MONASH University

Improving Resource-constrained Machine Translation and Text Generation Using Knowledge Transition

Fahimeh Saleh

A thesis submitted for the degree of *Doctor of Philosophy* at

Monash University in 2021

Faculty of Information Technology

This thesis is dedicated to my parents,

for their endless love, support, and encouragement.

Acknowledgements

I would like to express my gratitude to everybody who backed me throughout my PhD journey.

I would like to express my deepest gratitude to my amazing supervisors, Professor Wray Buntine and Dr Lan Du, for their generous and continuous support during my PhD. Their immense knowledge and encouragement inspired me and helped me grow both in my academic and personal life. Beyond research, Wray also cares for the well-being of his students. I would never forget his great compassion and sympathy with his students during the COVID-19 time and the Fridays' meetings we had during the lengthy lockdown in Melbourne. Thank you for teaching me so much and helping me grow during my study.

I would like to express my great appreciation to my fantastic internship supervisors in Naver Labs Europe, Professor Laurent Besacier, Dr Ioan Calapodescu, and Dr Alexandre Bérard, for their tremendous support, motivation, and patience throughout my internship program. Specially, I would like to express my sincere gratitude to Alex, whose expertise was invaluable in formulating my research. His insightful feedback pushed me to sharpen my thinking which brought my work to a higher level.

I would like to express my deep gratitude to Professor Bernd Meyer and Associate Professor Guido Tack, who gave me a feeling of relief and calm to share my concerns with and then I was sure that I receive the best support and advice. Thanks for your great,

continuous help during my PhD. I would like to thank all members of my panel committee, Dr Mario Boley, Dr Daniel Schmidt, Professor Mark Wallace, and Professor Ron Steinfeld, for their valuable feedback at my PhD milestones.

Furthermore, I would like to thank Professor Sue McKemmish, Danette Deriane, and Helen Cridland from FIT Postgraduate Research Team for their continued support from the very first day of my candidature. I am also very grateful to Julie Holden for her generous support in academic writing and speaking and her great passion for teaching us writing professionally and precisely.

I would also like to thank Associate Professor Reza Haffari for his efforts in the earlier part of my thesis as an advisor.

I am also grateful to Monash University for providing my PhD scholarship, conference travel funding, and Graduate Research Completion Award. Also, thanks to the support team at Monash Advanced Research Computing Hybrid (MonARCH) and Multi-modal Australian ScienceS Imaging and Visualization Environment (MASSIVE) for all their technical help with using computational resources.

I would like to extend my gratitude to my amazing colleagues, Narjes Askarian, Sameen Maruf, Snow Situ, Trang Vu, Poorya Zaremoodi, Xuanli He, and Najam Zaidi, for their empathy, trustworthiness, and encouragement during the time of study we had together.

I would like to extend my sincere thanks to my wonderful friends, Sanaz Nikfalazar, Shirin Ghaffarian, Ahmad Kazemi, Saman Ahmadi, Fatemeh Azhari, Hamid Askari, Leila Tavakoli, Morteza Haghighi, Sara Namin, Mohammad Esmaeil Zadeh, Mohammad Najafi, Mohammad Rahimi, Mahshid Jalilian, Fatemeh Shiri, and Dhananjay Thiruvady, with whom I shared a very close and warm bond in Australia. Thanks for being loyal, cordial, and generous to me and making me happy and feel at home. My special thanks to my dear friend, Ahmad Kazemi, who was very supportive during my stay in Clayton Campus. I value and appreciate the time you spent to listen to me whenever I needed someone to speak with.

I would also like to express my sincere thanks to my life-long, beloved friends, Nobahar Arian, Forouzan Chavoshi, Sanam Hooshver, Saye Merat, Maryam Keyvanara, Arezu Chehregosha, Mojgan Aghayee, and Amin Bahiraei, whose friendship has always been growing stronger regardless of the distance that separates us or the time that passes.

My heartfelt thanks go to my inspiring parents, who set me off on the road to this PhD a long time ago. Your love, support and encouragement were worth more than I can express on paper. This accomplishment would not have been possible without your support. Thanks to my dear sisters Tayebeh and Salimeh for their emotional support, and their endless love.

Finally, a very special word of thanks goes to my lovely sister and brother-in-law, Fatemeh and Sadegh. I am touched beyond words when it comes to speaking about your unconditional support, kindness, and care during my intense study in Australia. I would never forget your encouragement for me to apply for a PhD in Australia, your help to settle me in, and your support when the time got rough, especially during the COVID-19 days. You were not only my beloved family but also my amazing friends who have inspired me to think outside the box in my research and pushed me toward my goals.

Declaration

This thesis is an original work of my research and contains no material which has been accepted for the award of any other degree or diploma at any university or equivalent institution and that, to the best of my knowledge and belief, this thesis contains no material previously published or written by another person, except where due reference is made in the text of the thesis.

Signature: .....

Print Name: Fahimeh Saleh

Date : 01/01/2022

Abstract

Recent advances in neural models have significantly improved the performance of the Neural Text Generation (NTG) systems such as Machine Translation (MT) and Natural Language Generation (NLG). A typical Neural Machine Translation (NMT) task is the process of building and training a large neural network that takes a sentence in one language and generates a corresponding translation in another language. The classic neural NLG task aims to generates coherent, informative documents from limited structured metadata to fulfil a communicative goal. Despite their improvements, the constraint on resources is still an open problem in training accurate NTG models. Most of the existing NTG models are limited by i) their inability to fully exploit the training resources, e.g., sentence-level text generation approaches that can not go beyond the sentence-level context; and ii) their incapacity to deal with the data scarcity issue, an example of which is bilingually low-resource scenarios for training high-quality NMT/NLG models.

This research addresses those two limitations due to the resource constraints and introduces novel approaches that transit knowledge from high-resource NTG models to low-resource ones utilizing either transfer learning or knowledge distillation. In both transfer learning and knowledge distillation approaches, we aim to transit knowledge from an expert model with higher knowledge capacity usually trained in a high-resource setting to another model that needs to be trained in a low-resource scenario. In transfer learning, the parameters of the expert model are used to initialize the new model from the same or re-

lated task. This approach is practical when the source and target tasks have similar model components in the knowledge transition scheme. In knowledge distillation, however, the predictions of the expert model are used to train the new model where the source and target tasks are the same for both models, but no sharing of model components is required. In this thesis, we focus on three knowledge transition schemes in different low-resource scenarios: i) *Knowledge transition from a high-resource task (NMT) to a different low-resource task (NLG)*: In this approach, we introduce a compact way of encoding the metadata available in the original NLG database at the document-level and enrich the data with extra facts that can be inferred with minimal knowledge of the task. We then initialize the NLG model from a pretrained document-level NMT model while feeding it with *document-level* structured metadata from the NLG task. We show that our end-to-end NLG model trained with document-level metadata is not only able to capture document-level structure coherently, but also select and order metadata information in the generated story adequately. ii) *Knowledge transition from an ensemble of high-resource models to a low-resource model from the same task (MT) in a bilingual setting, regardless of the relatedness/similarity of contributing languages*: In this approach, we first propose a many-to-one transfer learning method that effectively transfers knowledge from multiple high-resource language pairs to a target low-resource language-pair of interest in a bilingual setting. Since the fine-tuned models from different high-resource language pairs may offer complementary syntactic and/or semantic knowledge in the target language pair, we dynamically adjust the contribution of teachers during the distillation process via our Adaptive Knowledge Distillation (AKD) approach, aiming to utilize the best of all teachers in the ensemble. iii) *Knowledge transition from the groups of NMT models to a single multilingual NMT model given the language similarity in a hierarchy structure*: In this approach, we cluster languages based on their typological similarities and then distill their related NMT models' knowledge to group-specific teacher assistant models. This process is repeated while different clusters are merged down to a single super-cluster. Our Hierarchical Knowledge Distillation (HKD) approach yields a multilingual NMT model that not only achieves high quality translations for low-resource language pairs, but also mitigates negative transfer by leveraging common characteristics of

languages belonging to the same language group.

In summary, contributions to this research improve the resource-constrained NMT and NLG utilizing knowledge transition. In this thesis, we use different knowledge transition methods, i.e., transfer learning and knowledge distillation, to address the research gaps in low-resource text generation scenarios. Furthermore, we highlight the deficiency of conventional transfer learning and knowledge distillation approaches and address it by introducing the novel proposed techniques. All the methods introduced in this research are evaluated on standard benchmark datasets, and the experiments provide compelling evidence that our approaches are more efficient than the contemporary baselines.

Table of Contents

List of Figures

List of Tables

[2] http://www.loc.gov/standards/iso639-2/php/English_list.php

List of Abbreviations

AKD . . . . . Adaptive Knowledge Distillation

BPE . . . . . Byte Pair Encoding

DGT . . . . . Document-level Generation and Translation

GPU . . . . . Graphics Processing/Processor Unit

GT . . . . . Ground Truth

HKD . . . . . Hierarchical Knowledge Distillation

IoT . . . . . The Internet of Things

KD . . . . . Knowledge Distillation

KT . . . . . Knowledge Transition

MNMT . . . . . Multilingual Neural Machine Translation

NLG . . . . . Natural Language Generation

NLP . . . . . Natural Language Processing

NMT . . . . . Neural Machine Translation

NTG . . . . . Neural Text Generation

RBMT . . . . . Rule Based Machine Translation

RNN . . . . . Recurrent Neural Network

SGD . . . . . Stochastic gradient descent

SKD . . . . . Selective Knowledge Distillation

SMT . . . . . Statistical Machine Translation

List of Publications

[EMNLP2019] **Fahimeh Saleh**, Alexandre Berard, Ioan Calapodescu, and Laurent Besacier. *Naver Labs Europe's Systems for the Document-Level Generation and Translation Task at WNGT 2019.* In Proceedings of the 3rd Workshop on Neural Generation and Translation, at EMNLP-IJCNLP, pp. 273-279. 2019.

[COLING2020] **Fahimeh Saleh**, Wray Buntine, and Gholamreza Haffari. *Collective Wisdom: Improving Low-resource Neural Machine Translation using Adaptive Knowledge Distillation.* In Proceedings of the 28th International Conference on Computational Linguistics, pp. 3413-3421. 2020.

[ACM Computing Surveys] Sameen Maruf, **Fahimeh Saleh**, and Gholamreza Haffari. *A survey on document-level neural machine translation: Methods and evaluation.* ACM Computing Surveys (CSUR) 54, no. 2 (2021): 1-36.

[EMNLP2021] **Fahimeh Saleh**, Wray Buntine, Gholamreza Haffari, and Lan Du. *Multilingual Neural Machine Translation: Can Linguistic Hierarchies Help?* In Proceedings of the finding papers in the 2021 Conference on Empirical Methods in Natural Language Processing.

1 | Introduction

Neural Text Generation (NTG) systems have recently been introduced as a promising approach and achieved high-level performance on Natural Language Processing (NLP) tasks such as Machine Translation (MT) and data-to-text generation (Stahlberg, 2019; Maruf, Saleh, & Haffari, 2021; Gatt & Krahmer, 2018).

Specifically, Machine Translation is the process of building and training a system that reads a sentence in source language and generates a corresponding translation in the target language. Until a few years ago, MT was mainly formalized through statistical techniques, known as statistical machine translation (SMT), in which translations are generated by applying statistical models based on the analysis of features extracted from the bilingual corpus. With the insurgence of neural networks, however, the most advanced Neural Machine Translation (NMT) systems emerged which require little to no feature engineering (Maruf et al., 2021).

Data-to-text generation is a classic problem in Natural Language Generation (NLG) which involves taking structured meta-data (e.g., a table) as input, and generating a fluent descriptive document as an output (Kukich, 1983; Holmes-Higgin, 1994; Dale & Reiter, 1997). An example of which is "Automated journalism" or "Robot journalism" (Montal & Reich, 2017) that have had a remarkable effect on the field of journalism in terms of efficiency and cost-cutting (Van Dalen, 2012; Clerwall, 2014; Young & Hermida, 2015).

Unlike the machine translation objective which is full transduction from a sentence in the source language to the translation in the target language, data-to-text generation, typically aims to transduce from the abstract structured meta-data to a full coherent document. This objective requires addressing two separate challenges: (a) identifying the most important information from input data, and (b) verbalizing data as a coherent document (Dale & Reiter, 1997; Jurafsky & Martin, 2009; Mei, Bansal, & Walter, 2016). These two challenges have traditionally been addressed separately as different modules in pipeline SMT-like systems (McKeown, 1985; Reiter & Dale, 2000a). However, neural generation systems, which are typically trained end-to-end as conditional language models (Mikolov, Karafiát, Burget, Černockỳ, & Khudanpur, 2010) make an integration in this pipeline and remove the distinction in these two challenges (Sutskever, Martens, & Hinton, 2011).

Basically, the most important advantage of neural models in both NMT and NLG is their flexibility (no need to feature engineering) and their ability to learn in an end-to-end manner. These advantages make the text generation tasks more effective and straightforward than the pipeline statistical approaches which have strong assumptions of locality and need feature engineering (Koehn, Och, & Marcu, 2003). Despite their advantages, neural models are notoriously data-hungry. Therefore, they require a significant amount of labelled data and have a steep learning curve according to the amount of training data, resulting in poorer quality in low-resource settings, but better performance in high-resource ones (Koehn & Knowles, 2017). Thus, when the training data is limited, neural models are prone to overfitting, resulting in inferior performance. Subsequently, for domains that suffer from a shortage of annotated data like machine translation and data-to-text generation, the applicability of deep learning methods becomes restricted to the small subset of world languages which have a sizeable available volume of translation or text generation training resources. The goal of this thesis is to develop effective methods for learning resource-constrained NTG tasks. In this chapter, we first provide the motivation behind this research, followed by our objectives. Then, we summarize our contributions and the thesis outline.

1.1 Motivation

Despite the significant improvements of NTG systems such as NMT and NLG, there are still some open challenges in this domain due to constraints on resources. The resource limitation is sometimes rooted in the model's incapability to use maximum resources (e.g., processing full document instead of processing sentences independently), or more typically because of the lack of annotated training data.

Most of the recent NMT systems process sentences in isolation and tend to ignore extra-sentential information from the context, even though an extended context help the system to prevent mistakes in vague cases and thus improve translation coherency. This simplifying independence assumption commonly made because of technical challenges raised by representing text as documents rather than sentences. This is due to the enormity of the search space over a large number of translation variables (i.e., the number of sentences in the document) as well as their unbounded domain (i.e., all sentences in the target language). However, the need for the wider context and discourse has been long identified by early works on MT in 60's (Bar-Hillel, 1964). However, this long-standing and challenging problem has not been given proper justice thus far. Bar Hillel's famous example concerned the following sentence:

The box was in the pen.

in the following paragraph:

Little John was looking for his toy box. Finally, he found it. The box was in the pen. John was very happy.

Assume *"pen"* in English has the following two meanings: (1) a certain writing utensil, (2) an enclosure where small children can play. No existing method will enable a machine to determine that the word *"pen"* in the given sentence has the second of the above meanings, whereas, every reader with sufficient knowledge of English will do this automatically. Therefore, the issue is not the transition from one language to another, but rather a preliminary stage of this process; the determination of the specific meaning in *context* of a word

which, in isolation, is semantically ambiguous (Bar-Hillel, 1964).

Moreover, neural systems begin to move toward generating longer outputs or full documents in response to longer and more complicated inputs (structured meta-data) (Puduppully, Dong, & Lapata, 2019a; Lebret, Grangier, & Auli, 2016). However, there are still some open challenges for generating descriptive document-level summaries conditioned on the structured meta-data (e.g., table records) in terms of adequacy[1], coherency[2], and fidelity to the source material (Wiseman, Shieber, & Rush, 2017). The two main challenges which are mostly addressed in data-to-text generation techniques are: "What to say?", identifying the most important information from input data, and "How to say?", verbalizing data as a coherent document (Mei et al., 2016). These two challenges have been addressed separately as different modules in pipeline systems (McKeown, 1985; Reiter & Dale, 2000a) or in an end-to-end manner with SMT-like approaches (Mooney & Wong, 2007; Angeli, Liang, & Klein, 2010; Konstas & Lapata, 2013), or more recently, with Neural Language Generation models (Wiseman et al., 2017; Lebret et al., 2016; Mei et al., 2016). In spite of generating fluent text, end-to-end neural generation models perform weakly in terms of best content selection (Wiseman et al., 2017). Recently, (Puduppully, Dong, & Lapata, 2019a) trained an end-to-end data-to-document generation model on the Rotowire dataset (English summaries of basketball games with structured data)[3]. Although they aimed to overcome the shortcomings of end-to-end NLG models, they still need explicitly modelling for content selection and planning in their architecture.

The scarcity of parallel corpora is another major challenge for training high-quality NMT and NLG models (Koehn & Knowles, 2017). Transfer learning by fine-tuning from a model trained with a high-resource language-pair is a standard approach to tackle the scarcity of the data in the target low-resource language-pair (Dabre, Nakagawa, & Kazawa, 2017; Kocmi & Bojar, 2018; Saleh, Bérard, Calapodescu, & Besacier, 2019; Y. Kim, Gao, & Ney, 2019). However, this one-to-one approach is not able to exploit models trained for

[1] state of being sufficient for the purpose concerned
[2] having logical connection or consistency
[3] https://github.com/harvardnlp/boxscore-data

multiple high-resource language-pairs for the target language-pair of interest. Furthermore, models transferred from different high-resource language-pairs may have complementary syntactic and/or semantic knowledge, hence using a single model may be sub-optimal. Another appealing approach is multilingual NMT, whereby a single NMT model is trained by combining data from multiple high-resource and low-resource language-pairs (Johnson et al., 2017; Ha, Niehues, & Waibel, 2016; Neubig & Hu, 2018). However, the performance of a multilingual NMT model is highly dependent on the types of languages used to train the model. Indeed, if languages[4] are from very distant language families, they lead to negative transfer, causing low translation quality in the multilingual system compared to the counterparts trained on the individual language-pairs (Tan, Chen, et al., 2019; Oncevay et al., 2020).

1.1.1 Problems

In summary, we can name three inherent weaknesses of current NTG systems (NMT and LNG) by focusing the constraints on resources, when constraints come from either the inability of the model to exploit the full resources or general data scarcity in low-resource scenarios. These problems are listed as follows:

- **Problem 1:** Sentence-based text generation models suffer from a deficiency in coherency, adequacy, and fidelity to the source data.
- **Problem 2:** The scarcity of parallel corpora is a major challenge for training high-quality NMT/NLG models.
- **Problem 3:** The conventional transfer learning and multilingual learning are sub-optimal when a diverse set of languages are of interest.

[4]The languages studied in this thesis are: English, Kazakh, Belarusian, Bengali, Basque, Malay, Bosnian, Azerbaijani, Urdu, Tamli, Mongolian, Marathi, Galician, Kurdish, Estonian, Georgian, Bokmal, Hindi, Slovenian, Armenian, Burmese, Finnish, Macedonian, Lithuanian, Albanian, Danish, Swedish, Slovak, Indonesian, Thai, Czech, Ukrainian, Croatian, Greek, Serbian, Hungarian, Persian, German, Japanese, Vietnamese, Bulgarian, Polish, Romanian, Turkish, Dutch, Chinese, Spanish, Italian, Korean, Russian, Hebrew, French, Arabic, Portuguese.

1.2 Aims and Objectives

The main goal of this research is to improve the performance of resource-constrained NMT/NLG models using **Knowledge Transition (KT)** from high-resource NMT models to low-resource NMT/NLG models. Note that "Knowledge Transition" is an overarching term defined for the purpose of analysing "Transfer Learning" and "Knowledge Distillation" jointly. In both transfer learning and knowledge distillation approaches, we aim to transit knowledge from an expert model with higher knowledge capacity, usually trained in a high-resource setting, to another model that needs to be trained in a low-resource scenario. In transfer learning, the expert model's parameters are often used to initialize the new model from the same or related task. This approach is practical when the source and target tasks have similar model components in the knowledge transition scheme. In knowledge distillation, however, the predictions of the expert model are used in training the new model where the source and target tasks are the same for both models, but no sharing of model components is required. Motivated by the gaps in the literature mentioned in the previous section, this thesis aim at the following objectives:

(i) Capturing document-wide interdependencies and generating faithful and fluent documents from limited structured meta-data using ***Transfer Learning*** from document-level NMT to data-to-document generation task.

(ii) Improving NMT in bilingually low-resource scenarios by incorporating knowledge from high-resource pretrained NMT models in the low-resource ones using ***Adaptive Knowledge Distillation***.

(iii) Improving low-resource NMT performance in a multilingual setting while avoiding negative transfer using ***Hierarchical Knowledge Distillation***.

To achieve the above objectives of this research, we present the following three knowledge transition schemes in low-resource scenarios:

(i) Knowledge transition from a high-resource task (NMT) to a different low-resource

task (NLG).

(ii) Knowledge transition from an ensemble of high-resource models to a low-resource model from the same task (MT) in a bilingual setting regardless of the relatedness/similarity of contributed languages.

(iii) Knowledge transition from the groups of bilingual NMT models to a single multilingual model, considering the language similarity in a hierarchy structure.

1.3 Contributions

The major contribution of this thesis is three-fold, each of which achieves one of the three objectives through one of the knowledge transition schemes mentioned above. The following paragraphs move on to describe the contributions in detail.

1.3.1 Knowledge transition from NMT to NLG

We propose a novel approach to tackle the data scarcity problem in NLG by transitioning knowledge from the NMT to the NLG task. This approach relies on the following intuition: Machine translation and natural language generation are inherently the two sides of one problem. One can define the data-to-text generation task as a type of translation task which has abstract meta-data in the source side rather than a complete document. Considering this fact, we suggest leveraging the data from both NMT and NLG tasks with transfer learning. As both tasks have the same target (e.g., English-language stories), they can share the same decoder. The same encoder can also be used for NLG and MT if the NLG metadata is encoded as a text sequence. In this contribution, we first train domain-adapted document-level NMT models on large amounts of parallel data. Then, we fine-tune these models on small amounts of NLG data, transitioning from NMT to NLG. To the best of our knowledge, this is the first work which suggests separate data selection and data ordering steps are not necessary in data-to-text generation if NLG model is transferred from a document-level translation model and is given all meta-data as a document sequence. We propose a com-

pact way to encode the data available in the original dataset and enrich it with some extra facts that can be inferred with minimal knowledge of the task. We also show that an NLG model trained with this data is able to capture document-level structure and select and order information automatically. This is in contrast to existing methods that typically do explicit modelling for content selection and planning in their architectures, thus perform weakly in terms of best content selection, specially for the facts which are not explicitly present in the meta-data but can be inferred with a good language model learned from the machine translation task. Finally, our NLG models, bootstrapped from the NMT models, do fluent and coherent text generation and are even able to infer some facts that are not explicitly encoded in the structured data and outperform the previous state of the art on the Rotowire NLG dataset [5]. Meanwhile, our systems submitted to WNGT 2019 obtained the best results on each of the 6 tasks (Hayashi et al., 2019).

