro Sumita. Dynamic sentence sampling for efficient training of neural machine translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 298–304, Melbourne, Australia, July 2018c. Association for Computational Linguistics. doi: 10.18653/v1/P18-2048. URL https://www.aclweb.org/anthology/P18-2048.

Shuo Wang, Yang Liu, Chao Wang, Huanbo Luan, and Maosong Sun. Improving back-translation with uncertainty-based confidence estimation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 791–802, Hong Kong, China, November 2019d. Association for Computational Linguistics. doi: 10.18653/v1/D19-1073. URL https://www.aclweb.org/anthology/D19-1073.

Weiyue Wang, Derui Zhu, Tamer Alkhouli, Zixuan Gan, and Hermann Ney. Neural hidden Markov model for machine translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 377–382, Melbourne, Australia, July 2018d. Association for Computational Linguistics. doi: 10.18653/v1/P18-2060. URL https://www.aclweb.org/anthology/P18-2060.

Xing Wang, Zhengdong Lu, Zhaopeng Tu, Hang Li, Deyi Xiong, and Min Zhang. Neural machine translation advised by statistical machine translation. In Satinder P. Singh and Shaul Markovitch, editors, *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, pages 3330–3336. AAAI Press, 2017c. URL http://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14451.

Xing Wang, Zhengdong Lu, Zhaopeng Tu, Hang Li, Deyi Xiong, and Min Zhang. Neural machine translation advised by statistical machine translation. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, pages 3330–3336, 2017d. URL https://www.aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14451.

Xing Wang, Zhaopeng Tu, Deyi Xiong, and Min Zhang. Translating phrases in neural machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1421–1431, Copenhagen, Denmark, September 2017e. Association for Computational Linguistics. doi: 10.18653/v1/D17-1149. URL https://www.aclweb.org/anthology/D17-1149.

Xing Wang, Zhaopeng Tu, Longyue Wang, and Shuming Shi. Exploiting sentential context for neural machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6197–6203, Florence, Italy, July 2019e. Association for Computational Linguistics. doi: 10.18653/v1/P19-1624. URL https://www.aclweb.org/anthology/P19-1624.

Xing Wang, Zhaopeng Tu, Longyue Wang, and Shuming Shi. Self-attention with structural position representations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1403–1409, Hong Kong, China, November 2019f. Association for Computational Linguistics. doi: 10.18653/v1/D19-1145. URL https://www.aclweb.org/anthology/D19-1145.

Yiren Wang, Fei Tian, Di He, Tao Qin, ChengXiang Zhai, and Tie-Yan Liu. Non-autoregressive machine translation with auxiliary regularization. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 5377–5384. AAAI Press, 2019g. doi: 10.1609/aaai.v33i01.33015377. URL https://doi.org/10.1609/aaai.v33i01.33015377.

Bingzhen Wei, Mingxuan Wang, Hao Zhou, Junyang Lin, and Xu Sun. Imitation learning for non-autoregressive neural machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1304–1312, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1125. URL https://www.aclweb.org/anthology/P19-1125.

Rongxiang Weng, Shujian Huang, Zaixiang Zheng, Xinyu Dai, and Jiajun Chen. Neural machine translation with word predictions. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 136–145, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1013. URL https://www.aclweb.org/anthology/D17-1013.

John Wieting, Taylor Berg-Kirkpatrick, Kevin Gimpel, and Graham Neubig. Beyond BLEU:training neural machine translation with semantic similarity. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4344–4355, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1427. URL https://www.aclweb.org/anthology/P19-1427.

Sam Wiseman and Alexander M. Rush. Sequence-to-sequence learning as beam-search optimization. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1296–1306, Austin, Texas, November 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1137. URL https://www.aclweb.org/anthology/D16-1137.

Catherine Wong, Neil Houlsby, Yifeng Lu, and Andrea Gesmundo. Transfer learning with neural automl. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 8356–8365. Curran Associates, Inc., 2018. URL http://papers.nips.cc/paper/8056-transfer-learning-with-neural-automl.pdf.

Bichen Wu, Xiaoliang Dai, Peizhao Zhang, Yanghan Wang, Fei Sun, Yiming Wu, Yuandong Tian, Peter Vajda, Yangqing Jia, and Kurt Keutzer. Fbnet: Hardware-aware efficient convnet design via differentiable neural architecture search. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019a. URL http://openaccess.thecvf.com/content_CVPR_2019/papers/Wu_FBNet_Hardware-Aware_Efficient_ConvNet_Design_via_Differentiable_Neural_Architecture_Search_CVPR_2019_paper.pdf.