This work has been published and orally presented in Proceedings of the 3rd Workshop on Neural Generation and Translation (Saleh et al., 2019) and submitted as an industrial patent. This work is done during the 6-month internship program in NAVER LABS Europe[6], France. The detail of this approach is presented in Chapter 3.

1.3.2 Adaptive Knowledge Distillation

We propose a novel approach to tackle the data scarcity problem in NMT using adaptive knowledge distillation, in which we suggest distilling the knowledge of an ensemble of high-resource teacher models to a single student model. What distinguishes our approach from the previous distillation-based methods is the way we choose the best teachers statistically based on the data and knowledge gap of the student model, rather than deterministically (Tan, Ren, et al., 2019). To the best of our knowledge, this is the first approach that addresses the inefficiency of the original transfer learning by making a wiser use of high-resource languages and pre-trained models in an effective collaborative learning manner. We firstly propose a many-to-one transfer learning approach which can effectively transfer

[5] https://github.com/harvardnlp/boxscore-data
[6] https://europe.naverlabs.com/

knowledge from multiple high-resource language pairs to a target low-resource language-pair of interest. As the quality of these teacher models varies, we then propose an effective Adaptive Knowledge Distillation (AKD) approach to dynamically adjust the contribution of the teacher models during the distillation process. This idea is derived from the intuition that models transferred from different high-resource language pairs may have complementary syntactic and/or semantic strengths on the target low-resource language. Furthermore, it is not generally clear which high-resource language-pair offers the best transfer learning for the target NMT setting in every mini-batch of data. In our AKD approach, the label smoothing coming from different teachers is combined and regulated based on the loss incurred by the teacher models during the distillation process, and thus the contribution of each teacher is changed based on its effectiveness of improving the student. Experiments on transitioning knowledge from a collection of six language pairs from IWSLT (Cettolo, Niehues, Stüker, Bentivogli, & Federico, 2014) to five low-resource language pairs from TED dataset (Qi, Sachan, Felix, Padmanabhan, & Neubig, 2018) demonstrate the effectiveness of our approach, achieving up to +0.9 BLEU score improvement compared to solid baselines.

This work has been published and orally presented in the Proceeding of The 28th International Conference of Computational Linguistics (COLING2020) (Saleh et al., 2020). The detail of this approach is presented in Chapter 4.

1.3.3 Hierarchical Knowledge Distillation

We improve and extend our previous contribution by focusing on optimized language transfer when transferring knowledge from multiple languages to a single multilingual student. We propose hierarchical knowledge distillation (HKD), where the hierarchy represents the similarity structure and relatedness of the languages based on different typological views. This is then reflected into the NMT models corresponding to the *language groups*, i.e. the nodes of the structure. Our main intuition is that distilling the knowledge of a *diverse* set of teacher models into a student model may be suboptimal, as the teachers may compete

instead of collaborate, resulting in a phenomenon called *negative transfer*, typically studied in multitask/transfer learning (Zoph, Yuret, May, & Knight, 2016). However, the proposed HKD process leads to training a higher-quality multilingual translation model by leveraging common characteristics of languages belonging to the same language group. Our approach to preventing negative transfer in knowledge distillation is clustering languages based on their typological similarities and then distilling their related NMT models' knowledge to group-specific teacher assistant models. This process is repeated while different clusters are merged down to a single super-group. To the best of our knowledge, this is the first work which considers typological language relations by distilling knowledge hierarchically in a multilingual regime. In this approach, we not only take advantage of multilingual learning by utilizing the training examples of multiple languages to improve the translation of low-resource language pairs, but also we avoid negative transfer by effectively capturing the language families relationships. Experiments on 53 languages from the TED dataset (Qi et al., 2018) demonstrates the effectiveness of our approach in mitigating the negative transfer in a multilingual translation setting. This work has been published in the 2021 Conference on Empirical Methods in Natural Language Processing (Saleh, Buntine, Haffari, & Du, 2021). We study and analyse this approach in chapter 5.

1.4 Thesis Outline

The remainder of this thesis is organised as follows:

Chapter 2 - Background.

This chapter provides a thorough overview of the methodologies and algorithms adopted for the research described in this thesis. We start with the technical background related to text generation systems such as NMT and NLG systems in general sequence to sequence framework. We further introduce the two important techniques widely used in this thesis, transfer learning and knowledge distillation.

Chapter 3 - From Machine Translation to Text Generation.

This chapter is based on the following paper:

> *Fahimeh Saleh, Alexandre Berard, Ioan Calapodescu, and Laurent Besacier. "Naver Labs Europe's Systems for the Document-Level Generation and Translation Task at WNGT 2019." In Proceedings of the 3rd Workshop on Neural Generation and Translation, pp. 273-279. 2019.*

In this chapter, we introduce our novel techniques for transitioning from NMT to NLG. We propose to leverage data from both NMT and NLG and do transfer learning between NMT, NLG, and NMT with source-side metadata. First, we train a document-based NMT system with the DGT parallel data. Then, we augment this NMT model to obtain a "Data + Text to Text" model. Finally, we remove the source text to get a pure NLG system, able to translate from metadata to full documents.

Chapter 4 - **Improving Low-resource NMT using Adaptive Knowledge Distillation.**
This chapter is based on the following paper:

> *Fahimeh Saleh, Wray Buntine, and Gholamreza Haffari. "Collective Wisdom: Improving Low-resource Neural Machine Translation using Adaptive Knowledge Distillation." In Proceedings of the 28th International Conference on Computational Linguistics, pp. 3413-3421. 2020.*

This chapter introduces our novel adaptive knowledge distillation approach which improves low-resource NMT. In this chapter, we tackle the data scarcity problem in NMT using knowledge distillation, where we propose to distill the knowledge of *ensemble of teacher* models to a single *student* model. As the quality of these teacher models varies, we propose an effective adaptive knowledge distillation approach to dynamically adjust the contribution of the teacher models during the distillation process.

Chapter 5 - **Multilingual NMT with Hierarchical Knowledge Distillation.**
This chapter is based on the following paper:

Fahimeh Saleh, Wray Buntine, Gholamreza Haffari, and Lan Du . "Multilingual Neural Machine Translation: Can Linguistic Hierarchies Help?" In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing.

In this chapter, we generalize our previous contribution to a multilingual setting, focusing on optimized language transfer. We propose a Hierarchical Knowledge Distillation (HKD) approach for MNMT which capitalises on language groups generated according to typological features and phylogeny of languages to overcome the issue of negative transfer. HKD generates a set of multilingual teacher-assistant models via a selective knowledge distillation mechanism based on the language groups, and then distills the ultimate multilingual model from those assistants in an adaptive way.

Chapter 6 - **Conclusion.** In this chapter, we summarise the main contributions of this thesis and discusses the most promising future research directions stemming from this research.

2 | Background

In this chapter, we introduce the machine learning techniques, architectures, and methods that have been used in this thesis. In particular, we review the neural text generation systems and sequence to sequence framework with a focus on transformer architecture which has been widely used in this thesis for both machine translation and text generation purposes. We also elaborate on different properties of sentence-level, document-level, and multilingual NMT. Later, we have an overview on transfer learning and knowledge distillation methods to help the readers better understand the following chapters.

2.1 Text Generation

Text Generation is the process of generating descriptive linguistic text or speech in one of the human's languages from text or non-linguistic input (Gatt & Krahmer, 2018). The Text Generation approaches can be categorized into two groups based on the type of the input: i) **text-to-text generation**, when the input is text; ii) **data-to-text generation**, when the input is non-linguistic representation of information (Reiter, Mellish, & Levine, 1995; Reiter & Dale, 1997, 2000b).

Text-to-text generation:

Text-to-text generation approaches take existing text as their input and automatically generate a coherent text as output. Some example applications of text-to-text generation include: machine translation (Delavenay & Delavenay, 1960; Och & Ney, 2003; Koehn, 2009), summarization (generating a concise text from an existing text) (Clarke & Lapata, 2010), simplification (generating a simple format of text from an existing complex text) (Siddharthan, 2014; Siddharthan & MacDonald, 2016), automatic spelling, grammar and text correction (Kukich, 1992; Dale, Anisimoff, & Narroway, 2012), automatic peer reviews generation (Bartoli, De Lorenzo, Medvet, & Tarlao, 2016), paraphrase generation (Bannard & Callison-Burch, 2005; Kauchak & Barzilay, 2006), and question generation (J. Brown, Frishkoff, & Eskenazi, 2005).

In particular, Machine Translation (MT) is the task of automatically translating the text in one natural language (source language) to text in another language (target language) (Russell & Norvig, 2010). The task of machine translation has been done for a long time (1950s-1980s) in a rule-based manner called rule-based machine translation (RBMT) which was based on linguistic information about the source and target languages basically retrieved from dictionaries and grammars (Toma, 1977). Later, with the advances in statistics, Statistical machine translation (SMT) has been introduced in which translations are generated by applying statistical models based on the analysis of bilingual corpus (Koehn, 2009; P. F. Brown et al., 1990). SMT is a data-driven approach and it is no longer required to specify the rules of translation as in RBMT. However, these traditional phrase-based SMT systems typically consist of many small sub-components that are tuned separately. This limitation has been addressed by introducing Neural machine translation (NMT) which was an attempt to build and train a single, end-to-end model using neural networks to predict the likelihood of a sequence of words, typically modeling entire sentences in a single integrated model (Cho et al., 2014; Sutskever, Vinyals, & Le, 2014).

Although the main research on NMT started with bilingual translation systems, which generally build a translation model between two languages, the NMT framework can also

leverage multiple languages in a translation process (Y. Chen, Liu, Cheng, & Li, 2017; Cheng, 2019; Y. Chen, Liu, & Li, 2018; Dong, Wu, He, Yu, & Wang, 2015; Firat, Cho, Sankaran, Vural, & Bengio, 2017; Dabre, Chu, & Kunchukuttan, 2020). In this thesis, we refer to the NMT system which incorporates more than one language pair in the translation process as Multilingual NMT (MNMT). The main aim of the MNMT system is doing the translation of multiple languages with a single model, which is effectively efficient considering the fact that training separate models for different language pairs is resource consuming while there are thousands of languages in the world (Campbell, 2008).

In an MNMT system, the translation direction can be from one-to-many (Dong et al., 2015), many-to-one (Lee, Cho, & Hofmann, 2017), or many-to-many (Firat et al., 2017) using parallel corpora for multiple language pairs. For bilingually low-resource language pairs, MNMT provides additional parallel training signals from the high-resource language pairs to improve the low-resource ones. In fact, representing multiple languages through the same vector space in the MNMT system provides a transfer learning (Pan & Yang, 2009) to utilize data from high-resource language pairs to improve the translation of low-resource language pairs.

The majority of works on MNMT mainly focus on different parameter sharing mechanisms for designing the MNMT models (Firat et al., 2017; Lu et al., 2018; Johnson et al., 2017; Ha, Niehues, & Waibel, n.d.). Different architectural choices are basically defined by the degree of parameter sharing among various supported languages, e.g., minimal parameter sharing by sharing either encoder, decoder or attention module (Firat et al., 2017; Lu et al., 2018) or complete parameter sharing by sharing the entire models (Johnson et al., 2017; Ha et al., n.d.). Both of these two groups of techniques generally focus on designing better parameter sharing for multilingual translation by implicitly assuming that a set of languages are pre-given without considering the effect of language transfer between the languages shared in one model. That is why they generally achieved comparable results with individual models (trained with individual language pairs) only when the languages are less diverse, or from the same language family (Littell et al., 2017b), or when the number of lan-

guages is small. When dozens or hundreds of diverse language pairs are taken to account for training a massive MNMT system, the *negative language transfer*[1] usually happens between more distant languages, and the translation accuracy of the multilingual model downgrades (Oncevay et al., 2020). It is indeed challenging to train a multilingual translation model supporting many diverse language pairs while achieving comparable accuracy to individual models. In the 4^{th} and 5^{th} chapters of this thesis, we focus exclusively on the former type of text generation, machine translation. In the 5^{th} chapter of this thesis we propose and develop an approach to overcome the shortcoming of MNMT.

Data-to-text generation:

Data-to-text generation approaches, automatically interpret, organize, and generate human-readable text from a non-linguistic meta-data. Basically, the process involves an algorithm that scans large amounts of provided data (tables, images, etc.), selects and orders the critical information, inserts details (e.g., names, places, numbers, statistics, etc.), and generates a readable, coherent, and descriptive text. The output can also be customized according to a specific voice, tone, or style (N. S. Cohen, 2015). "Automated journalism" or "Robot journalism" (Montal & Reich, 2017), is the most important example application of such approaches which have had a remarkable effect on the field of journalism in terms of efficiency and cost-cutting (Van Dalen, 2012; Clerwall, 2014; Young & Hermida, 2015). Some developed applications which generate text from meta-data, to name but a few, include: soccer reports generation (Theune, Klabbers, de Pijper, Krahmer, & Odijk, 2001; D. L. Chen & Mooney, 2008), virtual newspaper generation (Molina, Stent, & Parodi, 2011; Leppänen, Munezero, Granroth-Wilding, & Toivonen, 2017), weather and financial report generation (Goldberg, Driedger, & Kittredge, 1994; Turner, Sripada, Reiter, & Davy, 2007; Plachouras et al., 2016), and clinical patient information generation (Kraus, 2003; Banaee, Ahmed, & Loutfi, 2013).

[1]Negative transfer happens when differences between the two languages' structures cause systematic errors in learning the other language. Positive transfer transpires when the similarity between the two languages promotes learning the other language.

In the third chapter of this thesis, we propose and develop a text generation model with the type of data-to-text generation. In this approach, we generate the descriptive basketball games' reports from structured meta-data (table records). In the following, we will dive into details of text generation systems by introducing the Encoder-Decoder structure of sequence-to-sequence systems, the training objectives, attention mechanisms, and the two important architectures including Recurrent Neural Networks (RNNs) and Transformers.

2.1.1 Encoder-Decoder Structure

As mentioned, in text generation problem, we process an input sequence to generate an output sequence also named as sequence-to-sequence (seq2seq) problem. The encoder-decoder is the standard modeling paradigm to address this problem (Cho et al., 2014; Sutskever et al., 2014). Seq2seq models consist of two neural networks that are trained jointly to map an input sequence to an output sequence. As shown in Figure 2.1, the first neural network is the encoder which reads the sequence of source symbol representations, $\mathrm{x} = (x_1, x_2, ..., x_n)$, and as a result encodes a fixed compact representation c trying to summarize all of its information. Then, this representation vector acts as an input or initial state to the decoder. The decoder then predicts the probability of the target sequence, conditioned on the source sequence. In fact, at each time step, the decoder generates a symbol of the target sequence as $\mathrm{y} = (y_1, y_2, ..., y_m)$ based on the input received and its current state, as well as updating its own state for the next time step. Later in this section, we will introduce Recurrent Neural Networks and Transformers as the two well-known architectures for Encoder-Decoder pipeline.

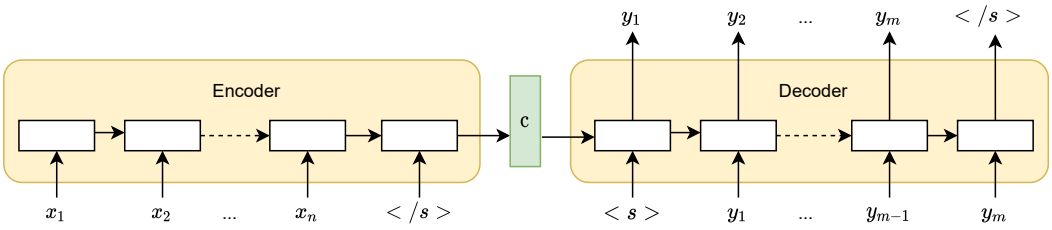

Figure 2.1: A general overview of an encoder-decoder model for sentence-level NMT.

2.1.2 Recurrent Neural Networks

Here, we describe briefly one of the well-known architectures, called Recurrent Neural Networks (RNNs) for the Encoder-Decoder pipeline, proposed by (Cho et al., 2014; Sutskever et al., 2014). As mentioned in the previous part, in the Encoder–Decoder framework, an encoder reads the input sequence, $\mathrm{x} = (x_1, x_2, ..., x_n)$ into a representation vector c. The most common approach is to use an RNN, in which at each time step t, the hidden state h_t of that RNN is updated by:

$$h_t = f(x_t, h_{t-1}) \tag{2.1}$$

where f is a non-linear activation function which can be a simple logistic sigmoid function or a complex function such as long short-term memory (LSTM) unit (Hochreiter & Schmidhuber, 1997). In the Encoder-Decoder pipeline, the encoder is an RNN that reads each element (symbol) of an input sequence x sequentially. By reading each symbol, the hidden state of the RNN changes according to Eq. 2.1. After reading the end of the sequence (marked by an end-of-sequence symbol, *e.g.*, $< /s >$), the hidden state of the RNN is a compact representation c of the entire input sequence.

The decoder is another RNN which is trained to generate the output sequence by predicting the next symbol y_t given the hidden state h_t. However, here both y_t and h_t are also conditioned on y_{t-1} and on the compact representation c of the input sequence. Hence, the hidden state of the decoder at time t is computed as:

$$h_t = f(h_{t-1}, y_{t-1}, \mathrm{c}) \tag{2.2}$$

and thus the conditional distribution of the next symbol is:

$$P(y_t|y_{t-1}, y_{t-2}, ..., y_1, \mathrm{c}) = g(h_t, y_{t-1}, \mathrm{c}) \tag{2.3}$$

where g is a nonlinear, potentially multi-layered, function that can generate valid outputs as

the probabilities.

2.1.3 Training Objective

Given a training set $\mathcal{D}$ containing N parallel source-target sequence pairs, (x, y), the goal of a seq2seq system is to model the probability of a target sequence given a source sequence, $p(\mathrm{y}|\mathrm{x})$. All parameters in the encoder-decoder architectures are jointly trained via back propagation (LeCun, 1988; Rumelhart, Hinton, & Williams, 1986) to minimize the negative log-likelihood (conditional) over the training set. The conditional log-likelihood is defined as the sum of the log-probability of predicting a correct symbol y_n in the output sequence for each instance x in the training set. Thus, we want to find the optimum set of model parameters θ^* as follows:

$$\begin{aligned} \theta^* &= \underset{\theta}{\operatorname{argmin}} \sum_{(\mathrm{x},\mathrm{y})\in\mathcal{D}} -\log p_\theta(\mathrm{y}|\mathrm{x}) \\ &= \underset{\theta}{\operatorname{argmin}} \sum_{(\mathrm{x},\mathrm{y})\in\mathcal{D}} \sum_{i=1}^{m} -\log p_\theta(y_i|y_{<i}, \mathrm{x}) \end{aligned} \tag{2.4}$$

2.1.4 Inference: Greedy Decoding and Beam Search

After training a seq2seq model, e.g., an NMT model, we need to use it to translate or decode unseen source sentences. Indeed, the objective of the decoding phase is finding the highest probability translation for a given source sentence which can be formulated as follow:

$$\hat{\boldsymbol{y}} = \arg\max_{\boldsymbol{y}} P_{\boldsymbol{\theta}}(\boldsymbol{y}|\boldsymbol{x}) \tag{2.5}$$

Where $\boldsymbol{y}$ is the space of possible translations for the source sentence $\boldsymbol{x}$. This is an intractable searching problem as the space of possible translations is exponentially large with respect to the output length $|\boldsymbol{y}|$ and solving this optimisation problem is computationally complex (Hoang, Haffari, & Cohn, 2017). An approximate solution to this problem can

be obtained using conventional heuristic-based searching strategies like greedy decoding or beam search.

The idea of greedy decoding is to pick the word that has the highest probability (i.e. act greedily) at each decoding step until the end-of-sentence token is generated. Choosing the best word might be advantageous for one timestep, but it may be sub-optimal when it comes to the whole sentence. Beam search (Graves, 2012), on the other hand, selects multiple (varied based on the tunable parameter of beam width) translation hypotheses with the highest log-probability at each timestep based on the conditional probability. A complete hypothesis (containing the end-of-sentence token) is added to the final candidate list. The algorithm then picks the translation with the highest log-probability (normalised by the number of target words) from this list (Maruf et al., 2021). Y. Chen, Li, et al. (2018) argued that the translation quality obtained via beam search algorithm with the beam-width of 4 is remarkably better than the translation obtained via greedy decoding. Nevertheless, beam search is computationally much more expensive than greedy decoding depending on the number of words to keep in memory at each step to permute the possibilities.

2.1.5 Attention Mechanism

In the Encoder-Decoder structure, the encoder compresses the entire source sequence into a single fixed representation vector c. For the encoder, the task of compressing all the information of the source sequence into a single vector is challenging as there is a high possibility that some of the information is forgotten, especially when the length of the source sequence is relatively long. Moreover, during the generation process in the decoder, at each generation step, some parts of the source sequence can be more relevant than others. However, with the current setting, the decoder has to extract relevant information for generating all of the elements of the target sequence from the same single representation. By introducing the "Attention Mechanism in NMT" (Bahdanau, Cho, & Bengio, 2014), this problem of fixed-length vector started to be solved by allowing a model to automatically search for parts of a source sequence that are relevant to predicting a target element. In this mechanism, instead

of relying on a fixed-length vector, a context vector is used for predicting each element of the target sequence. This weighted context vector is obtained dynamically by applying a content-based attention mechanism over the source sequence.

In order to formulate the attention mechanism, consider the source sequence $\mathrm{x} = [x_1, x_2, ..., x_n]$ of length n and we try to output a target sequence $\mathrm{y} = [y_1, y_2, ..., y_m]$ of length m. Also, assume that the encoder is a bidirectional RNN. Bidirectional RNN is two independent RNNs in which the input sequence is fed in positive time order (forward) for one network and in reverse time order (backward) for another. Then, the outputs of the two networks are usually concatenated at each time step to form the final output. We show the forward hidden state $\overrightarrow{h_i}$, and a backward one $\overleftarrow{h_i}$ and the concatenation of two hidden states represents the encoder state as,

$$h_i = [\overrightarrow{h_i}^\top, \overleftarrow{h_i}^\top]^\top, i = 1, 2, ..., n \tag{2.6}$$

Also, assume that the decoder network has hidden state $s_t = f(s_{t-1}, y_{t-1}, \mathrm{c}_t)$ for the output word at position t, $t = 1, \ldots, m$, where the context vector c_t is the weighted sum of hidden states of the input sequence as follows:

$$c_t = \sum_{i=1}^{n} \alpha_{t,i} h_i \tag{2.7}$$

where $\alpha_{t,i}$ indicates how well the inputs around position i and the output at position t match.

$$\alpha_{t,i} = \frac{\exp(score(s_{t-1}, h_i))}{\sum_{k=1}^{n} \exp(score(s_{t-1}, h_k))} \tag{2.8}$$

In particular, in (Bahdanau et al., 2014), the alignment model (score) is parametrized as a feed-forward neural network which is jointly trained with all the other components of the proposed system. Therefore, given that $tanh$ is used as the non-linear activation function,

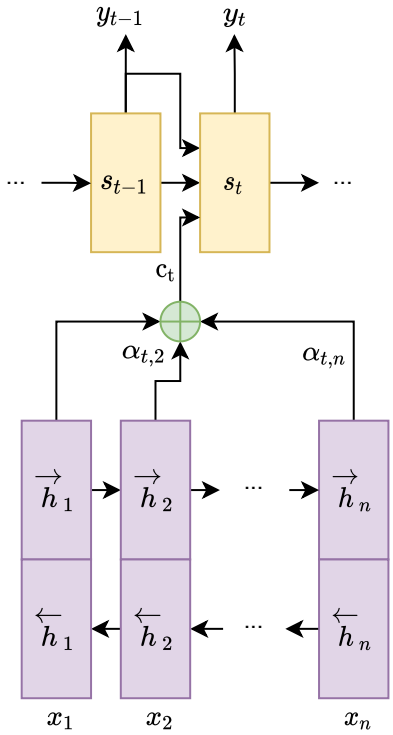

Figure 2.2: Attentional RNN-based architecture

the score function is as follows:

$$score(s_t, h_i) = \mathrm{v}_a^\top tanh(\mathrm{W}_a[s_t; h_i]) \tag{2.9}$$

where v_a and W_a are weight matrices to be learned in the alignment model. Note that the score function can be as simple as a dot product as well.

In summary, the weights $\alpha_{t,i}$, reflects the importance of the hidden state h_i corresponding to the word x_i with respect to the previous hidden state s_{t-1} of the target sequence in deciding the next state s_t and generating y_t. Intuitively, this is an attention mechanism in which the decoder decides about those parts of the source sentence to pay attention to.

2.1.6 Transformers

As mentioned in the previous section, the attention mechanism helped to improve the performance of the RNN encoder-decoder neural seq2seq system by making use of a weighted sum of all the past encoder states, which allows the decoder to assign more importance to certain elements of the input for generating each element of the output. However, this approach continues to have an important limitation as each sequence must be treated one element at a time. The fundamental constraint of sequential computation in RNN-based encoder-decoder approaches forces both the encoder and the decoder to wait for the completion of $t-1$ steps to process the t^{th} step, leading to a time-consuming and computationally inefficient process, especially when dealing with a huge corpus. To address this problem, Transformer architecture (Vaswani et al., 2017b) has been proposed that is relying entirely on self-attention mechanism, eliminates the need for recurrent network units in sequence to sequence modelling, leading to a significantly more parallelization.

Multi-Head Self-Attention

The main component in the transformer architecture is the multi-head self-attention unit illustrated in Figure 2.3. The attention mechanism maps a query and a set of key-value pairs to an output, where the query, keys, values, and output are all vectors. In particular, both the keys and values with dimension n (input sequence length) are the encoder's hidden states. In the decoder, the previous output is compressed into a query (of dimension m) and the next output is computed as a weighted sum of the values, where the weight assigned to each value is computed by a compatibility function such as dot-product of the query with the corresponding key computed as follows:

$$Attention(Q, K, V) = softmax(\frac{QK^\top}{\sqrt{n}})V \tag{2.10}$$

Moreover, in the transformer architecture, to let the model focus on different things, instead of using a single attention function, multi-head attention is utilized which contains sev-

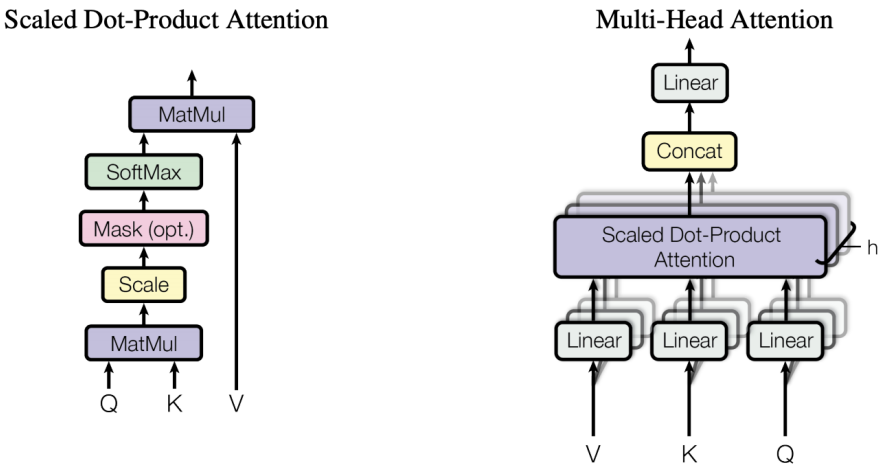

Figure 2.3: (left) Scaled Dot-Product Attention. (right) Multi-Head Attention consists of several attention layers running in parallel (Vaswani et al., 2017a)

eral "heads" working independently whose results are concatenated at the end. Multi-head attention allows the model to jointly attend to information from different representation subspaces at different positions. In particular, in transformer architecture, we linearly project the queries, keys and values h times with different, learned linear projections where the projections are parameter matrices as follows,

$$MultiHead(Q, K, V) = Concat(head_1, head_2, ..., head_h)W^O \tag{2.11}$$

where,

$$head_i = Attention(QW_i^Q, K_i^K, V_i^V) \tag{2.12}$$

Model Architecture

Given the information provided about the main component of the Transformer, in this part, we will look into the whole architecture of the transformer. The Transformer-based NMT is based on the encoder-decoder structure we have discussed in Section 2.1.1 which a different internal architecture. As mentioned before, the main innovation of the transformer architec-

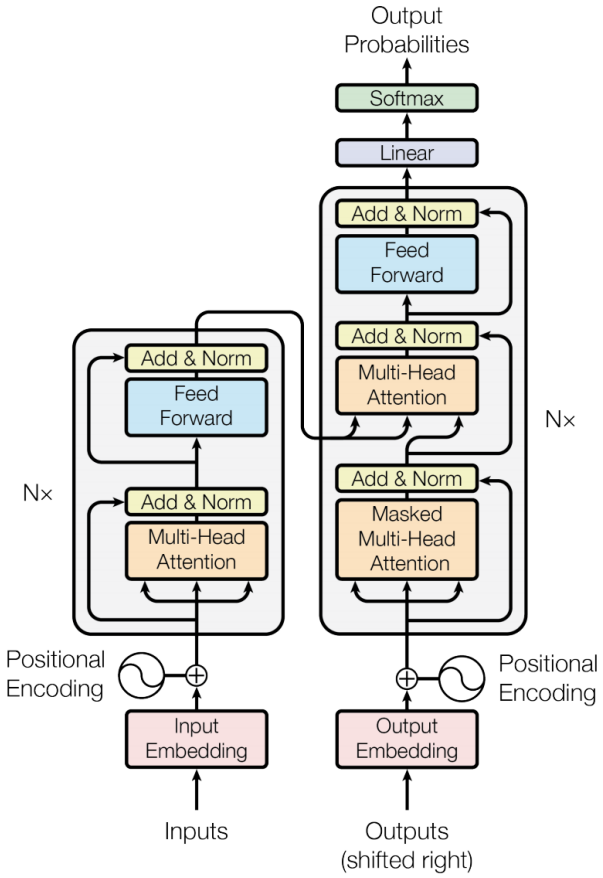

Figure 2.4: Transformer architecture (Vaswani et al., 2017a)

ture is the self-attention layer in both the encoder and the decoder. This attention mechanism is in addition to the standard attention in which the decoder attends to the encoder context vector. In the self-attention encoder layers, every word in the source sentence attends to every other word in that sentence, and the resulting attention vector is used as the representation of that word in that layer. The attention in the decoder works similarly, but with a masking mechanism to prevent the model from using words that have not yet been generated to generate the current word. As illustrated in Figure 2.4, the encoder is composed of a stack of N (originally $N = 6$) identical layers. Each layer has a multi-head self-attention layer and a simple position-wise fully connected feed-forward network. The output of each sub-layer follows the form of $LayerNorm(x + Sublayer(x))$ which is obtained by utilizing a residual connection (He, Zhang, Ren, & Sun, 2016) around each of the two sub-layers,

followed by layer normalization (J. L. Ba, Kiros, & Hinton, 2016). Note that all the sub-layers as well as the embedding layers output data of the same dimension $d_{model} = 512$. These layers are stacked on top of each other to encode the final representation of the source sentence.

The transformer decoder has a similar structure as the encoder but with an additional sub-layer which performs multi-head attention over the output of the encoder stack. Moreover, in the decoder architecture, the self-attention sub-layer is modified to prevent positions from attending to subsequent positions to ensures that the predictions for position i can depend only on the known outputs at positions less than i and not on the future.

Moreover and similar to other sequence modeling problems, the learned embeddings are utilized to convert the input tokens and output tokens to vectors of the same dimension $d_{model} = 512$. To preserve the position information, a sinusoid-wave-based positional encoding is applied and summed with the embedding output. Finally, a linear layer and a softmax function are added to the final decoder output to predict the next-token probabilities.

2.1.7 Evaluation

Several automatic evaluation metrics have been proposed so far to evaluate the quality of the machine translation and data-to-text generation models by comparing system outputs to reference text. Here, we mention two of the most popular n-gram matching metrics, BLEU and METEOR as they are the most relevant for the purposes of this thesis. At the end of this section, we also explain the "perplexity" measure which is the best evaluation metric for evaluating language models [2]

BLEU (Bilingual Evaluation Understudy). This metric is one of the most popular evaluation measures for machine translation and data-to-text generation tasks proposed by (Papineni, Roukos, Ward, & Zhu, 2002a) in 2002 which ranges from 0 to 1 (1 means an identical translation with the reference). The main idea behind this measure is to aggregate the count

[2]The interested readers can get an in-depth review of MT evaluation from (Chatzikoumi, 2020) and can find an empirical comparison from (Zhang*, Kishore*, Wu*, Weinberger, & Artzi, 2020).

of words and phrases (n-grams) that overlap between machine and reference texts based on a modified precision for n-grams. The modified n-gram precision for a candidate text generated by the model is calculated as a geometric average:

$$p_n = \frac{\sum_{\text{n-gram} \in \{\text{candidate}-\text{generated}-\text{text}\}} Count_{clip}(\text{n-gram})}{\sum_{\text{n-gram}' \in \{\text{candidate}-\text{generated}-\text{text}\}} Count(\text{n-gram}')} \tag{2.13}$$

where $Count(\text{n-gram})$ is is the number of mutual n-grams in a machine generated text and reference. $Count_{clip}(\text{n-gram})$ is the number of mutual n-grams clipped by the maximum repetition of n-grams in the reference. For a generated text with the length t and the reference of length r, the BLEU score is formulated as follows:

$$\text{BLEU} = \text{BP} \exp\left(\sum_{n=1}^{N} w_n \log p_n\right) \tag{2.14}$$

where N is the maximum length of n-grams (usually up to 4), w_n is weight for the modified n-gram precision and usually is uniform. BP is the brevity penalty used to penalise outputs longer than references and is defined as:

$$\text{BP} = \begin{cases} 1, & \text{if } t > r. \\ \exp^{(1-r/c)}, & \text{if } t \leq r. \end{cases} \tag{2.15}$$

Although BLEU has a high correlation with human judgment (Papineni et al., 2002a), it relies on precision alone and does not consider recall. METEOR then is proposed to address this drawback.

METEOR (Metric for Evaluation for Translation with Explicit Ordering). This metric is proposed by (Banerjee & Lavie, 2005; Lavie & Agarwal, 2007) and measures the sentence-level similarity by explicitly performing a word-to-word alignment between the generated text and a given reference.

The alignments are produced via a sequence of word matchers modules such as: Exact (identical words), Stem (identical stem using Porter stemmer), Synonym (the words are synonyms of each other). After the final alignment, METEOR computes an F-score:

$$F_{mean} = \frac{P \cdot R}{\alpha \cdot P + (1 - \alpha) \cdot R} \tag{2.16}$$

Which is the parameterised harmonic mean of unigram precision (R) and recall (R) (Rijsbergen, 1979). After obtaining the final alignment, the METEOR Score is calculated as follows:

$$\text{METEOR} = (1 - \text{Penalty}) \cdot F_{mean} \tag{2.17}$$

Accordingly, METEOR penalises the generated text based on the order of matched words in the output and reference. The $\text{Penalty} \propto (\frac{\text{chunks}}{m})$, where chunks is the smallest number of matched words, such that the matched words in each chunk are adjacent and in the same word order (in the output and reference), and m is the number of matched words.

PPL (Perplexity). Unlike the BLEU and METEOR which evaluate the models by employing them in an actual task (MT or NLG), PPL evaluates the language model itself without taking to the account the specific task. A language model is a probability distribution over entire sentences or texts and PPL is calculated based on the inverse *probability* of the translation sentence, normalised by the number of words:

$$\text{PPL}(y) = \sqrt[T]{\frac{1}{\Pi_{i=1}^{T} p(y_i | y_{<i}, x)}} \tag{2.18}$$

So, intuitively, PPL measures the confidence of the model regarding the estimated distribution. A low perplexity indicates the probability distribution is good at predicting the sample (Neubig, 2017).

2.1.8 Document-level NMT

Most of the MT models including the phrase-based models or the most advanced NMT models process sentences in isolation and ignore extra-sentential information from the context. However, from the linguistics point of view, an extended context can prevent mistakes in vague cases and improve the translation coherence (Maruf et al., 2021). To overcome this shortcoming of sentence-based NMT, a lot of works recently have been undertaken in the space of document-level NMT. In this section we have a brief overview of the problem definition, training, and decoding of document-level NMT[3].

Problem definition. Given a document $\boldsymbol{d}$, where the set of sentences in source language are $\boldsymbol{X} = \{\boldsymbol{x}^1, \ldots, \boldsymbol{x}^{|\boldsymbol{d}|}\}$, the goal of document-level MT is to generate the set of translations in the target language $\boldsymbol{Y} = \{\boldsymbol{y}^1, \ldots, \boldsymbol{y}^{|\boldsymbol{d}|}\}$. The probability of a document translation given the source document using the chain rule can be formulated as follows:

$$P_{\boldsymbol{\theta}}(\boldsymbol{Y}|\boldsymbol{X}) \propto \exp \sum_{j=1}^{|\boldsymbol{d}|} \log P_{\boldsymbol{\theta}}(\boldsymbol{y}^j|\boldsymbol{x}^j, \boldsymbol{Y}^{-j}, \boldsymbol{X}^{-j}) \tag{2.19}$$

where $\boldsymbol{Y}^{-j}$ represents *all* the other sentences in the target document except the j^{th} sentence. In the sentence-level NMT, the conditional probability of a target sentence $\boldsymbol{y}$ given the source sentence $\boldsymbol{x}$ is decomposed as:

$$P_{\boldsymbol{\theta}}(\boldsymbol{y} \mid \boldsymbol{x}) = \prod_{n=1}^{N} P_{\boldsymbol{\theta}}(y_n \mid \boldsymbol{y}_{<n}, \boldsymbol{x}) \tag{2.20}$$

where y_n is the current target word and $\boldsymbol{y}_{<n}$ are the previously generated words.

Training. Given a normalised document MT model, the training objective minimise the

[3]We direct interested readers to read the "A Survey on Document-level Neural Machine Translation: Methods and Evaluation" by (Maruf et al., 2021) which covers this topic in detail.

negative log-likelihood over the set of bilingual training documents ($\mathcal{D}$):

$$\begin{aligned}\boldsymbol{\theta}^* &= \arg\min_{\boldsymbol{\theta}} \sum_{\boldsymbol{d}\in\mathcal{D}} \sum_{j=1}^{|\boldsymbol{d}|} -\log P_{\boldsymbol{\theta}}(\boldsymbol{y}^j|\boldsymbol{x}^j, \boldsymbol{Y}^{-j}, \boldsymbol{X}^{-j}) \\ &= \arg\min_{\boldsymbol{\theta}} \sum_{\boldsymbol{d}\in\mathcal{D}} \sum_{j=1}^{|\boldsymbol{d}|} \sum_{n=1}^{|\boldsymbol{y}^j|} -\log P_{\boldsymbol{\theta}}(y_n^j \mid \boldsymbol{y}_{<n}^j, \boldsymbol{x}^j, \boldsymbol{Y}^{-j}, \boldsymbol{X}^{-j}) \end{aligned} \tag{2.21}$$

This training objective is a generalisation of the objective for sentence-level NMT. In sentence-level NMT, the conditional log-likelihood is defined as the sum of the log-probability of predicting a correct symbol y_n in the output sentence for each instance $\boldsymbol{x}$ in the training set $\mathcal{D}$. Thus, the optimum set of parameters $\boldsymbol{\theta}^*$ are found as follows:

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \sum_{(\boldsymbol{x},\boldsymbol{y})\in\mathcal{D}} -\log P_{\boldsymbol{\theta}}(\boldsymbol{y} \mid \boldsymbol{x}) \tag{2.22}$$

$$= \arg\min_{\boldsymbol{\theta}} \sum_{(\boldsymbol{x},\boldsymbol{y})\in\mathcal{D}} \sum_{n=1}^{|\boldsymbol{y}|} -\log P_{\boldsymbol{\theta}}(y_n \mid \boldsymbol{y}_{<n}, \boldsymbol{x}) \tag{2.23}$$

The partition function in Eq. 2.19 is intractable searching problem due to the huge search space over the possible translations which is exponentially large with respect to the number of sentences in the documents. However, the parameters can be learned by resorting to minimising the negative *pseudo-likelihood* (Besag, 1975) as suggested in (Maruf & Haffari, 2018).

Decoding. The same as sentence-level NMT, for document-level NMT we also need to generate the highest probability translation but for a given source document instead of source sentence. To do so, the following optimisation problem needs to be solved:

$$\arg\max_{\boldsymbol{Y}} P_{\boldsymbol{\theta}}(\boldsymbol{Y}|\boldsymbol{X})$$

where $\boldsymbol{Y} := \{\boldsymbol{y}^1, \ldots, \boldsymbol{y}^{|\boldsymbol{d}|}\}$ is the translation document, and $\boldsymbol{\theta}$ refers to the parameters of normalised model. In sentence-level NMT, the best output sentence $\hat{\boldsymbol{y}}$ for a given input

sentence $\boldsymbol{x}$ and corresponding reference sentence $\boldsymbol{y}$ is produced by:

$$\hat{\boldsymbol{y}} = \arg\max_{\boldsymbol{y}} P_{\boldsymbol{\theta}}(\boldsymbol{y}|\boldsymbol{x}) \tag{2.24}$$

In the third chapter of this thesis, we train a document-level NMT model and we use it to initialize our data-to-document generation model.

2.2 Transfer Learning

Transfer Learning is a machine learning technique where the knowledge learned in one setting (task/domain) is utilized as the starting point for learning a new task in a new setting. Humans also do this rather inherently (Torrey & Shavlik, 2010). For example, in human cognitive system, learning to recognize apples helps to recognize pears (Pan & Yang, 2009). In neural text generation, transfer learning is an appealing approach to address the dearth of data for low-resource scenarios when fine-tuning a pre-trained model (trained with high-resource pair of languages) with low-resource data improves the accuracy of low-resource model (Dabre et al., 2017; Kocmi & Bojar, 2018).

Here, we first introduce transfer learning and highlights its benefits. To do so, we elaborate the differences between traditional Machine Learning and Transfer Learning (See Figure 2.5 and Figure 2.6). Traditional machine learning approaches are isolated and make predictions purely based on the models trained on specific tasks and specifically collected labelled or unlabeled training data. The models trained with traditional machine learning should be rebuilt from scratch once the feature-space distribution changes. No knowledge is preserved to transfer from the earlier trained models to the related new models. Transfer learning, in contrast, transcends the isolated learning paradigm by leveraging knowledge (features, weights, etc.) gained in previously trained models for training the related new models. Reusing knowledge by transfer learning usually tackles the problems of data scarcity for newer tasks (Sarkar, Bali, & Ghosh, 2018).

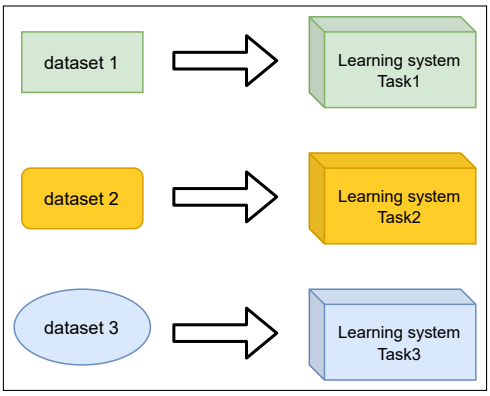

Figure 2.5: Traditional ML

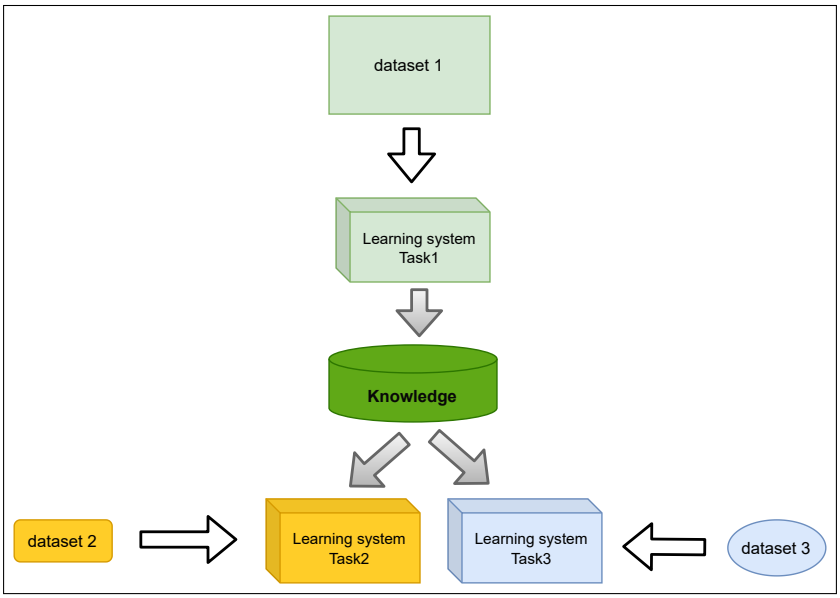

Figure 2.6: Transfer learning

Formal definition. To define the formulation of transfer learning, we need first to define the terms of *domain* and *task* (Pan & Yang, 2009). A domain $\mathcal{D}$ is defined based on two components: a) feature space $\mathcal{X}$ and b) marginal probability distribution $P(X)$, where $X = \{x_1, \ldots, x_n\} \in \mathcal{X}$. A task $\mathcal{T}$ given a specific domain $\mathcal{D} = \{\mathcal{X}, P(X)\}$ also consists of two components: a) a label space $\mathcal{Y}$ and b) an objective predictive function $f : \mathcal{X} \rightarrow \mathcal{Y}$ which is a predictive function for a new data point x. The task, $\mathcal{T} = \{\mathcal{Y}, f(x)\}$ is learned from the training data consisting of pairs $\{x_i, y_i\}$, where $x_i \in X$ and $y_i \in \mathcal{Y}$ (Torrey & Shavlik, 2010).