Dekai Wu. Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. *Computational Linguistics*, 23(3):377–403, 1997. URL https://www.aclweb.org/anthology/J97-3002.

Felix Wu, Angela Fan, Alexei Baevski, Yann Dauphin, and Michael Auli. Pay less attention with lightweight and dynamic convolutions. In *International Conference on Learning Representations*, 2019b. URL https://openreview.net/forum?id=SkVhlh09tX.

Lijun Wu, Yiren Wang, Yingce Xia, Fei Tian, Fei Gao, Tao Qin, Jianhuang Lai, and Tie-Yan Liu. Depth growing for neural machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5558–5563, Florence, Italy, July 2019c. Association for Computational Linguistics. doi: 10.18653/v1/P19-1558. URL https://www.aclweb.org/anthology/P19-1558.

Shuangzhi Wu, Dongdong Zhang, Nan Yang, Mu Li, and Ming Zhou. Sequence-to-dependency neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 698–707, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1065. URL https://www.aclweb.org/anthology/P17-1065.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. Google's neural machine translation system: Bridging the gap between human and machine translation. *CoRR*, abs/1609.08144, 2016. URL http://arxiv.org/abs/1609.08144.

Fengshun Xiao, Jiangtong Li, Hai Zhao, Rui Wang, and Kehai Chen. Lattice-based transformer encoder for neural machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3090–3097, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1298. URL https://www.aclweb.org/anthology/P19-1298.

Sirui Xie, Hehui Zheng, Chunxiao Liu, and Liang Lin. SNAS: stochastic neural architecture search. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=rylqooRqK7.

Deyi Xiong, Qun Liu, and Shouxun Lin. Maximum entropy based phrase reordering model for statistical machine translation. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 521–528, Sydney, Australia, July 2006. Association for Computational Linguistics. doi: 10.3115/1220175.1220241. URL https://www.aclweb.org/anthology/P06-1066.

Hao Xiong, Zhongjun He, Hua Wu, and Haifeng Wang. Modeling coherence for discourse neural machine translation. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 7338–7345. AAAI Press, 2019. ISBN 978-1-57735-809-1. doi: 10.1609/aaai.v33i01.33017338. URL https://doi.org/10.1609/aaai.v33i01.33017338.

Hongfei Xu and Qiuhui Liu. Neutron: an Implementation of the Transformer Translation Model and its Variants. *arXiv preprint arXiv:1903.07402*, March 2019. URL https://arxiv.org/abs/1903.07402.

Hongfei Xu, Qiuhui Liu, Josef van Genabith, Deyi Xiong, and Jingyi Zhang. Lipschitz constrained parameter initialization for deep transformers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 397–402, Online, July 2020a. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/2020.acl-main.38.

Hongfei Xu, Josef van Genabith, Deyi Xiong, and Qiuhui Liu. Dynamically adjusting transformer batch size by monitoring gradient direction change. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3519–3524, Online, July 2020b. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/2020.acl-main.323.

Hongfei Xu, Josef van Genabith, Deyi Xiong, Qiuhui Liu, and Jingyi Zhang. Learning source phrase representations for neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 386–396, Online, July 2020c. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/2020.acl-main.37.

Hongfei Xu, Deyi Xiong, Josef van Genabith, and Qiuhui Liu. Efficient context-aware neural machine translation with layer-wise weighting and input-aware gating. In Christian Bessiere, editor, *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 3933–3940. International Joint Conferences on Artificial Intelligence Organization, 7 2020d. doi: 10.24963/ijcai.2020/544. URL https://doi.org/10.24963/ijcai.2020/544. Main track.

Mingzhou Xu, Derek F. Wong, Baosong Yang, Yue Zhang, and Lidia S. Chao. Leveraging local and global patterns for self-attention networks. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3069–3075, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1295. URL https://www.aclweb.org/anthology/P19-1295.

Baosong Yang, Derek F. Wong, Tong Xiao, Lidia S. Chao, and Jingbo Zhu. Towards bidirectional hierarchical representations for attention-based neural machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1432–1441, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1150. URL https://www.aclweb.org/anthology/D17-1150.

Baosong Yang, Zhaopeng Tu, Derek F. Wong, Fandong Meng, Lidia S. Chao, and Tong Zhang. Modeling localness for self-attention networks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4449–4458, Brussels, Belgium, October-November 2018a. Association for Computational Linguistics. doi: 10.18653/v1/D18-1475. URL https://www.aclweb.org/anthology/D18-1475.