Given a source domain $\mathcal{D}_s$ and learning task $\mathcal{T}_s$, a target domain $\mathcal{D}_t$ and learning task $\mathcal{T}_t$, transfer learning aims to help improve the learning of the target predictive function $f_t(\cdot)$ in $\mathcal{D}_t$ using the knowledge in $\mathcal{D}_s$ and $\mathcal{T}_s$, where $\mathcal{D}_s \neq \mathcal{D}_t$, or $\mathcal{T}_s \neq \mathcal{T}_t$.

Once we make sure that we have a machine learning setting where transfer learning is applicable, we have to answer three critical questions: *What, When, and How to transfer?*

This is important to know which part of the knowledge is useful and transferable to transfer from the source task/domain to the target task/domain. To find the answer to this question, we need to identify which part of knowledge is common between source and target task/domain and in which cases knowledge is specified for only the source task/domain (Sarkar et al., 2018; Yosinski, Clune, Bengio, & Lipson, 2014). Choosing the less relevant knowledge from the source domain/task to the target domain/task may inversely hurt the target performance, a phenomenon known as *negative transfer* (Z. Wang, Dai, Póczos, & Carbonell, 2019).

Once the most relevant and useful knowledge to transfer is identified, we need to know in which stage of the training, transfer learning should and in which situations should *not* occur. The optimal stage for transferring knowledge can vary based on the discrepancy in the joint distributions between the source and target domains and the size of the data in source and target tasks (Torrey & Shavlik, 2010). Brute-force transfer may cause a negative transfer when the source and target domain are unrelated.

Different transfer learning strategies can be categorized and applied under three sub-settings regarding the various situations between the source and target domains and tasks. These three categories are listed as follows: a) *Inductive transfer learning* setting in which the source and target domains are similar; however, the source and target tasks are not. b) *Unsupervised transfer learning* setting in which the source and target domains are similar, but the tasks are different and labelled data is unavailable in either domain. c) *Transductive transfer learning* setting where the source and target tasks are the same but the source and target domains are not. In this setting, the source domain has many labelled data, while the

target domain has none. This setting can be classified into subcategories, where either the feature spaces or the marginal probabilities are different (Sarkar et al., 2018).

Most of transfer learning approaches implicitly assuming that the source and target domains are related. Nevertheless, *how to avoid negative transfer* is still an important open question when the source and target domains are different (Sarkar et al., 2018). In chapters 4 and 5 of this thesis, we will introduce two different solutions to avoid negative transfer when we aim to transfer knowledge from different domains (language pairs).

Language transfer. Transfer learning through a multilingual setting from high-resource to low-resource language pairs is a widely used approach for improving the accuracy of low-resource scenarios in NLP. In such approaches, the high-resource language is usually referred to as the parent language (teacher model), and the low-resource language is referred to as the child language (student model) (Pan & Yang, 2009). This approach has been explored widely in the key tasks of NLP (Hwa, Resnik, Weinberg, Cabezas, & Kolak, 2005; McDonald, Petrov, & Hall, 2011; Petrov, Das, & McDonald, 2012; Zhang & Barzilay, 2015) and particularly in machine translation (Zoph et al., 2016; Dabre, Fujita, & Chu, 2019; Gu, Wang, Chen, Li, & Cho, 2018). Transfer learning approaches that are conducted in NMT can be categorized into five groups: i) transfer learning on the source-side, ii) transfer learning on the target-side, iii) transferring lexical knowledge, iv) transferring syntactic knowledge, and v) transfer learning based on language similarity (Dabre et al., 2020).

In **transfer learning from the source-side**, both high-resource and low-resource NMT tasks share the same target language. For example, we have German-to-English as an high-resource NMT task and Galician-to-English as a low-resource NMT task. The simplest way to apply such an approach is *fine-tuning* the high-resource model (German-to-English) with low-resource data (Galician-to-English). In other words, the child model is initialized with the parent's parameters while trained with the low-resource parallel corpus (Zoph et al., 2016). In such approaches, deciding about the amount of fine-tuning (selectively tune a subset of the child's parameter or tune the whole model) is very important. In a study

proposed by (Zoph et al., 2016), the optimal solution is to fine-tune all parameters except the input and output embedding. However, this might be different for different language pairs and different settings in NMT. Given the languages utilized to train the child and the parent models, and the size of the data, optimizing the parent model may not be the best objective for the child task. A better approach can be training a parent model to some extent which can be adjusted and fine-tuned on child tasks quickly. Gu, Wang, et al. (2018) proposed a meta-learning approach by extending the model-agnostic meta-learning algorithm (Finn, Abbeel, & Levine, 2017) to learn the best parameter initialization from multilingual high-resource language tasks that facilitates transfer learning to the new language pairs with the minimum training examples. This approach with meta-learned parameter initialization shows its significant effectiveness comparing to the general multilingual transfer learning, especially for highly low-resource tasks.

While the majority of transfer learning based approaches in multilingual NMT have the same target languages in parent and child models, there are also some approaches which have different target languages and do **transfer learning on the target-side** (Johnson et al., 2017; Dabre et al., 2019). In these approaches, transfer learning is more challenging due to the catastrophic forgetting risk (Robins, 1995). To this end, Dabre et al. (2019) proposed a multi-stage training approach when multiple target languages are involved, and the source language is fixed as English. The training stages include the pre-training stage on a parallel corpus for a high resource language pair, the mixed pre-training/fine-tuning stage on a mixture of parallel corpora including helper high-resource and one or more low-resource language pairs, and the last pure fine-tuning stage using only a parallel corpus for a particular low-resource language pair. They performed the sequential fine-tuning by involving the parallel corpora with the relatively larger size in the intermediate level of fine-tuning to help the low-resource language pairs training in the last stage of fine-tuning. This approach outperforms the one-to-one transfer learning in one-to-many multilingual settings.

Transferring knowledge from high-resource to low-resource NMT task not only is effective for initializing the child model's parameters but also can be done for initializing the

word embedding of the child source language known as the **lexical transfer**. Basically, the multilingual NMT model is trained with a vocabulary set consisting of the vocabularies of all involving languages. In this setting, each word has its own representation, and there is no shared embedding space between the words in multiple languages. This is not problematic for high-resource languages with enough data; however, it is sub-optimal for truly low-resource languages which have not most of the vocabulary units in their training data. Sharing the sub-units between the languages is a possible solution when there is a big overlap between the lexicons in high-resource and low-resource languages (Sennrich, Haddow, & Birch, 2016b; Y. Kim, Jernite, Sontag, & Rush, 2016; Luong & Manning, 2016; Lee et al., 2017). Otherwise, a shared semantic representation between languages is needed. This can be done by mapping pre-trained monolingual word embeddings of the parent and child sources to a shared vector space (Gu, Hassan, Devlin, & Li, 2018; Y. Kim, Gao, & Ney, 2019).

Fine-tuning the high-resource NMT model with low-resource data is not optimal when high-resource and low-resource languages do not have *syntactic converge*. There are few pieces of research that work on **syntactic transfer** to address this aspect of language transfer in multilingual transfer learning. Murthy et al. (2019) suggested an approach to reduce the syntactic divergence between source languages in parent and child models by reordering the parent's sentences to match the word order in child language. This approach achieved significant improvement for multilingual NMT in an extremely low-resource setting. Reordering the words was also proposed in SMT (M. Collins, Koehn, & Kučerová, 2005), and NMT (Zhao, Zhang, & Zong, 2018) to get over syntactic divergence between source and target languages in machine translation. Besides the mentioned word reordering approaches, which all are applied in the preprocessing step, (Y. Kim, Petrov, Petrushkov, Khadivi, & Ney, 2019) introduced a new approach to overcome the syntactic divergence by training the parent encoder with noisy source data. The main motivation of this approach is preventing the parent encoder from being over-optimized on the syntactic of the parent's source language, which allows the model to learn better the syntactic of the child's source language. Using

multiple high-resource languages in the parent side is another effective approach proposed by (Gu, Hassan, et al., 2018) to control the syntactic language divergence in multilingual transfer learning.

All the aforementioned approaches outperform unsupervised learning and gain significant improvement, especially for extremely low-resource scenarios regardless of the language relatedness. This success can be credited to the NMT models' ability to learn cross-lingual representations. However, when distant languages come to account, learning cross-lingual representations does not achieve its optimal performance (Pires, Schlinger, & Garrette, 2019; Søgaard, Ruder, & Vulić, 2018; Dabre et al., 2020). A new direction that has recently emerged uses *typological guidance* as a form of non-parallel linguistic information in knowledge transfer. This direction comes from the fact that languages have some systematic inter-lingual relations at different typological levels (e.g. similarities in language structure or language morphology, etc.), in spite of their significant diversity (O'Horan et al., 2016). Those relations has been captured in *typological classifications* and then applied as a guide to define the optimal language transfer direction in a variety of NLP tasks (Bender, 2011; Hana, Feldman, & Brew, 2004; Wisniewski, Pécheux, Gahbiche-Braham, & Yvon, 2014; S. B. Cohen & Smith, 2009; McDonald et al., 2011; Berg-Kirkpatrick & Klein, 2010; Naseem, Chen, Barzilay, & Johnson, 2010; Täckström, McDonald, & Nivre, 2013). In particular, for multilingual NMT, (Zoph et al., 2016) and (Dabre et al., 2017) argued that having related languages in parent and child models has a substantial impact on transfer learning result. (Maimaiti, Liu, Luan, & Sun, 2019) also showed that using highly related high-resource language pairs on the parent side along with multistage fine-tuning can significantly improve the translation results. Following this literature, in Chapter 5, we will introduce our hierarchical knowledge distillation approach which can effectively transit knowledge from clusters of relevant high-resource language pairs to the low-resource ones in a hierarchy structure.

2.3 Knowledge Distillation

Deep neural networks have achieved significant achievements in real-world applications, particularly when a massive amount of data is available. Nevertheless, the resource-intensive deployment process of deep neural models in constrained environments such as mobile appliances or IoT devices is still a major challenge due to the limited memory capacity and low computational power of these devices. To address this issue, different model compression (Cheng, Wang, Zhou, & Zhang, 2017) techniques have been developed, which can be categorized as the parameter pruning and quantization, low-rank factorization, transferred/compact convolutional filters, and knowledge distillation. In this section, we only focus on knowledge distillation as it is the most relevant model compression technique which is used for the purpose of this thesis.

The idea of knowledge distillation is initiated for the first time with Bucilua^ et al. (2006). They proposed a compression approach to decrease deep neural models' deployment footprint in low-power and resource-limited devices without a significant drop in accuracy. The main idea is to compress the knowledge which is learned by a complex model (*teacher*) or ensemble of complex models and transfer it into a smaller, faster model (*student*) while the student is able to produce competitive or even better results compared to the cumbersome teacher network. This idea is then generalized and named *Knowledge Distillation* with Hinton et al. (2015) motivated by the well-known property of deep neural networks, which is the universal approximation.

A vanilla knowledge distillation framework usually has three important components: Knowledge, distillation approach, and the teacher-student architecture. The small "**student**" neural model learns to mimic the behavior of the large "**teacher**" model under a *supervision signal* from a teacher, referred to the "**knowledge**". This knowledge which is learned by the teacher and then transferred to the student, is not explicitly provided by the training data samples and thus named *dark knowledge* (Hinton et al., 2015). In vanilla knowledge distillation, which is used for the purpose of this thesis, the *logits* (the last layer's output in

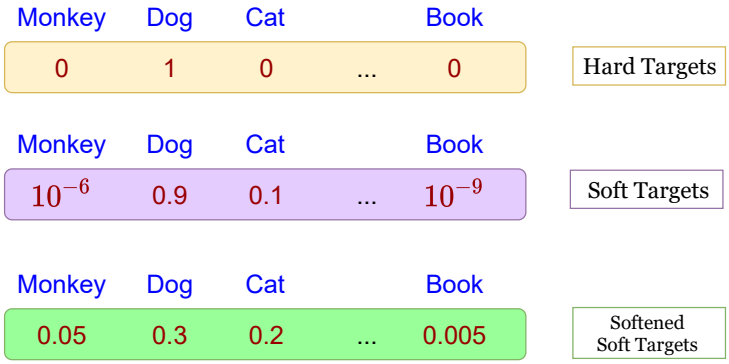

Figure 2.7: Examples of hard and soft targets and the effect of temperature raising in a network's softmax function which results as softened soft targets.

a deep neural network) of a large deep model are used as the teacher knowledge (Hinton et al., 2015; J. Kim, Park, & Kwak, 2018; L. J. Ba & Caruana, 2014; Mirzadeh et al., 2020). However, the activations, neurons or features of intermediate teacher network's layers or the parameters of the teacher model can also be transferred as knowledge to the student model (Gou et al., 2020; Romero et al., 2015; Huang & Wang, 2017; Ahn, Hu, Damianou, Lawrence, & Dai, 2019; Heo, Lee, Yun, & Choi, 2019; Komodakis & Zagoruyko, 2017; Liu et al., 2019).

Formal definition. In the distillation process, for a general classification problem, the teacher's knowledge is transferred to the student by minimizing a loss function in which the target is the output of a softmax function on the teacher model's logits. In other words, given a vector of *logits* z as the output of the last fully connected layer of a deep teacher model, in which z_i is the logit for the i-th class, the probability p_i of the i-th class can be estimated by a softmax function,

$$p_i = \frac{\exp(z_i)}{\sum_j \exp(z_j)} \tag{2.25}$$

This prediction obtained from the teacher model is referred to as ***soft target***. Compared to conventional one-hot ground truth, named ***hard targets***, soft targets has scores for all classes. However, in many cases, this probability distribution assigns a very high probability (close to 1) to the correct class while all other class probabilities are very close to

0, which is indicating higher certainty in the prediction; however, it does not provide much information beyond the one-hot ground truth labels already provided in the dataset. Hinton et al. (2015) suggest that the distribution of the incorrect class labels holds valuable information which defines a rich similarity structure over the data that can be learned from. They explain that soft targets provide much more information per training case than hard targets as the soft targets have higher entropy and less variance in the gradient between training cases. Therefore, the small student model can take advantage of a higher learning rate while using much less data than the original complex teacher model. So, the concept of *softmax temperature* denoted by T is introduced to soften the distribution of probabilities over the class labels (Hinton et al., 2015):

$$p_i = \frac{\exp\left(z_i/T\right)}{\sum_j \exp\left(z_j/T\right)} \tag{2.26}$$

According to Equation 2.26, when temperature grows, the probability distribution over classes becomes softer. For instance, when $T \to \infty$, the same probability is assigned to all classes and when $T \to 0$, the soft targets become the same as the hard targets. In Figure 2.7 you can see an example of hard and soft targets and the effect of temperature raising in a network's softmax function, which results in softened soft targets. The softened outputs reveal the dark knowledge embedded in the teacher model and are transferred to the student during the distillation process. As you can see in Figure 2.8, both original hard targets and the softened targets are involved during the training of the student. The first is used for calculating the general *student loss* and the second is utilized for *distillation loss*. In fact, in the distillation process, the student model is trained to imitate the soft targets by minimizing the sum of two different cross-entropy functions which are used for ultimate loss: one involving the softened targets referred to as *distillation loss*, and one involving the original hard targets named *student loss* (see Figure 2.8. The distillation loss can be formulated as following where the logits of student model z_s and the logits of teacher model

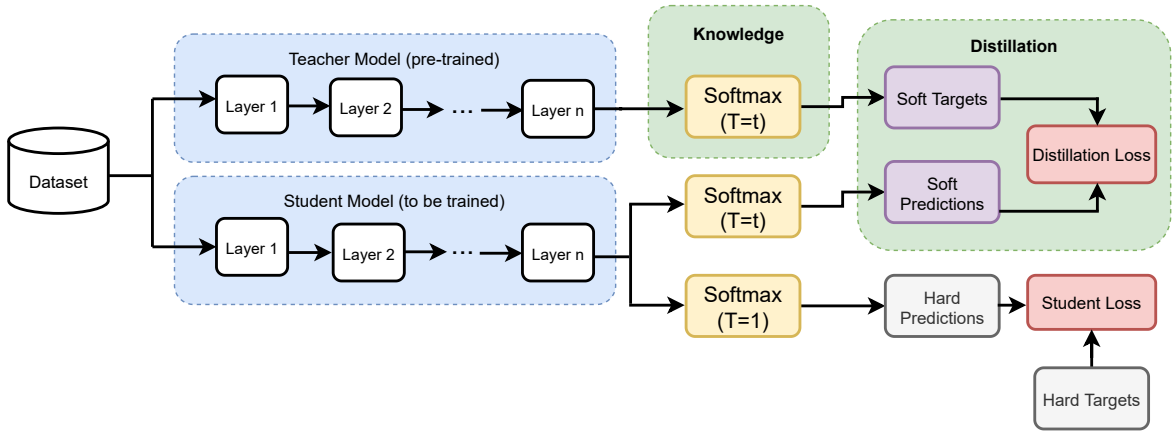

Figure 2.8: Vanilla knowledge distillation framework (Gou et al., 2020)

z_t are matched through a cross-entropy loss:

$$L_D(p(z_t, T), p(z_s, T)) = \sum_i -p_i(z_{t_i}, T) \log(p_i(z_{s_i}, T)) \tag{2.27}$$

The student loss is formulated as the cross-entropy between the ground truth vector (hard targets) y and the soft logits of the student model (Gou et al., 2020):

$$L_S(y, p(z_s, T)) = \sum_i -y_i \log(p_i(z_{s_i}, T)) \tag{2.28}$$

As it is shown in figure 2.8, the same logits of the student model are used for both student loss and distillation loss; however, for the student loss with softmax temperature = 1 and for distillation loss with softmax temperature = t. The ultimate loss for training the vanilla knowledge distillation method is a joint of student and distillation loss as follow (Gou et al., 2020):

$$L_{KD}(x, \theta) = \alpha * L_D(p(z_t, T), p(z_s, T)) + (1 - \alpha) * L_s(y, p(z_s, T)) \tag{2.29}$$

Where x is the training input sample, θ are the parameters of the student model, and α is the weighting factor to control the contribution of components of the ultimate loss.

In Chapter 4, we will present our novel ensemble knowledge distillation approach which adaptively distill knowledge from ensemble of teachers to a student. In this approach, the label smoothing coming from different teachers is combined and regulated, based on the loss incurred by the teacher models during the distillation process, and thus the contribution of each teacher is changed based on its effectiveness to improve the student.

3 | From Machine Translation to Document Generation

Recently, neural models led to significant improvements in both machine translation (MT) and natural language generation tasks (NLG). However, generation of long descriptive summaries conditioned on structured data remains an open challenge. Likewise, MT that goes beyond sentence-level context is still an open issue (e.g., document-level MT or MT with metadata). To address these challenges, we propose to leverage data from both tasks and do transfer learning between MT, NLG, and MT with source-side metadata. First, we train a document-based NMT system with the DGT parallel data. Then, we augment this NMT model to obtain a "Data + Text to Text" model. Finally, we remove the source text to get a pure NLG system, able to translate from metadata to full documents. This end-to-end NLG approach, without data selection and planning, outperforms the previous state of the art on the Rotowire NLG dataset. Meanwhile, our systems submitted to WNGT 2019 obtained the best results on each of the 6 tasks.

3.1 Introduction

Neural Machine Translation (NMT) and Neural Language Generation (NLG) are the top lines of the recent advances in Natural Language Processing. Although state-of-the-art NMT systems have reported impressive performance on several languages, there are still

many challenges in this field especially when context is considered. Currently, the majority of NMT models translate sentences independently, without access to a larger context (e.g., other sentences from the same document or structured information). Additionally, despite improvements in text generation, generating long descriptive summaries conditioned on structured data is still an open challenge (e.g., table records). Existing models lack accuracy, coherence, or adequacy to source material (Wiseman et al., 2017).

The two aspects which are mostly addressed in data-to-text generation techniques are identifying the most important information from input data, and verbalizing data as a coherent document: *"What to talk about and how?"* (Mei et al., 2016). These two challenges have been addressed separately as different modules in pipeline systems (McKeown, 1985; Reiter & Dale, 2000a) or in an end-to-end manner with PCFGs or SMT-like approaches (Mooney & Wong, 2007; Angeli et al., 2010; Konstas & Lapata, 2013), or more recently, with neural generation models (Wiseman et al., 2017; Lebret et al., 2016; Mei et al., 2016). In spite of generating fluent text, end-to-end neural generation models perform weakly in terms of best content selection (Wiseman et al., 2017). Recently, (Puduppully, Dong, & Lapata, 2019a) trained an end-to-end data-to-document generation model on the Rotowire dataset (English summaries of basketball games with structured data).[1] They aimed to overcome the shortcomings of end-to-end neural NLG models by explicitly modelling content selection and planning in their architecture.

We suggest in this chapter to leverage the data from both MT and NLG tasks with transfer learning. As both tasks have the same target (e.g., English-language stories), they can share the same decoder. The same encoder can also be used for NLG and MT if the NLG metadata is encoded as a text sequence (See Figure 3.1). We first train domain-adapted document-level NMT models on large amounts of parallel data. Then we fine-tune these models on small amounts of NLG data, transitioning from MT to NLG. We show that separate data selection and ordering steps are not necessary if NLG model is trained at document-level and is given enough information. We propose a compact way to encode the data avail-

[1] https://github.com/harvardnlp/boxscore-data

able in the original database, and enrich it with some extra facts that can be easily inferred with a minimal knowledge of the task. We also show that NLG models trained with this data capture document-level structure and can select and order information by themselves.

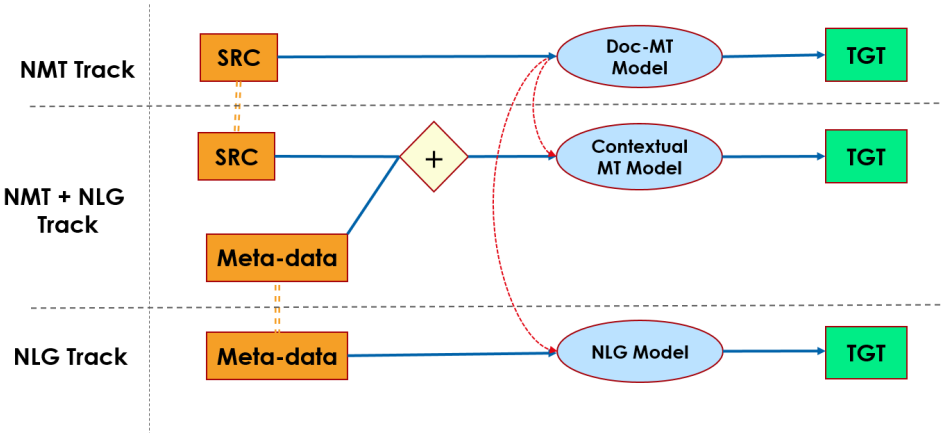

Figure 3.1: Our transfer learning scheme in DGT shared task for transitioning from machine translation task to data-to-text generation task. In NMT track, the source document is translated to the target document (Doc-MT model). In NMT+NLG track, the source document concatenated with structured meta data is translated to the target document (contextual MT model). In NLG track, the meta-data is translated to the target document (NLG model).

3.2 Related Work

The data-to-text generation literature mostly focuses on two main challenges including: i) content selection and planning, and ii) summary generation or surface realization. These challenges have been addressed either in the pipeline approaches or in end-to-end systems. A considerable amount of research has been published on content planning as the key component in data-to-text generation problems. This component, which is named in the literature by "Text Planner", "Document Planner", or "Macroplanner", basically combines content selection and content structuring. The resulting text plan of content planning stage is the input to the "Sentence Planner" or "Microplanner", which combines and aggregates all selected words and phrases into well-formed sentences to express the information in summary generation (Gatt & Krahmer, 2018). Content selection has been developed with either hand-built rules (Kukich, 1983; Dale & Reiter, 1997; Duboué & McKeown, 2003) or has

been learned from the data (Barzilay & Lapata, 2005; Duboue & McKeown, 2001; Duboué & McKeown, 2003; Liang, Jordan, & Klein, 2009; J. Kim & Mooney, 2010; Konstas & Lapata, 2013). Earlier content selection components have been developed based on generic planners (Dale, 1989) or discourse-dependent planners (Hovy, 1993) which are based on hand-crafted domain-dependent structuring rules (called "schemata" by (McKeown, 1985)) obtained from the target text[2]. For example, Williams and Reiter (2008) used a constraint approach to maximise readability where constraints are based on corpus analysis and linguistically motivated rules using Rhetorical Structure Theory (RST) relations. In their approach, document planner generates a tree, in which core messages are connected by discourse relations. Duboué and McKeown (2003) learn Content Selection rules, using texts paired with a frame-based knowledge graph. The semantic data which is used in the text matching was clustered and scored according to its occurrence in text. More recent work gradually moves towards data-driven statistical approaches and focuses on end-to-end systems in which content selection and surface realization are jointly learned. Konstas and Lapata (2013) train a model which captures the implicit relationship between the records of the database and the text. The key contribution of this work is representing content plans as grammar rules using a probabilistic context-free grammar (PCFG) approach and intersecting this grammar with an n-gram language model and a dependency model. Thus, this approach casts the two phases of content selection and summary generation to a common parsing problem. As they train a single model for both content selection and surface realization stages, their model is conceptually more straightforward than the previous similar approaches (Angeli et al., 2010; J. Kim & Mooney, 2010) which broke up the generation task into a sequence of local decisions to select the meaningful and related record of the database. This approach has been applied on weather forecasts dataset and performed well compared to its counterparts. However, it is not easily generalizable to larger data with longer summaries and more diverse vocabularies.

Later, Wiseman et al. (2017) introduced a larger dataset of basketball games for data-

[2]Example: In basketball game summary generation, two consecutive scores of a player should be described in one sentence (Gatt & Krahmer, 2018).

to-text generation task with 628 input records and the average length of 330 words per summary. They also proposed the new qualitative metrics to measure the quality and adequacy of generated text. Another large dataset is the MLB dataset which is introduced by (Puduppully, Dong, & Lapata, 2019a). This dataset contains about 25k records along with the summaries which describe the baseball games based on the the game statistics. These datasets with sufficient data opened the way for neural generation systems with encoder-decoder architecture to be used for data-to-text generation (Lebret et al., 2016; Mei et al., 2016; Wiseman et al., 2017). Although the neural generation models perform well in generating fluent and coherent documents, they still tend to generate hallucinations and underperform in content selection and generating adequate document. One of the widely suggested solution to tackle this problem is changing the way that statistical information (entities) are represented by entity modelling (Puduppully, Dong, & Lapata, 2019b) or using entity-tracking module (Iso et al., 2020). In these approach, the entity-specific representations are dynamically updated during the text generation and fed to the model along with the textual input table.

Inspired by our work represented in this chapter, (Puduppully & Lapata, 2021) proposed a plan-and-generate approach called macro planning. The paragraph plans are basically the sequence of paragraphs separated by special indicators. Each paragraph is a text sequence which contains special tags to indicate the entity's types along with the textual entities' values. They first trained a macro plan model to compute the paragraph plan representations. Then for the text generation step, they maximized the log likelihood of the output text given the macro plan's representations. They showed that macro-planning approach is fairly successful in generating faithful and coherent long documents.

3.3 Document-Level Generation and Translation Task

The goal of the Document-Level Generation and Translation (DGT) task (Shown in Figures 3.2, 3.3) is to generate summaries of basketball games, in two languages (English and German), by using either structured data about the game, a game summary in the other language,

or a combination of both. The task features 3 tracks, times 2 target languages (English or German) which are described as follows (Hayashi et al., 2019):

- NLG (Data → En, Data → De): Generate document summaries in a target language (German or English) given only structured data.
- MT (De ↔ En): Translate documents in the source language (German or English) to the target language (English or German).
- MT+NLG (Data+En → De, Data+De → En): Generate document summaries given the structured data and the summaries in another language.

The data and evaluation are document-level, encouraging participants to generate full documents, rather than sentence-based outputs. The main dataset used in this task is RotoWire[3]. The original RotoWire dataset is a monolingual English dataset that has been used for data-to-text natural language generation, and we have had a portion of this dataset manually translated into German called DGT dataset. RotoWire data set has 4853 distinct rotowire summaries/data, covering NBA games played between 2014 to 2017. Each sample is a JSON file and contains a list of JSON objects corresponding to each aligned summary/data pair. Each JSON object has the following fields: Name of the home team, city of the home team, name of visiting team, city of visiting team, date of the game, tokenized summary of the game, home team line-scores and visiting team line-scores. DGT dataset has subsets of 242 samples for train, 240 samples for validation, and 241 samples for test in both German and English. Table 3.1 describes other allowed parallel and monolingual corpora.

3.4 Proposed Transfer Learning Approach

The main contribution of our systems submitted to the DGT shared task is related to our NLG approach which is proposed for data-to-text generation tracks. In this section, we will explain the problem definition and the modeling assumptions and hypotheses underlying our

[3] https://github.com/harvardnlp/boxscore-data
[4] https://sites.google.com/view/wngt19/home?authuser=0

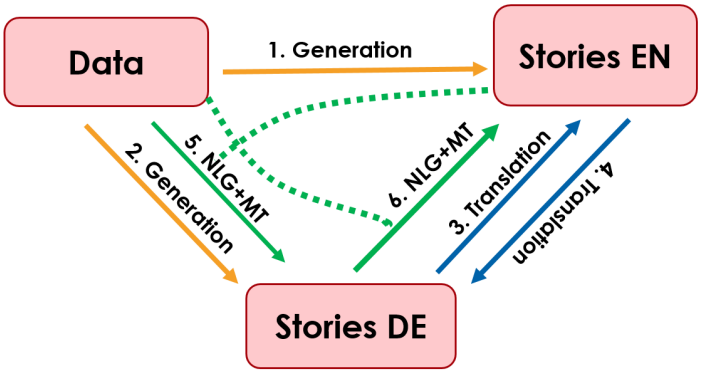

Figure 3.2: All sub-tasks of DGT challenge. This task has 3 tracks, times 2 target languages (English or German): **NLG** (Data to Text), **MT** (Text to Text), and **MT+NLG** (Text + Data to Text).

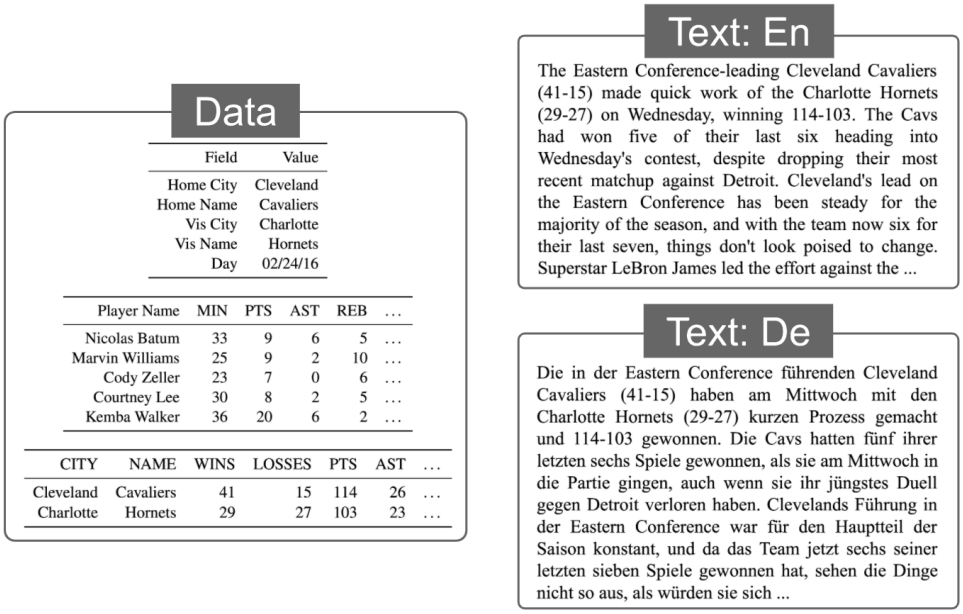

Figure 3.3: A sample of training data for DGT shared task[4]. The left table shows the structured metadata which shows the information about basketball game such as scores, name of the players and the teams, and etc. The documents shown in the right side of this figure are the text summaries in German and English generated based on the game's metadata.

Corpus	Lang(s)	Split	Docs	Sents
DGT	EN-DE	train	242	3247
		valid	240	3321
		test	241	3248
Rotowire	EN	train	3398	45.5k
		valid	727	9.9k
		test	728	10.0k
WMT19-sent	EN-DE	train	–	28.5M
WMT19-doc			68.4k	3.63M
News-crawl	EN	train	14.6M	420M
	DE		25.1M	534M

Table 3.1: Statistics of the allowed resources. The English sides of DGT-train, valid and test are respectively subsets of Rotowire-train, valid and test. More monolingual data is available, but we only used Rotowire and News-crawl.

system design.

3.4.1 Problem Definition

We hypothesize that document-level story generation given a single document-level sequential metadata should work better compared to generating from a set of sparse table records using content selection and planning. By having a single document-level sequential metadata as an input we can simulate the data-to-text generation task to the machine translation task by training a model which jointly learns "what to say?" and "how to say?". In other words, we can see the machine translation and data-to-text generation as two sides of one problem. A machine translation model should find the words in a new language and verbalize the translated words in the syntactic structure of the target language. A data-to-text generation model should choose the information from the structured meta-data and verbalize the selected information in a coherent story.

Indeed, document-level sequential metadata is a high-level organization of the document content and document structure which are especially useful for generating document-level stories with a large vocabulary space that requires capturing long-range dependencies in text.

We assume the input to our model is a set of meta-data consisting of special tokens (entity) followed with positional information (attribute) related to the entities. We model the process of generating output summary y given the meta-data input x as a two-step process, namely, generating a document-level story-plan p given the structured meta-data x and generating the output summary y given the document-level story-plan p. So, a story-plan is a sequence of meta-data with entities and attributes describing the game. By entities, we mean the key fields of our structured meta-data, while attributes refer to the values of respected keys. An example of a story-plan is shown at the first row of Table 3.2. Within a story plan, meta-data are verbalized into a text sequence. We retain the same position for entities and attributes in all story-plans.

3.4.2 Model Description

The generated document-level story-plan is a verbalized version of table records that can be treated as a version of the story in the same language but with a very special syntax.

In our first task, document-level NMT, the encoder learned to capture the document-level representation of the story in either English or German. The conditional probability of a target document $\boldsymbol{y}$ given the source document $\boldsymbol{x}$ is decomposed as:

$$P_{\boldsymbol{\theta}}(\boldsymbol{y} \mid \boldsymbol{x}) = \prod_{n=1}^{N} P_{\boldsymbol{\theta}}(y_n \mid \boldsymbol{y}_{<n}, \boldsymbol{x}) \tag{3.1}$$

where y_n is the current target word and $\boldsymbol{y}_{<n}$ are the previously generated words in our document. The conditional log-likelihood is defined as the sum of the log-probability of predicting a correct symbol y_n in the output document for each instance $\boldsymbol{x}$ in the training set $\mathcal{D}$. Thus, the optimum set of parameters $\boldsymbol{\theta}^*$ are found as follows:

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \sum_{(\boldsymbol{x},\boldsymbol{y})\in\mathcal{D}} -\log P_{\boldsymbol{\theta}}(\boldsymbol{y} \mid \boldsymbol{x}) \tag{3.2}$$

$$= \arg\min_{\boldsymbol{\theta}} \sum_{(\boldsymbol{x},\boldsymbol{y})\in\mathcal{D}} \sum_{n=1}^{|\boldsymbol{y}|} -\log P_{\boldsymbol{\theta}}(y_n \mid \boldsymbol{y}_{<n}, \boldsymbol{x}) \tag{3.3}$$

So, we already have a very strong language model that can predict each word in the document given the previous words. In the new task, data-to-text generation, the trained model needs to recall what it learned in the NMT task while learning the new syntax of the augmented language and generating the output stories given the story-plan.

Given a story-plan $\boldsymbol{p}$, where the set of meta-data segments in the input are $\boldsymbol{S} = \{\boldsymbol{s}^1, \ldots, \boldsymbol{s}^{|\boldsymbol{p}|}\}$, the goal of document-level NLG is to generate a document y. The probability of a document generation given the source story-plan using the chain rule can be formulated as follows:

$$P_{\boldsymbol{\theta}}(\boldsymbol{y} \mid \boldsymbol{p}) = \prod_{n=1}^{N} P_{\boldsymbol{\theta}}(y_n \mid \boldsymbol{y}_{<n}, \boldsymbol{p}) \tag{3.4}$$

The conditional log-likelihood is defined as the sum of the log-probability of predicting a correct symbol y_n in the output document for each instance $\boldsymbol{p}$ in the training set $\mathcal{D}$ and the optimized $\boldsymbol{\theta}$ is defined as follows:

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \sum_{(\boldsymbol{p},\boldsymbol{y})\in\mathcal{D}} -\log P_{\boldsymbol{\theta}}(\boldsymbol{y} \mid \boldsymbol{p}) \tag{3.5}$$

$$= \arg\min_{\boldsymbol{\theta}} \sum_{(\boldsymbol{p},\boldsymbol{y})\in\mathcal{D}} \sum_{n=1}^{|\boldsymbol{y}|} -\log P_{\boldsymbol{\theta}}(y_n \mid \boldsymbol{y}_{<n}, \boldsymbol{p}) \tag{3.6}$$

3.5 Our MT and NLG Systems

All our systems submitted to this shared task (MT, NLG, MT+NLG) are based on Transformer Big (Vaswani et al., 2017b). Details for each track are given in the following sections.

3.5.1 Machine Translation Track

For the MT track, we followed these steps (Shown in Figure 3.4):

1. Train sent-level NMT models on all the WMT19 parallel data (document and sentence)

plus DGT-train.

2. Back-translate (BT) the German and English News-crawl by sampling (Edunov, Ott, Auli, & Grangier, 2018).

3. Re-train sentence-level NMT models on a concatenation of the WMT19 parallel data, DGT-train and BT. The later was split into 20 parts, one part for each training epoch. This is almost equivalent to oversampling the non-BT data by 20 and doing a single epoch of training.

4. Fine-tune the best sentence-level checkpoint (according to valid perplexity) on document-level data. Like (Junczys-Dowmunt, 2019), we truncated the WMT documents into sequences of maximum 1100 BPE tokens. We also aggregated random sentences from WMT-sent into documents, and upsampled the DGT-train data. Contrary to (Junczys-Dowmunt, 2019), we do not use any sentence separator or document boundary tags.

5. Fine-tune the best document-level checkpoint on DGT-train plus back-translated Rotowire-train and Rotowire-valid.

We describe the pre-processing and hyperparameters in Section 3.6. In steps (1) and (3), we train for at most 20 epochs, with early stopping based on newstest2014 perplexity. In step (4), we train for at most 5 additional epochs, with early stopping according to DGT-valid perplexity (document-level). In the last step, we train for 100 epochs, with BLEU evaluation on DGT-valid every 10 epochs. We also compute the BLEU score of the best checkpoint according to DGT-valid perplexity, and keep the checkpoint with highest BLEU.

The models in step (5) overfit very quickly, reaching their best valid perplexity after only 1 or 2 epochs. For DE-EN, we found that the best DGT-valid BLEU was achieved anywhere between 10 and 100 epochs (sometimes with a high valid perplexity). For EN-DE, perplexity and BLEU correlated better, and the best checkpoint according to both scores was generally the same. The same observations apply when fine-tuning on NLG or NMT+NLG data in the next sections.

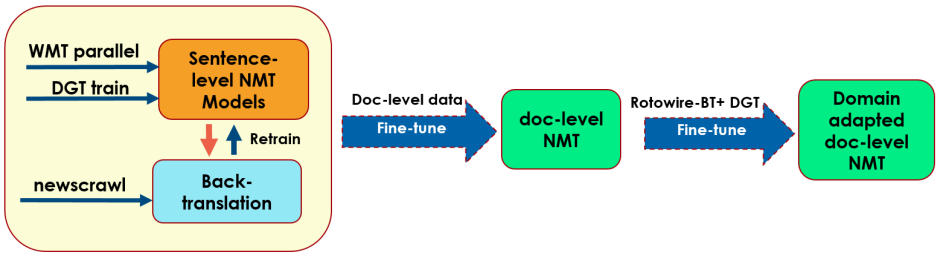

Figure 3.4: Our machine translation system submitted for MT track in DGT shared task.

Like (Berard, Ioan, & Roux, 2019), all our NMT models use corpus tags: each source sentence starts with a special token which identifies the corpus it comes from (e.g., Paracrawl, Rotowire, News-crawl). At test time, we use the DGT tag.

One thing to note, is that document-level decoding is much slower than its sentence-level counterpart.[5] The goal of this document-level fine-tuning was not to increase translation quality, but to allow us to use the same model for NMT and NLG, which is easier to do at the document-level.

3.5.2 Natural Language Generation Track

Original metadata consists of one JSON document per game, containing information about teams and their players. We first generate compact representations of this metadata as text sequences (story-plan). Then, we fine-tune our document-level NMT models (from step 4) on the NLG task by using this representation on the source side and full stories on the target side. We train on a concatenation of DGT-train, Rotowire-train and Rotowire-valid. We filter the later to remove games that are also in DGT-valid. Our story-plan has the following structure:

1. Date of the game as text.

2. Home team information (winner/loser tag, team name and city, points in the game, season wins and losses and team-level scores) and information about its next game (date, home/visitor tag, other team's name), inferred from the other JSON documents in Rotowire-train.

[5]On a single V100, sent-level DGT-valid takes 1 minute to translate, while document-level DGT-valid takes 6 minutes.

3. Visiting team information and details on its next game.
4. N best players of the home team (player name, followed by all his non-zero scores in a fixed order and his starting position). Players are sorted by points first, then by rebounds and assists.
5. N best players of the visiting team.

To help the models identify useful information, we use a combination of special tokens and positional information. For instance, the home team is always first, but a `<WINNER>` tag precedes the winning team and its players. We ignore all-zero statistics, but always use the same position for each type of score (e.g., points, then rebounds, then assists) and special tokens to help identify them (e.g., `<PTS> 16` and `<REB> 8`). We try to limit the number of tags to keep the sequences short (e.g., made and attempted free throws and percentage: `<FT> 3 5 60`). An example of metadata representation (story-plan) is shown in Table 3.2.

3.5.3 MT+NLG Track

For the MT+NLG track, we concatenate the MT source with the NLG data. We use the same metadata encoding method as in the NLG track and we fine-tune our document-level NMT models (from step 4). We also randomly mask tokens in the MT source (by replacing them with a `<MASK>` token), with 20% or 50% chance (with one different sampling per epoch). The goal is to force the model to use the metadata because of missing information in the source. At test time, we do not mask any token.

3.6 Experiments

3.6.1 Data Pre-processing

We filter the WMT19-sent parallel corpus with `langid.py` (Lui & Baldwin, 2012) and remove sentences of more than 175 tokens or with a length ratio greater than 1.5. Then, we apply the official DGT tokenizer (based on NLTK's `word_tokenize`) to the non-tokenized text (everything but DGT and Rotowire).