Baosong Yang, Jian Li, Derek F. Wong, Lidia S. Chao, Xing Wang, and Zhaopeng Tu. Context-aware self-attention networks. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 387–394. AAAI Press, 2019a. doi: 10.1609/aaai.v33i01.3301387. URL https://doi.org/10.1609/aaai.v33i01.3301387.

Baosong Yang, Longyue Wang, Derek F. Wong, Lidia S. Chao, and Zhaopeng Tu. Assessing the ability of self-attention networks to learn word order. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3635–3644, Florence, Italy, July 2019b. Association for Computational Linguistics. doi: 10.18653/v1/P19-1354. URL https://www.aclweb.org/anthology/P19-1354.

Baosong Yang, Longyue Wang, Derek F. Wong, Lidia S. Chao, and Zhaopeng Tu. Convolutional self-attention networks. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4040–4045, Minneapolis, Minnesota, June 2019c. Association for Computational Linguistics. doi: 10.18653/v1/N19-1407. URL https://www.aclweb.org/anthology/N19-1407.

Mingming Yang, Rui Wang, Kehai Chen, Masao Utiyama, Eiichiro Sumita, Min Zhang, and Tiejun Zhao. Sentence-level agreement for neural machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3076–3082, Florence, Italy, July 2019d. Association for Computational Linguistics. doi: 10.18653/v1/P19-1296. URL https://www.aclweb.org/anthology/P19-1296.

Xuewen Yang, Yingru Liu, Dongliang Xie, Xin Wang, and Niranjan Balasubramanian. Latent part-of-speech sequences for neural machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 780–790, Hong Kong, China, November 2019e. Association for Computational Linguistics. doi: 10.18653/v1/D19-1072. URL https://www.aclweb.org/anthology/D19-1072.

Zhen Yang, Wei Chen, Feng Wang, and Bo Xu. Improving neural machine translation with conditional sequence generative adversarial nets. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1346–1355, New Orleans, Louisiana, June 2018b. Association for Computational Linguistics. doi: 10.18653/v1/N18-1122. URL https://www.aclweb.org/anthology/N18-1122.

Zonghan Yang, Yong Cheng, Yang Liu, and Maosong Sun. Reducing word omission errors in neural machine translation: A contrastive learning approach. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6191–6196, Florence, Italy, July 2019f. Association for Computational Linguistics. doi: 10.18653/v1/P19-1623. URL https://www.aclweb.org/anthology/P19-1623.

Fisher Yu, Dequan Wang, Evan Shelhamer, and Trevor Darrell. Deep layer aggregation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. URL http://openaccess.thecvf.com/content_cvpr_2018/papers/Yu_Deep_Layer_Aggregation_CVPR_2018_paper.pdf.

Biao Zhang, Deyi Xiong, Jinsong Su, Hong Duan, and Min Zhang. Variational neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 521–530, Austin, Texas, November 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1050. URL https://www.aclweb.org/anthology/D16-1050.

Biao Zhang, Deyi Xiong, and jinsong Su. Accelerating neural transformer via an average attention network. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1789–1798. Association for Computational Linguistics, 2018a. URL http://aclweb.org/anthology/P18-1166.

Biao Zhang, Deyi Xiong, Jinsong Su, Qian Lin, and Huiji Zhang. Simplifying neural machine translation with addition-subtraction twin-gated recurrent networks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4273–4283, Brussels, Belgium, October-November 2018b. Association for Computational Linguistics. doi: 10.18653/v1/D18-1459. URL https://www.aclweb.org/anthology/D18-1459.

Biao Zhang, Ivan Titov, and Rico Sennrich. Improving deep transformer with depth-scaled initialization and merged attention. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 898–909, Hong Kong, China, November 2019a. Association for Computational Linguistics. doi: 10.18653/v1/D19-1083. URL https://www.aclweb.org/anthology/D19-1083.

Jiacheng Zhang, Yanzhuo Ding, Shiqi Shen, Yong Cheng, Maosong Sun, Huanbo Luan, and Yang Liu. Thumt: An open source toolkit for neural machine translation. *arXiv preprint arXiv:1706.06415*, 2017a. URL http://arxiv.org/abs/1706.06415.

Jiacheng Zhang, Yang Liu, Huanbo Luan, Jingfang Xu, and Maosong Sun. Prior knowledge integration for neural machine translation using posterior regularization. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1514–1523, Vancouver, Canada, July 2017b. Association for Computational Linguistics. doi: 10.18653/v1/P17-1139. URL https://www.aclweb.org/anthology/P17-1139.