Story-plan	<DATE> Freitag Februar 2017 <WINNER> Oklahoma City Thunder <PTS> 114 <WINS> 29 <LOSSES> 22 <REB> 47 <AST> 21 <TO> 20 <FG> 38 80 48 <FG3> 13 26 50 <FT> 25 33 76 <NEXT> Sonntag Februar 2017 <HOME> Portland Trail Blazers <LOSER> Memphis Grizzlies <PTS> 102 <WINS> 30 <LOSSES> 22 <REB> 29 <AST> 21 <TO> 12 <FG> 40 83 48 <FG3> 3 19 16 <FT> 19 22 86 <NEXT> Samstag Februar 2017 <VIS> Minnesota Timberwolves <WINNER> <PLAYER> Russell Westbrook <PTS> 38 <REB> 13 <AST> 12 <STL> 3 <PF> 2 <FG> 8 20 40 <FG3> 5 7 71 <FT> 17 17 100 <POS> Guard <PLAYER> Steven Adams <PTS> 16 <REB> 12 <AST> 2 <STL> 1 <BLK> 2 <PF> 4 <FG> 7 13 54 <FT> 2 6 33 <POS> Center <PLAYER> Joffrey Lauvergne <PTS> 16 <REB> 8 <AST> 2 <PF> 3 <FG> 6 7 86 <FG3> 3 4 75 <FT> 1 2 50 <POS> Bank <LOSER> <PLAYER> Marc Gasol <PTS> 31 <REB> 4 <AST> 8 <STL> 2 <BLK> 1 <PF> 4 <FG> 14 24 58 <FG3> 0 4 0 <FT> 3 3 100 <POS> Center <PLAYER> Mike Conley <PTS> 18 <REB> 1 <AST> 2 <STL> 3 <FG> 7 16 44 <FG3> 1 5 20 <FT> 3 5 60 <POS> Guard <PLAYER> Zach Randolph <PTS> 16 <REB> 10 <AST> 3 <STL> 1 <PF> 4 <FG> 6 14 43 <FG3> 0 1 0 <FT> 4 4 100 <POS> Bank
Reference story	The Oklahoma City Thunder defeated the visiting Memphis Grizzlies 114 - 102 , at Chesapeake Energy Arena on Friday evening . The Grizzlies led by four after three quarters , but then Russell Westbrook went absolutely ballistic in the fourth quarter , scoring 19 points in the quarter , including 15 points straight and unanswered , to take his team from down 102 - 99 to the final score of 114 - 102 . This snaps the Grizzlies three-game win streak , while Westbrook added to his ridiculous triple-double count , as he notched his 25th of the season . The Thunder (29 - 22) only scored 21 points in the first quarter , before outscoring the Grizz by 12 in the second , to take an eight-point lead into half time . They were then outscored by 12 in the third , and entered the fourth down by four . The Thunder outscored the Grizz by 14 in the fourth , behind Russell Westbrook and his takeover .Westbrook finished with 38 points , 13 rebounds , 12 assists and three steals to lead his squad to a big win . Westbrook also matched a career-best as he went 17-of-17 from the foul line . Three other players scored in double digits for the Thunder , including Steven Adams and Joffrey Lauvergne , who pitched in 16 points each . The Thunder go on to visit Portland on Sunday . Meanwhile , the Grizzlies (30 - 22) saw their win streak come to an end as they went scoreless in the last 2:50 as they were stifled by an overwhelming push of energy and intensity from the Thunder and their home crowd . Marc Gasol finished with a team-high 31 points , to go along with eight assists , four rebounds , two steals and one block , in 42 minutes . Zach Randolph came up big in 29 minutes off the bench , with 16 points and 10 rebounds , while Mike Conley chipped in 18 points , three steals and two assists . Chandler Parsons added 12 points on 3-of-6 shooting , in 25 minutes . The Grizzlies will have to put this one in the past , as they go right back in action on Saturday , when they visit Minnesota to take on the Timberwolves .
Generated story	The Oklahoma City Thunder defeated the Memphis Grizzlies , 114 - 102 , at Chesapeake Energy Arena on Friday . Despite there being just five ties and six lead changes in this game , it was close throughout . In fact , neither team led by more than 12 points , as the nine-point final margin was one of the largest leads of the game . Oklahoma City actually won this game , despite being out shot from the field , from three-point range and from the free-throw line . Rebounding was the decisive statistic , with the Thunder winning that battle , 47 - 29 . They also dominated the assist-to-turnover ratio , recording 10 more assists and committing three less turnovers . The Grizzlies (30 - 22) have now lost three of their last four games , as they continue to fall down the standings . Marc Gasol did all he could for Memphis , as he led the team with 31 points , four rebounds , eight assists and two steals . Mike Conley finished second on the team , totaling 18 points , two assists and three steals . Mike Conley finished second on the team , amassing 18 points and three steals . Zach Randolph was huge off the bench , providing 16 points , 10 rebounds and three assists . The Grizzlies will be off until Saturday , when they play the Minnesota Timberwolves on the road for their next game . The Thunder (29 - 22) have rode Russell Westbrook 's triple-double train all year and they got another one here . Westbrook collected 38 points , 13 rebounds , 12 assists and three steals . Steven Adams recorded a double-double , amassing 16 points and 12 rebounds . Joffrey Lauvergne was a nice spark off the bench , providing 16 points and eight rebounds . The Thunder will look to keep rolling on Sunday against the Portland Trail Blazers .

Table 3.2: **Story-plan:** our encoded metadata. **Reference story:** story #48 from DGT-valid. **Generated story:** output of the English NLG model (3-player). **Green**: text based on facts from the metadata. **Blue:** correct facts which are not explicitly in the metadata. **Red**: hallucinations or incorrect facts. **Orange**: repetitions.

We apply BPE segmentation (Sennrich, Haddow, & Birch, 2016a) with a joined SentencePiece-like model (Kudo & Richardson, 2018), with 32k merge operations, obtained on WMT + DGT-train (English + German). The vocabulary threshold is set to 100 and inline casing is applied (Berard et al., 2019). We employ the same joined BPE model and Fairseq dictionary for all models. The metadata is translated into the source language of the NMT model used for initialization,[6] and segmented into BPE (except for the special tokens) to allow transfer between NMT and NLG. Then, we add a corpus tag to each source sequence, which specifies its origin (Rotowire, News-crawl, etc.)

Like (Junczys-Dowmunt, 2019), we split WMT19 documents that are too long into shorter documents (maximum 1100 BPE tokens). We also transform the sent-level WMT19 data into document-level data by shuffling the corpus and grouping consecutive sentences into documents of random length. Finally, we upsample the document-level data (WMT19 and DGT) by 8 times its original size (in terms of sentence count). We do so by sampling random spans of consecutive sentences until reaching the desired size.

The DGT and Rotowire data is already tokenized and does not need filtering nor truncating. We segment it into BPE units and add corpus tags.

3.6.2 Settings

All the models are Transformer Big (Vaswani et al., 2017b), implemented in Fairseq (Ott, Edunov, Grangier, & Auli, 2018). We use the same hyper-parameters as (Ott et al., 2018), with Adam and an inverse square root schedule with warmup (maximum LR 0.0005). We apply dropout and label smoothing with a rate of 0.1. The source and target embeddings are shared and tied with the last layer. We train with half-precision floats on 8 V100 GPUs, with at most 3500 tokens per batch and delayed updates of 10 batches. When fine-tuning on DGT-train or Rotowire + DGT-train (Step 5 of the MT track, or NLG/MT+NLG fine-tuning), we use a fixed learning rate schedule (Adam with 0.00005 LR) and a much smaller batch size (1500 tokens on a single GPU without delayed updates). We train for 100 epochs,

[6]Only week days, months and player positions need to be translated.

Track	Target	Constrained	Valid	Test
NLG	EN	no	23.5	20.5
MT		yes	60.2	58.2
MT		no	64.2	62.2
MT+NLG		yes	64.4	62.2
NLG	DE	no	16.9	16.1
MT		yes	49.8	48.0
MT+NLG		yes	49.4	48.2

Table 3.3: Document-level BLEU scores on the DGT valid and test sets of our submitted models in all tracks.

compute DGT-valid perplexity at each epoch, and DGT-valid BLEU every 10 epochs.

3.6.3 BLEU evaluation

Submitted models. For each track, we selected the best models according to their BLEU score on DGT-valid. The scores are shown in Table 3.3, and a description of the submitted models is given in Table 3.4. We compute BLEU using SacreBLEU with its tokenization set to *none*,[7] as the model outputs and references are already tokenized with NLTK. (Hayashi et al., 2019) give the full results of the task: the scores of the other participants, and values of other metrics (e.g., ROUGE). Our NLG models are "unconstrained" because the WMT19 parallel data, which we used for pre-training, was not allowed in this track. Similarly, we do two submissions for DE-EN MT: one constrained, where we fine-tuned the document-level NMT model on DGT-train only, and one unconstrained, where we also used back-translated Rotowire-train and valid. All the NMT and NMT+NLG models are ensembles of 5 fine-tuning runs. Cascading the English NLG model with the ensemble of EN-DE NMT models gives a BLEU score of 14.9 on DGT-test, slightly lower than the end-to-end German NLG model (16.1). We see that in the same data conditions (unconstrained mode), the MT+NLG models are not better than the pure MT models. Furthermore, we evaluated the NMT+NLG models with MT-only source, and found only a slight decrease of ≈ 0.3 BLEU, which confirms our suspicion that the NLG information is mostly ignored.

[7] SacreBLEU signature: *BLEU+case.mixed+numrefs.1+smooth.exp+tok.none+version.1.3.1*

Track	N best players	Details
NLG (EN)	4	Rotowire BT + DGT-train + tags
NLG (DE)	6	Rotowire BT + DGT-train + tags
MT (DE-EN)	N/A	*Unconstrained:* Rotowire BT + DGT-train + tags + ensemble *Constrained:* DGT-train only + ensemble
MT (EN-DE)	N/A	DGT-train only + ensemble
MT+NLG (EN)	3	Rotowire BT + DGT-train + 20% text masking + tags + ensemble
MT+NLG (DE)	3	Rotowire BT + DGT-train + tags + ensemble

Table 3.4: Description of our all submissions for 6 tracks of DGT shared task.

Model	Target	Valid	Test	News 2019
FAIR 2019	EN	48.5	47.7	**41.0**
Sent-level		55.6	54.2	40.9
Document-level		56.5	55.0	38.5
Fine-tuned		**61.7**	**59.6**	21.7
FAIR 2019	DE	37.5	37.0	40.8
Sent-level		47.3	46.7	**42.9**
Document-level		**48.2**	**47.5**	41.6
Fine-tuned		48.0	46.7	41.3

Table 3.5: BLEU scores of the NMT models at different stages of training, and comparison with the state of the art. Scores on DGT-valid and DGT-test are document-level, while News 2019 is sent-level (and so is decoding). On the latter, we used the DGT corpus tag for DE-EN, and the Paracrawl tag for EN-DE (we chose the tags with best BLEU on newstest2014). Scores by the "fine-tuned" models are averaged over 5 runs.

NMT analysis. Table 3.5 shows the BLEU scores of our NMT models at different stages of training (sent-level, document-level, fine-tuned), and compares them against one of the top contestants of the WMT19 news translation task (Ng et al., 2019a). The reason of conducting this experiences is to compare different version of proposed NMT model with other popular test sets (News 2019).

English NLG analysis. Table 3.6 shows a 5.7 BLEU improvement on Rotowire-test by our English NLG model compared to the previous state of the art. Figure 3.5 shows the DGT-valid BLEU scores of our English NLG models when varying the number of players

Model	Rotowire test
(Wiseman et al., 2017)	14.5
(Puduppully, Dong, & Lapata, 2019a)	16.5
Ours (4-player)	**22.2**

Table 3.6: English NLG comparison against state-of-the-art on Rotowire-test. BLEU of submitted NLG (EN) model, averaged over 3 runs. Because Rotowire tokenization is slightly different, we apply a set of fixes to the model outputs (e.g., `1-of-3` → `1 - of - 3`).

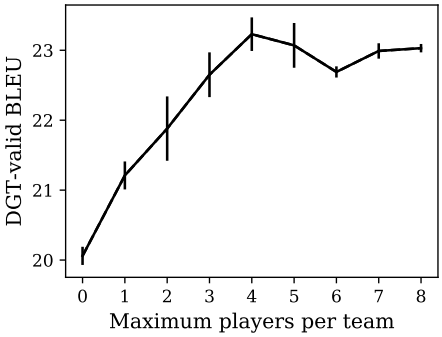

Figure 3.5: DGT-valid BLEU (by the best checkpoint) depending on the maximum number of selected players for the English NLG track.

Model	Valid	Test
Baseline (3 players, sorted)	**22.7**	20.4
No player	20.1	18.8
All players, sorted	**22.7**	20.9
All players, shuffled	22.0	20.0
(1) No next game	22.0	19.9
(2) No week day	22.2	20.5
(3) No player position	22.6	20.5
(4) No team-level sums	22.5	20.5
(5) Remove most tags	22.6	20.8
(1) to (5)	21.3	19.7

Table 3.7: English NLG ablation study, starting from a 3 best player baseline (the submitted NLG model has 4 players). BLEU averages over 3 runs. Standard deviation ranges between 0.1 and 0.4.

selected in the metadata. We see that there is a sweet spot at 4, but surprisingly, increasing the number of players up to 8 does not degrade BLEU significantly. We hypothesize that because the players are sorted from best to worst, the models learn to ignore the last players.

From Table 3.7, we see that sorting players helps, but only slightly. Using only team-

Stadium name (+)	REF: The Golden State Warriors (56 - 6) defeated the Orlando Magic (27 - 35) 119 - 113 at **Oracle Arena** on Monday . NLG: The Golden State Warriors (56 - 6) defeated the Orlando Magic (27 - 35) 119 - 113 on Monday at **Oracle Arena** .
Team alias (+)	REF: The Heat held the **Sixers** to 38 percent shooting and blocked 14 shots in the win . NLG: The **Sixers** shot just 38 percent from the field and 32 percent from the three-point line , while the Heat shot 44 percent from the floor and a meager 28 percent from deep .
Double-doubles or triple-doubles (+)	REF: **Kevin Love 's 29-point , 13-rebound double-double** led the way for the Cavs , who 'd rested Kyrie Irving on Tuesday . NLG: **Love** led the way for Cleveland with a **29-point ,** textbf 13-rebound double-double that also included three assists and two steals .
Player injuries (-)	NLG: The Timberwolves (28 - 44) checked in to Saturday 's contest with an injury-riddled frontcourt , as Ricky Rubio **(knee)** and Karl-Anthony Towns **(ankle)** were sidelined .
Ranking (-)	NLG: The Heat (10 - 22) fell to 10 - 22 and **remain in last place** in the Eastern Conference 's Southeast Division .
Season-level player stats (-)	NLG: It was a season-high in points for Thomas , **who 's now** textbf averaging 17 points per game **on the season**

Table 3.8: Correctly predicted information that is not explicitly in the metadata (+), or hallucinations (-).

level information, and no information about players gives worse but still decent BLEU scores.

Week day, player position or team-level aggregated scores can be removed without hurting BLEU. However, information about next games seems useful. Interestingly, relying on position only and removing most tags (e.g., <PTS>, <FT>) seems to be fine. In this case, we also print all-zero stats, for the position of each statistic to be consistent across players and games.

Train-test overlap on Rotowire. We found a significant overlap between Rotowire train and test: 222 out of 728 Rotowire-test games are also in Rotowire-train (68/241 for DGT-test). The corresponding stories are always different but bear many similarities (some sentences are completely identical). Rotowire-train gets 24.2 BLEU when evaluated against

Rotowire-test (subset of 222 stories). This gives us an estimate of human-level performance on this task. Our submitted NLG model gets 21.8 on the same subset. This overlap may cause an artificial increase in BLEU, that would unfairly favor overfitted models. Indeed, when filtering Rotowire-train to remove games that were also in DGT test, we found a slight decrease in BLEU (19.8 instead of 20.4).

3.6.4 Qualitative evaluation

As shown in Table 3.2, the NLG model (3-player) has several good properties besides coherent document-level generation and the ability to "copy" metadata. It has learned generic information about the teams and players. As such, it can generate relevant information which is absent from metadata (see Table 3.8). For example, the model correctly predicts the name of the stadium where the game was played. This implies that it knows which team is hosting (this information is encoded implicitly by the position of the team in the data), and what is the stadium of this team's city (not in the metadata). Other facts that are absent from the metadata, and predicted correctly nonetheless, are team aliases (e.g., the *Sixers*) and player nicknames (e.g., *the Greek Freak*). The model can also generate other surface forms for the team names (e.g., *the other Cavalier*).

The NLG model can infer some information from the structured data, like double-digit scores, "double-doubles" (e.g., when a player has more than 10 points and 10 assists) and "triple-doubles". On the other hand, some numerical facts are inaccurate (e.g., score differences or comparisons). Some facts which are not present in the structured data, like player injuries, season-level player statistics, current ranking of a team, or timing information are hallucinated. We believe that most of these hallucinations could be avoided by adding the missing facts to the structured data. More rarely, model duplicates a piece of information.

Another of its flaws is a poor generalization to new names (team, city or player). This can quickly be observed by replacing a team name by a fictional one in the metadata. In this case, the model almost always reverts to an existing team. This may be due to overfitting, as earlier checkpoints seem to handle unknown team names better, even though they give

lower BLEU. This generalization property could be assessed by doing a new train/test split, that does not share the same teams.

3.6.5 DGT shared task evaluation

In this section, we outline the summary of all evaluated systems in DGT shared task 2019 (listed in Table 3.9). We first briefly describe the evaluation metrics used for the evaluation and then we show the results of all participated teams accordingly.

Evaluation Metrics. The systems submitted to DGT shared task are evaluated based on *textual accuracy* and *content accuracy*. Standard automatic metrics, BLEU (Papineni et al., 2002a) and ROUGE (Lin, 2004) has been used as the textual accuracy measures for both MT and NLG tracks. To measure the content accuracy for the (monolingual) NLG track, three metrics have been used: (i) **C**ontent **S**election, which is a precision and recall metric of unique relations extracted from a document, (ii) **R**elation **G**eneration, which shows the precision and number of unique relations extracted from a document, and (iii) **C**ontent **O**rdering, which measures how well the system orders the the table records in the generated document. These metrics introduced for the first time by (Wiseman et al., 2017).

System	Ref.
EdiNLG	(Puduppully, Mallinson, & Lapata, 2019)
Naver Labs Europe	(Saleh et al., 2019)
FIT-Monash	(Maruf & Haffari, 2019)
Microsoft	(Werlen, Marone, & Hassan, 2019)
SYSTRAN-AI	(Li, Crego, & Senellart, 2019)

Table 3.9: Participated systems in DGT share task. "Naver Labs Europe" is our submitted system.

Evaluation Results : All systems are evaluated for three tracks: MT, NLG, and MT+NLG. The evaluation results for NLG track are listed in Figures 3.6, 3.7, 3.8, and 3.9 and in Tables 3.12, 3.13, 3.10, and 3.11. In the NLG-related evaluations, DGT shared task had two baselines: i) **NCP+CC (pretrained)**: Pretrained model made available by (Puduppully, Dong, &

Lapata, 2019a), and ii) **NCP+CC (de)**: Model (Puduppully, Dong, & Lapata, 2019a) trained ***only*** on RotoWire En-De. "Oracle" is the evaluation using reference summary.

The evaluation results of MT+NLG-related tasks are shown in Tables 3.14, 3.15, 3.16, and 3.17. The results related to MT tasks also represented in Table 3.18. In MT and MT+NLG-related tracks, the baseline is **FairSeq (WMT'19)**, a pretrained (single) model from (Ng et al., 2019b).

All the evaluation results reported in this section are based on the official website of DGT-shared task 2019[8] and the DGT public leader-board [9].

NLG: Data → En				
		ROUGE-L		
System	BLEU	P	R	F
SYSTRAN-AI	17.59	27.96	26.97	25.6
SYSTRAN-AI-Detok	18.32	28.00	26.97	25.61
EdiNLG	17.01	29.34	24.31	25.38
Naver Labs Europe	**20.52**	**30.83**	**27.77**	**27.29**
Microsoft-GPT-90	13.03	21.39	23.33	21.34
Microsoft-GPT-50	15.17	25.43	22.66	22.93
Microsoft-End-to-End	15.03	28.79	22.29	23.86
NCP+CC(pretrained)	15.80	25.72	23.5	23.46

Table 3.10: DGT shared task's result based on the textual accuracy for NLG (data → En) track.

[8] https://sites.google.com/view/wngt19/home?authuser=0

[9] https://docs.google.com/spreadsheets/d/18ZYbK67uJ2yGlJ48IRWEIkVHN_fP135Ecg-BVPhJeXI/edit#gid=2090491847

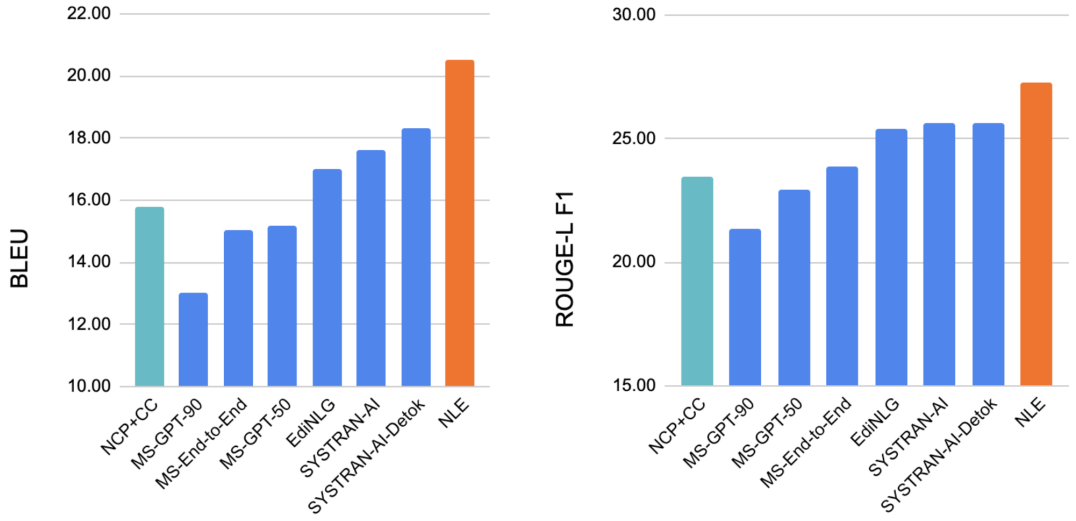

Figure 3.6: DGT shared task's result based on the textual accuracy for NLG (data → En) track (WNGT, 2019).

NLG: Data → En				
	RG	CS		CO
System	P	P	R	DLD
SYSTRAN-AI	83.22	31.74	44.9	20.73
SYSTRAN-AI-Detok	84.16	34.88	43.29	22.72
EdiNLG	91.41	30.91	**64.13**	21.72
Naver Labs Europe	94.08	**41.13**	54.20	**25.64**
Microsoft-GPT-90	88.70	32.84	50.58	17.36
Microsoft-GPT-50	**94.35**	33.91	53.82	19.30
Microsoft-End-to-End	93.38	32.4	58.02	18.54
NCP+CC(pretrained)	88.59	30.47	55.38	18.31
Oracle	100.00	100.00	100.00	100.00

Table 3.11: DGT shared task's result based on the content accuracy for NLG (data → En) track.

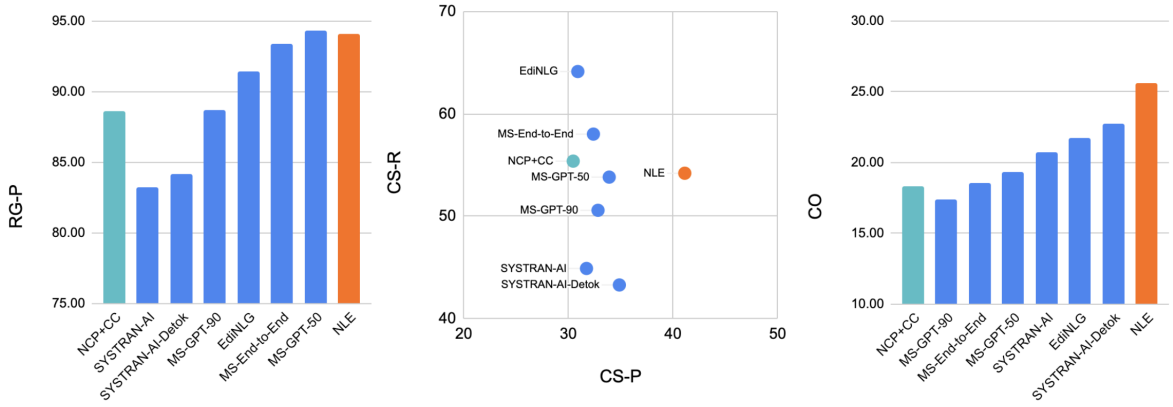

Figure 3.7: DGT shared task's result based on the content accuracy for NLG (data → En) track (WNGT, 2019).

NLG: Data → De				
		ROUGE-L		
System	BLEU	P	R	F
EdiNLG	10.95	**25.94**	17.99	19.7
Naver Labs Europe	**16.13**	25.06	**23.67**	**23.09**
Microsoft-GPT-90	10.43	18.34	20.26	18.43
Microsoft-GPT-50	11.84	21.71	19.52	19.68
Microsoft-End-to-End	11.66	25.09	19.16	20.67
NCP+CC(de)	7.29	18.75	16.18	16.06

Table 3.12: DGT shared task's result based on the textual accuracy for NLG (data → De) track.

NLG: Data → De				
	RG	CS		CO
System	P	P	R	DLD
EdiNLG	70.23	23.40	41.83	16.06
Naver Labs Europe	79.47	29.40	**54.31**	**20.62**
Microsoft-GPT-90	75.05	31.23	41.32	16.32
Microsoft-GPT-50	**82.79**	**34.81**	42.51	17.12
Microsoft-End-to-End	80.30	28.33	49.13	16.54
NCP+CC(de)	49.69	21.61	26.14	11.84
Oracle	100.00	100.00	100.00	100.00

Table 3.13: DGT shared task's result based on the content accuracy for NLG (data → De) track.

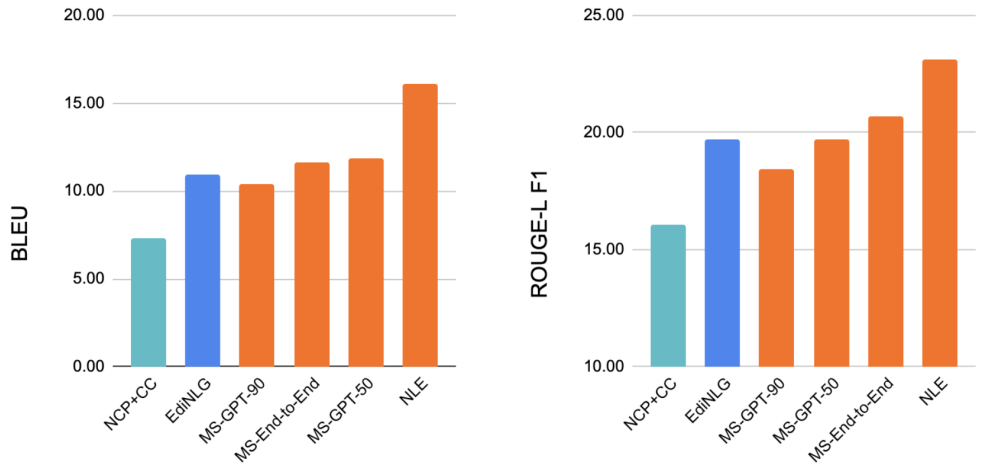

Figure 3.8: DGT shared task's result based on the textual accuracy for NLG (data → De) track (WNGT, 2019).

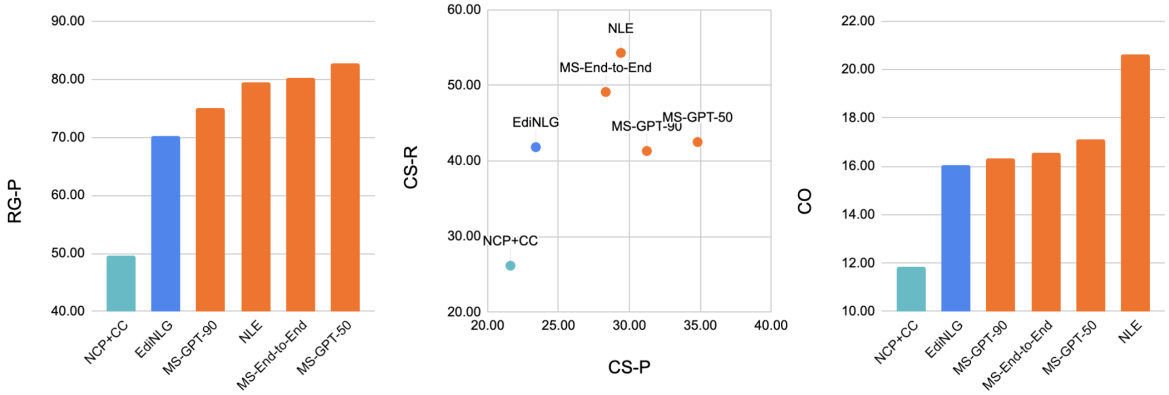

Figure 3.9: DGT shared task's result based on the content accuracy for NLG (data → De) track (WNGT, 2019).

MT+NLG: Data + En → De				
		ROUGE-L		
System	BLEU	P	R	F
EdiNLG	36.85	59.07	55.62	57.25
Microsoft	47.90	65.98	65.33	65.61
Naver Labs Europe	**48.24**	**66.38**	**65.50**	**65.90**
FairSeq (WMT'19)	36.26	57.16	55.68	56.38

Table 3.14: DGT shared task's result based on the textual accuracy for MT+NLG (data+En → De) track.

MT+NLG: Data + En → De				
	RG	CS		CO
System	P	P	R	DLD
EdiNLG	**81.01**	77.32	78.49	62.21
Microsoft	80.98	76.88	84.57	67.84
Naver Labs Europe	80.65	75.10	**88.72**	**69.17**
FairSeq (WMT'19)	81.64	**77.67**	75.82	60.83
Oracle	100.00	100.00	100.00	100.00

Table 3.15: DGT shared task's result based on the content accuracy for MT+NLG (data + En → De) track.

MT+NLG: Data + De → En				
		ROUGE-L		
System	BLEU	P	R	F
EdiNLG	41.15	68.87	64.60	66.62
Microsoft	57.99	76.43	74.54	75.44
Naver Labs Europe	**62.24**	**77.78**	**76.62**	**77.17**
FairSeq (WMT'19)	42.91	69.94	67.50	68.66

Table 3.16: DGT shared task's result based on the textual accuracy for MT+NLG (data + De → En) track.

MT+NLG: Data + De → En				
	RG	CS		CO
System	P	P	R	DLD
EdiNLG	91.40	78.99	63.04	51.73
Microsoft	**95.77**	**92.49**	91.62	84.70
Naver Labs Europe	95.63	91.71	**92.69**	**85.05**
FairSeq (WMT'19)	93.53	83.33	84.22	70.47
Oracle	100.00	100.00	100.00	100.00

Table 3.17: DGT shared task's result based on the content accuracy for MT+NLG (data + De → En) track

3.7 Conclusion

We participated in the 3 tracks of the DGT task: MT, NLG and MT+NLG. Our systems rely heavily on transfer learning, from document-level MT (high-resource task) to document-level NLG (low-resource task). Our submitted systems obtained the best results on each of

System	BLEU	
	(DE → EN)	(EN → DE)
FIT-Monash	47.39	41.46
EdiNLG	41.15	36.85
Naver Labs Europe I	**62.16**	**48.02**
Naver Labs Europe II	58.22	47.90
Microsoft	57.99	47.90
FairSeq (WMT'19)	42.91	36.26

Table 3.18: DGT shared task's result based on the textual accuracy for MT tracks.

the 6 tasks, and this regardless of the metric used.

For the MT task, the usual domain adaptation techniques performed well. The NMT+NLG models did not show any significant improvement over pure NMT. The NMT models are already very good and probably do not need the extra context (which is generally encoded in the source-language summary already). Finally, our NLG models, bootstrapped from the NMT models, do fluent and coherent text generation and are even able to infer some facts that are not explicitly encoded in the structured data. Some of their current limitations (mostly hallucinations) could be solved by adding extra information (e.g., injured players, current team rank, number of consecutive wins, etc.). Our approach is generalizable to other data-to-text generation tasks specially when there is enough available in-domain data which can be used for adapting the machine translation model to the target domain.

Our aggressive fine-tuning allowed us to specialize NMT models into NLG models, but it will be interesting to study a single model can solve both tasks at once (i.e., with multi-task learning), possibly in both languages and whether this multi-task learner is better adaptable when NMT and NLG data are from two different domains.

4 | Improving Low-resource NMT using Adaptive Knowledge Distillation

Scarcity of parallel sentence-pairs poses a significant hurdle for training high-quality Neural Machine Translation (NMT) models in bilingually low-resource scenarios. A standard approach is transfer learning, which involves taking a model trained on a high-resource language-pair and fine-tuning it on the data of the low-resource MT condition of interest. However, it is not clear generally which high-resource language-pair offers the best transfer learning for the target MT setting. Furthermore, different transferred models may have complementary semantic and/or syntactic strengths, hence using only one model may be sub-optimal. In this chapter, we tackle this problem using knowledge distillation, where we propose to distill the knowledge of *ensemble of teacher* models to a single *student* model. As the quality of these teacher models varies, we propose an effective adaptive knowledge distillation approach to dynamically adjust the contribution of the teacher models during the distillation process. Experiments on transferring from a collection of six language pairs from IWSLT to five low-resource language-pairs from TED Talks demonstrate the effectiveness of our approach, achieving up to +0.9 BLEU score improvement compared to strong baselines.

4.1 Introduction

Neural models have been revolutionising machine translation (MT), and have achieved state-of-the-art for many high-resource language pairs (M. X. Chen et al., 2018; Stahlberg, 2019; Maruf et al., 2021). However, the scarcity of bilingual parallel corpora is still a major challenge for training high-quality NMT models (Koehn & Knowles, 2017). Transfer learning by fine-tuning, from a model trained for a high-resource language-pair, is a standard approach to tackle the scarcity of the data in the target low-resource language-pair (Dabre et al., 2017; Kocmi & Bojar, 2018; Saleh et al., 2019; Y. Kim, Gao, & Ney, 2019). However, this is a one-to-one approach, which is not able to exploit models trained for multiple high-resource language-pairs for the target language-pair of interest. Furthermore, models transferred from different high-resource language-pairs may have complementary syntactic and/or semantic strengths, hence using a single model may be sub-optimal.

Another appealing approach is multilingual NMT, whereby a single NMT model is trained by combining data from multiple high-resource and low-resource language-pairs (Johnson et al., 2017; Ha et al., 2016; Neubig & Hu, 2018). However, the performance of a multilingual NMT model is highly dependent on the types of languages used to train the model. Indeed, if languages are from very distant language families, they lead to negative transfer, causing low translation quality in the multilingual system compared to the counterparts trained on the individual language-pairs (Tan, Chen, et al., 2019; Oncevay et al., 2020). To address this problem, (Tan, Ren, et al., 2019) has proposed a knowledge distillation approach to effectively train a multilingual model, by selectively distilling the knowledge from individual teacher models to the multilingual student model. However, still all the language pairs are trained in a single model with a blind contribution during training.

In this chapter, we propose a many-to-one transfer learning approach which can effectively transfer models from multiple high-resource language-pairs to a target low-resource language-pair of interest. As the fine-tuned models from different high-resource language pairs can have complementary syntactic and/or semantic strengths in the target language-

pair, our idea is to distill their knowledge into a single student model to make the best use of these teacher models. We further propose an effective adaptive knowledge distillation (AKD) approach to dynamically adjust the contribution of the teacher models during the distillation process, enabling making the best use of teachers in the ensemble. Each teacher model provides dense supervision to the student via dark knowledge (Hinton et al., 2015) using a mechanism similar to label smoothing (Szegedy, Vanhoucke, Ioffe, Shlens, & Wojna, 2016; Müller, Kornblith, & Hinton, 2019), where the amount of smoothing is regulated by the teacher. In our AKD approach, the label smoothing coming from different teachers is combined and regulated, based on the loss incurred by the teacher models during the distillation process.

Experiments on transferring from a collection of six language pairs from IWSLT to five low-resource language-pairs from TED Talks demonstrate the effectiveness of our approach, achieving up to +0.9 BLEU score improvements compared to strong baselines.

4.2 Proposed Method

We address the problem of low-resource NMT, assuming that we have access to models for high-resource languages, and data for low-resource model. Our approach relies on two main steps, (i) Transferring from high-resource to low-resource language-pairs by fine tuning the high-resource models using the small amount of bilingual data, and (ii) Adaptive distillation of knowledge from the teacher models to the student model.

More specifically, given a training dataset for a low-resource language-pair, $\mathcal{D}_{LR} := \{(\boldsymbol{x}_1, \boldsymbol{y}_1), .., (\boldsymbol{x}_n, \boldsymbol{y}_n)\}$ and multiple individual high-resource NMT models $\{\theta^l\}_{l=1}^{L}$ fine-tuned on $\mathcal{D}_{LR}$ (teachers), we are interested in training a single NMT model (student) by adaptively distilling knowledge from all teachers based on their effectiveness to improve the accuracy of the student. Knowledge distillation (KD) is a process of improving the performance of a simple *student* model by using a distribution over soft labels obtained from an expert *teacher* model instead of hard ground-truth labels (Hinton et al., 2015). The training

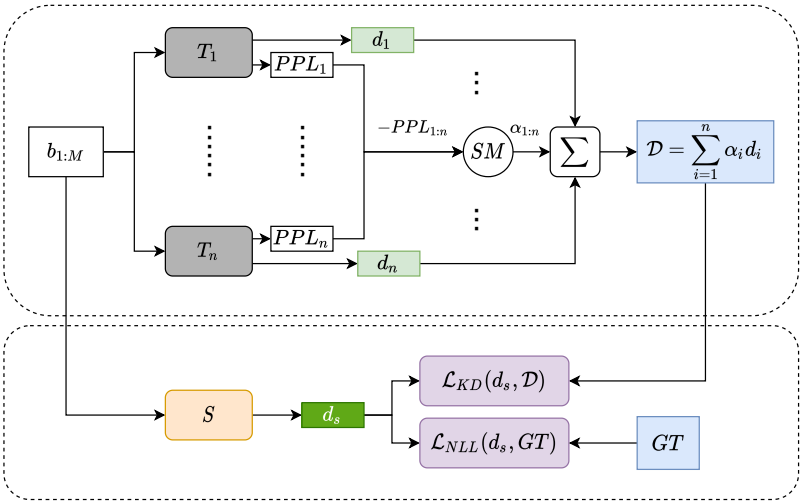

Figure 4.1: Adaptive Knowledge Distillation. **(Top)** Teachers' contribution weight calculation. $T_{1:n}$ and $d_{1:n}$ denote the freezed teacher models and their corresponding probability distributions respectively. **(Bottom)** Training the student with adaptive knowledge distillation. S, SM, and GT denote the student model, softmax function, and ground-truth respectively.

objective to distill the knowledge from a single teacher to the student involves,

$$-\sum_{\boldsymbol{x},\boldsymbol{y}\in\mathcal{D}_{LR}} \sum_{t=1}^{|\boldsymbol{y}|} \sum_{v\in V} Q(v|\boldsymbol{y}_{<t}, \boldsymbol{x}, \theta^l) \log P(v|\boldsymbol{y}_{<t}, \boldsymbol{x}, \theta_{LR}) \tag{4.1}$$

where θ^l and θ_{LR} are the parameters of the teacher and student models, respectively. $P(. \mid .)$ is the conditional probability with the student model and $Q(. \mid .)$ denotes the output distribution of the teacher model. According to Equation 1, knowledge distillation provides dense training signal as *each* word in the vocabulary (V) contributes to the training objective, regulated by a weight coming from the teacher. This is in contrast to the negative log-likelihood training objective, which only provides supervision signal based on the correct target words according to the bilingual training data,

$$\mathcal{L}_{NLL}(\mathcal{D}_{LR}, \theta_{LR}) := -\sum_{\boldsymbol{x},\boldsymbol{y}\in\mathcal{D}_{LR}} \sum_{t=1}^{|\boldsymbol{y}|} \log P(y_t|\boldsymbol{y}_{<t}, \boldsymbol{x}, \theta_{LR}). \tag{4.2}$$

Given a collection of teacher models $\{\theta_l\}_{l=1}^{L}$, we pose the following training objective,

$$\mathcal{L}_{KD}^{adaptive}(\mathcal{D}_{LR}, \theta_{LR}, \{\theta^l\}_1^L, \boldsymbol{\alpha}) := \\ -\sum_{\boldsymbol{x},\boldsymbol{y}\in\mathcal{D}_{LR}} \sum_{l=1}^{L} \alpha_l \sum_{t=1}^{|\boldsymbol{y}|} \sum_{v\in V} Q(v|\boldsymbol{y}_{<t}, \boldsymbol{x}, \theta^l) \log P(v|\boldsymbol{y}_{<t}, \boldsymbol{x}, \theta_{LR}) \tag{4.3}$$

where α_l regulates the contribution of the l-th teacher. We dynamically adjust the contribution weights over the course of the distillation process, in order to effectively address the knowledge gap of the student during the training process. This is achieved based on the rewards (negative perplexity) attained by the teachers on the data, where these values are passed through a softmax transformation to turn into a distribution. To stabilize these contribution weights over the course of the training process, we smooth them using a running geometric average.

The student model is trained end-to-end with a weighted combination of losses coming from the ensemble of teachers and the data,

$$\mathcal{L}_{ALL}^{adaptive}(\mathcal{D}_{LR}, \theta_{LR}, \{\theta^l\}_1^L, \boldsymbol{\alpha}) := \\ \lambda_1 \mathcal{L}_{NLL}(\mathcal{D}_{LR}, \theta_{LR}) + \lambda_2 \mathcal{L}_{KD}^{adaptive}(\mathcal{D}_{LR}, \theta_{LR}, \{\theta^l\}_1^L, \boldsymbol{\alpha}) \tag{4.4}$$

where $\lambda_1 = 0.5$ and λ_2 is started from 0.5 and gradually increased to 3 following the annealing function of (Bowman et al., 2016) in our experiments. Our approach is summarized in Algorithm 4.1 and Figure 4.1.

4.3 Experiments

4.3.1 Settings

Data. We conduct our experiments on the European languages of IWSLT and TED datasets. The language pairs with more than 100K training data are considered as high-resource and the ones less than 15k are assumed as low-resource. The high-resource models are trained on IWSLT2014 (ru, de, it, pl, nl, es-en). IWSLT 2014 MT task data (sl-en) (Cettolo et al., 2014), and TED talk data (gl, et, nb, eu-en) (Qi et al., 2018) are used as low-resource languages. Detail about the preprocessing step and the statistics of data and language codes

Algorithm 4.1: Soft Adaptive Knowledge Distillation

Input : $\mathcal{D}_{LR} := \{(\boldsymbol{x}_1, \boldsymbol{y}_1), .., (\boldsymbol{x}_n, \boldsymbol{y}_n)\}$, low-resource dataset, Individual models $\{\theta^l\}_{l=1}^{L}$ for L language pairs, Total training epochs: N

Output: θ_{LR}: low-resource model

Randomly initialize low-resource model θ_{LR} ;

$n = 0$;

while $n < N$ **do**

  $D_{LR} = random\_permute(\mathcal{D}_{LR})$;

  $\boldsymbol{b}_1, .., \boldsymbol{b}_M = create\_minibatches(\mathcal{D}_{LR})$;

  $m = 1$;

  while $m \leq M$ **do**

    // compute contribution weights;

    for $l \in L$ **do**

      $\Delta_l = -ppl(\theta^l(b_m))$;

    $\boldsymbol{\alpha} = \text{softmax}(\Delta_1, .., \Delta_L)$;

    // compute the gradient ;

    $\boldsymbol{g} = \nabla_{\theta_{LR}} \mathcal{L}_{ALL}^{adaptive}(\boldsymbol{b}_m, \theta_{LR}, \{\theta^l\}_1^L, \boldsymbol{\alpha})$;

    // updates the parameters using the optimiser ADAM ;

    $\theta_{LR} = \text{update\_param}(\theta_{LR}, \boldsymbol{g})$;

    $m = m + 1$;

  $n = n + 1$;

based on ISO 639-1 standard[1] are listed in table 4.1.