Jiacheng Zhang, Huanbo Luan, Maosong Sun, Feifei Zhai, Jingfang Xu, Min Zhang, and Yang Liu. Improving the transformer translation model with document-level context. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 533–542, Brussels, Belgium, October-November 2018c. Association for Computational Linguistics. doi: 10.18653/v1/D18-1049. URL https://www.aclweb.org/anthology/D18-1049.

Jinchao Zhang, Mingxuan Wang, Qun Liu, and Jie Zhou. Incorporating word reordering knowledge into attention-based neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1524–1534, Vancouver, Canada, July 2017c. Association for Computational Linguistics. doi: 10.18653/v1/P17-1140. URL https://www.aclweb.org/anthology/P17-1140.

Jingyi Zhang, Masao Utiyama, Eiichro Sumita, Graham Neubig, and Satoshi Nakamura. Guiding neural machine translation with retrieved translation pieces. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1325–1335, New Orleans, Louisiana, June 2018d. Association for Computational Linguistics. doi: 10.18653/v1/N18-1120. URL https://www.aclweb.org/anthology/N18-1120.

Meishan Zhang, Zhenghua Li, Guohong Fu, and Min Zhang. Syntax-enhanced neural machine translation with syntax-aware word representations. In *Proceedings of the 2019 Conference of the*

North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 1151–1161, Minneapolis, Minnesota, June 2019b. Association for Computational Linguistics. doi: 10.18653/v1/N19-1118. URL https://www.aclweb.org/anthology/N19-1118.

Michael Zhang, James Lucas, Jimmy Ba, and Geoffrey E Hinton. Lookahead optimizer: k steps forward, 1 step back. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 9593–9604. Curran Associates, Inc., 2019c. URL http://papers.nips.cc/paper/9155-lookahead-optimizer-k-steps-forward-1-step-back.pdf.

Min Zhang, Hongfei Jiang, Aiti Aw, Haizhou Li, Chew Lim Tan, and Sheng Li. A tree sequence alignment-based tree-to-tree translation model. In *Proceedings of ACL-08: HLT*, pages 559–567, Columbus, Ohio, June 2008. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/P08-1064.

Wen Zhang, Liang Huang, Yang Feng, Lei Shen, and Qun Liu. Speeding up neural machine translation decoding by cube pruning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4284–4294, Brussels, Belgium, October-November 2018e. Association for Computational Linguistics. doi: 10.18653/v1/D18-1460. URL https://www.aclweb.org/anthology/D18-1460.

Wen Zhang, Yang Feng, Fandong Meng, Di You, and Qun Liu. Bridging the gap between training and inference for neural machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4334–4343, Florence, Italy, July 2019d. Association for Computational Linguistics. doi: 10.18653/v1/P19-1426. URL https://www.aclweb.org/anthology/P19-1426.

Zhisong Zhang, Rui Wang, Masao Utiyama, Eiichiro Sumita, and Hai Zhao. Exploring recombination for efficient decoding of neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4785–4790, Brussels, Belgium, October-November 2018f. Association for Computational Linguistics. doi: 10.18653/v1/D18-1511. URL https://www.aclweb.org/anthology/D18-1511.

Zhengli Zhao, Dheeru Dua, and Sameer Singh. Generating natural adversarial examples. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=H1BLjgZCb.

Zaixiang Zheng, Hao Zhou, Shujian Huang, Lei Li, Xin-Yu Dai, and Jiajun Chen. Mirror-generative neural machine translation. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=HkxQRTNYPH.

Hattie Zhou, Janice Lan, Rosanne Liu, and Jason Yosinski. Deconstructing lottery tickets: Zeros, signs, and the supermask. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 3597–3607. Curran Associates, Inc., 2019. URL http://papers.nips.cc/paper/8618-deconstructing-lottery-tickets-zeros-signs-and-the-supermask.pdf.

Jie Zhou, Ying Cao, Xuguang Wang, Peng Li, and Wei Xu. Deep recurrent models with fast-forward connections for neural machine translation. *Transactions of the Association for Computational Linguistics*, 4:371–383, 2016. doi: 10.1162/tacl_a_00105. URL https://www.aclweb.org/anthology/Q16-1027.

Jinhua Zhu, Yingce Xia, Lijun Wu, Di He, Tao Qin, Wengang Zhou, Houqiang Li, and Tieyan Liu. Incorporating bert into neural machine translation. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=Hyl7ygStwB.