Training configuration. Individual low-resource and high-resource NMT models are trained on the low-resource data. The first trained from scratch and the later by finetuning with the vanilla transformer architecture. We used a shared vocabulary across all languages used for teachers and student models to make the knowledge distillation feasible. For multilingual NMT, we train a single model with all high-resource and the up-sampled of low-resource language pairs by using a decoder language embedding layer to identify the type of language during the inference step. Multilingual selective knowledge distillation (Tan, Ren, et al., 2019) is trained with all language pairs while matching the outputs of each low-resource model simultaneously through knowledge distillation. For training our approach, we fine-tune the high-resource models with low-resource languages and treat them as teach-

[1] http://www.loc.gov/standards/iso639-2/php/English_list.php

Algorithm 4.2: Hard Adaptive Knowledge Distillation

Input : $\mathcal{D}_{LR} := \{(\boldsymbol{x}_1, \boldsymbol{y}_1), .., (\boldsymbol{x}_n, \boldsymbol{y}_n)\}$, low-resource dataset, Individual models $\{\theta^l\}_{l=1}^{L}$ for L language pairs, Total training epochs: N

Output: θ_{LR}: low-resource model

Randomly initialize low-resource model θ_{LR} ;

$n = 0$;

while $n < N$ **do**

  $D_{LR} = random\_permute(\mathcal{D}_{LR})$;

  $\boldsymbol{b}_1, .., \boldsymbol{b}_M = create\_minibatches(\mathcal{D}_{LR})$;

  $m = 1$;

  while $m \leq M$ **do**

    // compute contribution weights;

    for $l \in L$ **do**

      $\Delta_l = -ppl(\theta^l(b_m))$;

    $\boldsymbol{\alpha} = \text{softmax}(\Delta_1, .., \Delta_L)$;

    // chooses one teacher by multinomial sampling from the α distribution ;

    $l \sim \boldsymbol{\alpha}$;

    // compute the gradient ;

    $\boldsymbol{g} = \nabla_{\theta_{LR}} \mathcal{L}_{ALL}^{adaptive}(\boldsymbol{b}_m, \theta_{LR}, \theta^l)$;

    // updates the parameters using the optimiser ADAM ;

    $\theta_{LR} = \text{update\_param}(\theta_{LR}, \boldsymbol{g})$;

    $m = m + 1$;

  $n = n + 1$;

ers. When training on the low-resource language, we load teacher models into memory and train a single low-resource model (student) from scratch while using the weighted average of teachers' probabilities based on their contribution weight. In order to make clear how different teachers contribute during training the student, we illustrate contribution weights of all teachers for first 30 iterations of different mini-batches during the training in Figure 4.2. To measure the performance of the teachers, we choose perplexity rather than BLEU[2] unlike (Tan, Ren, et al., 2019), since the perplexity shows how close the teacher's estimated distribution is to that of the ground truth (S. F. Chen & Goodman, 1999). The lower the perplexity, the better teacher estimation.

[2]BLEU score (Papineni, Roukos, Ward, & Zhu, 2002b) aggregates the count of words and phrases (n-grams) that overlap between machine and reference translations and does not measure the confidence of the model regarding the estimated distribution.

High-resource Languages						
Language name	Russian	German	Italian	Spanish	Polish	Dutch
Code	ru	de	it	es	pl	nl
size (#sent(k))	153\6.9\5.5	160\7.2\6.7	167\7.5\5.5	169\7.6\5.5	128\5.8\5.4	153\6.9\5.3
Low-resource Languages						
Language name	Basque	Galician	Norwegian	Slovenian	Estonian	
Code	eu	gl	nb	sl	et	
size (#sent(k))	3.3\0.3\0.3	8.4\0.6\1	14\0.8\0.8	14.5\1.4\0.6	7.7\0.7\1	

Table 4.1: Language names and statistics for bilingual resources (Language → English), (train\dev\test)

Model configuration. All models are trained with Transformer architecture (Vaswani et al., 2017b), with the model hidden size of 256, feed-forward hidden size of 1024, and 2 layers, implemented in Fairseq framework (Ott et al., 2019). We use the Adam optimizer (Kingma & Ba, 2015) and an inverse square root schedule with warm-up (maximum LR 0.0005). We apply dropout and label smoothing with a rate of 0.3 and 0.1 respectively. The source and target embeddings are shared and tied with the last layer. We train with half-precision floats on one V100 GPU, with at most 4028 tokens per batch.

4.3.2 Results

In Table 4.2, we compare our approach with individual NMT models, transferred models from high-resource language pairs, multilingual NMT, and multilingual selective knowledge distillation (Tan, Ren, et al., 2019). We selected the best models according to the SacreBLEU[3] score on the validation set. In our experiments, bold numbers indicate the best results and underlined numbers show the second best ones. According to the conducted experiments, transfer learning results are inline with the language family relationships (Littell et al., 2017b). The high-resource languages which are linguistically close to the low-resource languages have the most impact on low-resource model's improvement. Likewise, the contribution weights of different teachers are consistent with the performance

[3]SacreBLEU signature: BLEU+case.mixed+numrefs.1+ smooth.exp+tok.none+version.1.3.1

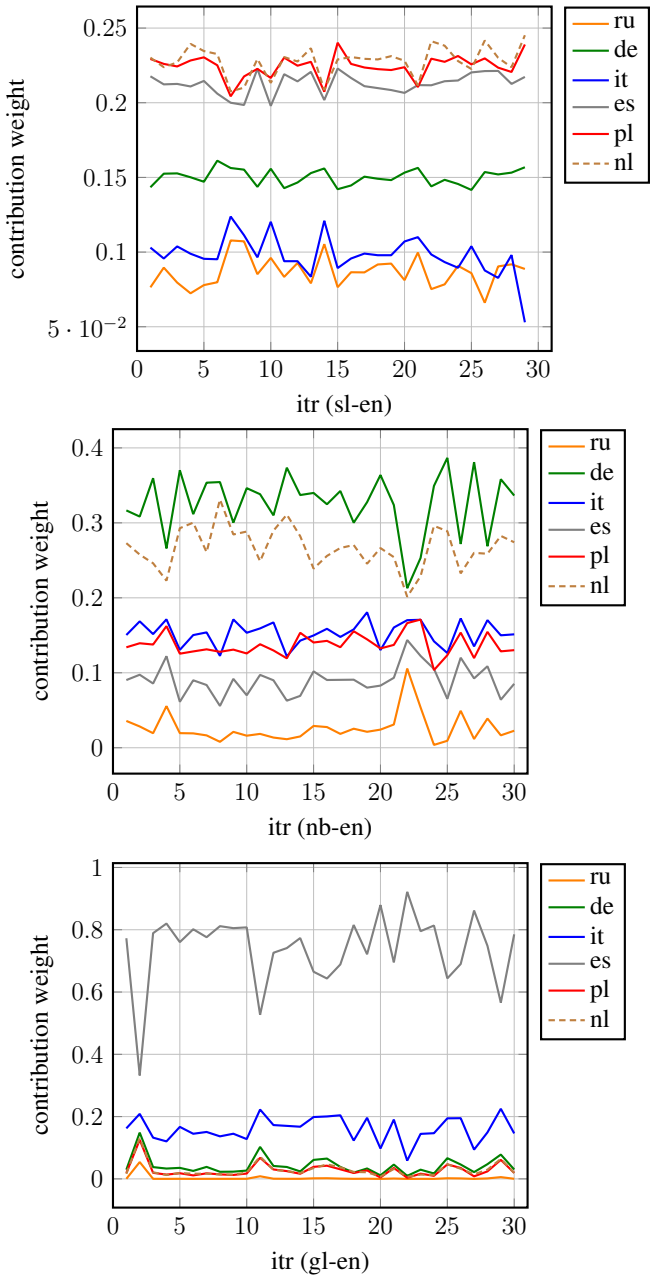

Figure 4.2: Teachers' contribution weights during the training of low-resource NMT models for "sl-en", "gl-en", and "nb-en" language pairs, first 30 iterations for different mini-batches.

of the teachers as hypothesized (See Figure 4.2). According to Table 4.2, the multilingual models (with and without knowledge distillation) are less accurate than at least one of the transferred models from high-resource languages[4]. This suggests a weak link may exist

[4]Except for the Basque language which is extremely low-resource and is linguistically as distant to all the languages in the multilingual setting.

MT Task	Individual	Individual Teachers						Multi-Lingual		Multi-Teacher
x→en	student	ru	de	it	es	pl	nl	Uniform	Selec. KD	Adap. KD
sl	10.58	10.36	14.09	13.29	16.89	<u>17.63</u>	16.67	15.97	16.17	**18.35**
nb	26.38	32.24	32.77	31.90	30.04	30.66	<u>32.86</u>	30.06	31.08	**33.72**
gl	13.87	11.88	17.66	21.90	**27.49**	16.67	17.05	<u>25.27</u>	25.08	24.50
eu	6.50	9.54	10.68	9.92	11.00	10.50	10.02	10.11	<u>11.03</u>	**11.38**
et	10.15	12.18	14.85	14.93	<u>15.53</u>	14.25	13.66	14.91	15.15	**16.20**

Table 4.2: BLEU scores of the translation tasks from five languages into English. Our approach (last column) is compared with individual NMT models, transferred models from high-resource language pairs (individual teacher models), multilingual uniform NMT, and multilingual selective knowledge distillation (Tan, Ren, et al., 2019) The bold numbers show the best result and the underlined numbers indicate the second-best results.

between the impact of each high-resource language and its contribution during the training multilingually. Adaptive knowledge distillation compensates this blind collaboration between teachers by weighting the teachers' contributions particularly for the cases where majority of teachers and student are linguistically close such as "nb-en". The qualitative examples are presented in Section 4.4.3. It is worth noting that, we empirically observed when there is more diversity in teachers (e.g, in case of "gl-en" in Table 4.2), adaptive KD underperforms compared to the best teacher and we hypothesise this happens because there is an empirically dominant teacher ("es"). This observation suggests that a prior effort for choosing the proper teacher languages (e.g., based on the language family information) will directly impact the performance of the low-resource NMT model.

4.4 Analysis

This section analyses our result more in detail based on three essential factors: Contribution weight policy, contribution temperature scaling, and finally, the generated translation quality.

4.4.1 Contribution Weight Analysis

To analyse the effect of teachers' contribution weights, we compare three different contribution settings:

- *(i) Soft adaptive contribution:* This contribution policy assigns the contribution weights to all the teachers based on their performance per mini-batch as explained in Section 4.2 and Algorithm 4.1.
- *(ii) Hard adaptive contribution:* This contribution policy chooses one teacher by multinomial sampling from the α distribution per mini-batch (See Algorithm 4.2).
- *(iii) Equal contribution:* This contribution policy gives simply all the teachers the same contribution weights.

According to Table 4.3, the worst contribution weight setting is for *equal contribution* especially for the languages with more inconsistent teachers (based on BLEU score) e.g., "gl-en" where *hard adaptive* performs better. However, in other cases with more consistent teachers (based on BLEU score), *soft adaptive* performs the best (e.g.,"nb-en").

Contribution weight setting	gl-en	nb-en
Soft Adaptive contribution	24.50	**33.72**
Hard adaptive contribution	**25.67**	33.55
Equal contribution	19.10	32.60

Table 4.3: Effect of different contribution settings.

4.4.2 Contribution Temperature Scaling

Through the experiments, we observed that when most of the teachers do not agree (in terms of perplexity), a constant temperature is not an ideal option. An alternative is to adaptively change the value of the temperature given the *agreement* among the teachers determined based on the distance between the maximum and minimum perplexity between teachers

which can be formulated as:

$$\tau = \frac{1 - (\max(S) - \min(S))}{L} \tag{4.5}$$

where S is the output of the softmax operation on the negative perplexity of all L teachers and (max(S)−min(S)) is inversely proportional to the extent of the agreement between teachers. Such temperature scaling encourages the contribution of better teachers in case of the existence of a disagreement, while it allows similar contributions when all teachers agree on a mini-batch. Table 4.4 shows the effect of adaptive temperature for two languages.

Contribution temperature	eu-en	sl-en
with adaptive temp	**11.38**	**18.35**
without temp	10.52	18.05

Table 4.4: Effect of adaptive temperature.

4.4.3 Translation Examples

Table 4.5 showcases the generated English translations by the individual student, all the teachers, and student trained through adaptive knowledge distillation from Norwegian language. This example shows that while there is a diversity between different teachers' translations e.g., for the verb of *"provoke"*, the student is impacted by the agreement of the majority of teachers. Moreover, this example shows that our adaptive KD model captures the best of all teachers resulting in a higher quality translation.

4.5 Conclusion

In this chapter, we presented an adaptive knowledge distillation approach to improve NMT for low-resource languages. We addressed the inefficiency of the original transfer learning and multilingual learning by making wiser use of all high-resource languages and models in an effective collaborative learning manner. Our approach shows its effectiveness in the translation of low-resource languages, especially when there is complementary knowledge

Model	Translation
Ref	And great creativity is needed to do what it does so well : to provoke us to think differently with dramatic creative statements .
Individual	kepler **great** mission mission to do it as **well** : to grow us to think **with dramatic** creativity .
Teacher (ru-en)	and the first **creativity** needed **to do what it does** : to promote us to think about the **dramatic** creativity .
Teacher (de-en)	now , the future **creativity** needs to do it as it does : to **provoke** us to **think differently with dramatic** creative expression .
Teacher (it-en)	now , the future **creativity** is needed **to do what it does** so **well** : to provocate us to **think differently** about **dramatic** reactive .
Teacher (es-en)	the future of **creativity** to do that as it's doing so good : to provocate us to **think differently** about **dramatic** creativity .
Teacher (pl-en)	the future of **creativity to do what it does** so good : to promise others **with dramatic** creativity .
Teacher (nl-en)	now , the frequent **creativity** is to make it that it makes so good : to **provoke** us with **dramatic** creative .
Proposed Adapt. KD	now , they need **great creativity to do what it does** so **well** : **provoke** us to **think differently with dramatic** creativity.

Table 4.5: The generated outputs from the individual student, all teachers, and student trained with multi-teachers (Proposed Adapt. KD) for "nb-en" MT task. Some of the correct keyword translations are indicated with green colour while hallucinations are represented by red. The bold-green shows the best of the teachers' output which is also captured with the student.

in multiple high-resource languages from the same linguistic family and it is not explicitly clear which language has more impact in every mini-batch of low-resource training data. Experiments on the translation of five extremely low-resource languages to English show improvements compared to the strong baselines.

5 | Multilingual NMT with Hierarchical Knowledge Distillation

Following the previous chapter, we continue studying the problem of low-resource Neural Machine Translation using knowledge distillation with the focus on using multiple languages in a multilingual translation regime while avoiding the negative transfer. Multilingual Neural Machine Translation (MNMT) trains a single NMT model that supports translation between multiple languages, rather than training separate models for different languages. Learning a single model can enhance the low-resource translation by leveraging data from multiple languages pairs in a unified training process. However, the performance of an MNMT model is highly dependent on the type of languages used in training, as transferring knowledge from a diverse set of languages degrades the translation performance due to negative transfer. In this chapter, we propose a Hierarchical Knowledge Distillation (HKD) approach for MNMT which capitalises on language groups generated according to typological features and phylogeny of languages to overcome the issue of negative transfer. HKD generates a set of multilingual teacher-assistant models via a selective knowledge distillation mechanism based on the language groups, and then distills the ultimate multilingual model from those assistants in an adaptive way. Experimental results derived from the TED dataset with 53 languages demonstrate the effectiveness of our approach in avoiding the negative transfer effect in MNMT, leading to an improved translation performance (about 1

BLEU score on average) compared to strong baselines. In summary, this chapter serves a two-fold purpose: to demonstrate how hierarchical knowledge distillation is helpful to address the low-resource challenges in multilingual setting, and how cluster-based teachers in a hierarchy are effective to avoid negative transfer.

5.1 Introduction

The surge over the past few decades in the number of languages used in electronic texts for international communications has promoted Machine Translation (MT) systems to shift towards multilingualism. However, most successful MT applications, i.e., Neural Machine Translation (NMT) systems, usually rely on supervised deep learning, which is notoriously data-hungry (Koehn & Knowles, 2017). Despite decades of research, high-quality annotated MT resources are only available for a subset of the world's thousands of languages (Paolillo & Das, 2006). Hence, data scarcity is one of the significant challenges which comes along with the language diversity and multilingualism in MT. One of the most widely-researched approaches to tackle this problem is unsupervised learning which takes advantage of available unlabeled data in multiple languages (Lample, Conneau, Denoyer, & Ranzato, 2017; Arivazhagan et al., 2019; Snyder et al., 2010; Xu, Qin, Wang, & Liu, 2019). However, unsupervised approaches have relatively lower performance compared to their supervised counterparts (Dabre et al., 2020). Nevertheless, the performance of the supervised MNMT models is highly dependent on the types of languages used to train the model (Tan, Chen, et al., 2019). If languages are from very distant language families, they can lead to *negative transfer* (Torrey & Shavlik, 2010; Rosenstein, 2005), causing lower translation quality compared to the individual bilingual counterparts.

To address this problem, some improvements have been achieved recently with solutions that employ some sort of supervision to guide MNMT using *linguistic typology* (Oncevay et al., 2020; Chowdhury, España-Bonet, & van Genabith, 2020; Kudugunta, Bapna, Caswell, & Firat, 2019; Bjerva, Östling, Veiga, Tiedemann, & Augenstein, 2019). The linguistic typology provides this supervision by treating the world's languages based on

their functional and structural characteristics (O'Horan et al., 2016). Taking advantage of this property, which explains both language similarity and language diversity, we aim in our approach to combine two solutions for training an MNMT model: (a) creating a universal, language-independent MNMT model (Johnson et al., 2017); (b) systematically designing the possible variations of language-dependent MNMT models based on the language relations (Maimaiti et al., 2019).

Our approach to preventing negative transfer in MNMT is to group models which behave similarly in separate language clusters. Then, we perform a Knowledge Distillation (KD) (Hinton et al., 2015) approach by selectively distilling the bilingual teacher models' knowledge in the same language cluster to a multilingual teacher-assistant model. The intermediate teacher-assistant models are representative of their own language cluster. We further adaptively distill knowledge from the multilingual teacher-assistant models to the ultimate multilingual student. In summary, our main contributions are as follows:

- We use *cluster-based teachers* in a hierarchical knowledge distillation approach to prevent negative transfer in MNMT. Different from the previous cluster-based approaches in multilingual settings (Oncevay et al., 2020; Tan, Chen, et al., 2019), our approach makes use of all the clusters with a *universal* MNMT model while retaining the language relatedness structure in a hierarchy.
- We distill the ultimate MNMT model from multilingual teacher-assistant models, each of which represents one language family and usually perform better than the individual bilingual models from the same language family. Thus, the cluster-based teacher-assistant models can lead to a better knowledge distillation compared to a diverse set of bilingual teacher models as used in multilingual KD (Tan, Ren, et al., 2019).
- We explore *a mixture of linguistic features* by utilizing different clustering approaches to obtain the cluster-based teacher-assistants. As the language groups created by different language feature vectors can contribute differently to translation, we adaptively distill knowledge from teacher-assistant models to the ultimate student to improve the knowledge gap of the student.

- We perform extensive experiments on 53 languages, showing the effectiveness of our approach in avoiding negative transfer in MNMT, leading to an improved translation performance (about 1 BLEU score on average) compared to strong baselines. We also conduct comprehensive ablation studies and analysis, demonstrating the impact of language clustering in MNMT for different language families and in different resource-size scenarios.

5.2 Related Work

The majority of works on MNMT mainly focus on different architectural choices varying in the degree of parameter sharing in the multilingual setting. For example, the works based on the idea of minimal parameter sharing share either encoder, decoder, or attention module (Firat et al., 2017; Lu et al., 2018), and those with complete parameter sharing tend to share entire models (Johnson et al., 2017; Ha et al., n.d.). In general, these techniques implicitly assume that a set of languages is pre-given without considering the positive or negative effect of language transfer between the languages shared in one model. Hence, they can usually achieve comparable results with individual models (trained with individual language pairs) only when the languages are less diverse or the number of languages is small. When several diverse language pairs are involved in training an MNMT system, the *negative transfer* (Torrey & Shavlik, 2010; Rosenstein, 2005) usually happens between more distant languages, resulting in degraded translation accuracy in the multilingual setting. To address this problem, Tan, Chen, et al. (2019) suggested a clustering approach using either prior knowledge of language families or using language embedding. They obtained the language embedding by retrieving the representation of a language tag which is added to the input of an encoder in a universal MNMT model. Later, Oncevay et al. (2020) introduced another clustering technique using the multi-view language representation. They fused language embeddings learned in an MNMT model with syntactic features of a linguistic knowledge base (Dryer & Haspelmath, 2013). Tan, Ren, et al. (2019) proposed a knowledge distillation approach which transfers knowledge from bilingual teachers to a multilingual student when the accuracy of teachers are higher than the student. Their approach eliminates the accuracy

gap between the bilingual and multilingual NMT models. However, we argue that distilling knowledge from a *diverse* set of parent models into a student model can be sub-optimal, as the parents may compete instead of collaborating with each other, resulting in negative transfer due to language discrepancy.

5.3 Technical Background

As we already discussed the technical aspects related to Neural Machine Translation and Knowledge Distillation in Chapter 2, here we only focus on the background which is technically related linguistic topology and language clustering.

5.3.1 Linguistic Typology

Language, a structured system of communication used by humans, evolved and diversified over time. The language divisions are mostly developed arbitrarily or based on the political or geographical basis (e.g., "German", "Japanese", "Hindi"). In a scientific study of language, called *linguistic typology*, languages have been studied and classified based on "functional" and "structural" properties to explain both the similarities and the structural diversity of languages (Campbell, 2013; O'Horan et al., 2016). One of the simple and early used property by typologists was word order in different languages like SVO (Subject, Verb, Object), VSO, SOV, etc. This property then extended to a wide range of features such as phonological, semantic, lexical, and morphosyntactic properties [1].

The linguistic study can be categorized based on three typologies: i) *Qualitative typology*, which defines the language features and their diversity. ii) *Quantitative typology*, which measures and analyzes the linguistic features across empirical data, and iii) *Theoretical typology*, which explains the pattern observed in qualitative typology (Bickel, 2007).

The theoretical typology is not always in line with authenticated theories of language

[1] We direct the interested readers to (Bickel, 2007; Daniel, 2011) for an in-depth overview about these features.

relations based on geographical or historical parameters. The typological classification of languages supports the linguistic theories of causation, such as historical, areal or phylogenetic relations, however, these causation hypotheses come after theoretical typology derived from the measurements and analyses of the linguistic features across empirical data (Bickel, 2007). Therefore, Turkish and Korean, which are usually considered as highly dissimilar languages based on lexical features, are categorized as structurally similar based on syntactic features. Such deep and abstract evidence of similarity is of high value for training deep neural networks in NLP, which essentially tries to model cross-linguistic relations and variations rather than explain language relations (O'Horan et al., 2016).

Although, much invaluable information conducted in linguistic studies is not usable by NLP due to the inconsistent definitions across languages, there are some publicly accessible databases suitable for NLP applications. O'Horan et al. (2016) introduced the most important available databases in their survey on the use of typological information in NLP and we also show their list here in Table 5.1 for further reference. The table represents some basic information such as type, coverage, and additional notes about the following databases: Syntactic Structures of the World's Languages (SSWL) (C. Collins & Kayne, 2009), the World Atlas of Language Structures (WALS) (Dryer & Haspelmath, 2013) which is the most popular and commonly-used typological resource in NLP, the Phonetics Information Base and Lexicon (PHOIBLE) (Moran & McCloy, 2019), the URIEL Typological Compendium (Littell et al., 2017a), the Atlas of Pidgin and Creole Language Structures (APiCS) (Michaelis et al., 2013), and the Lyon-Albuquerque Phonological Systems Database (LAPSyD) (Maddieson et al., 2013).

Thanks to the aforementioned available databases, in recent years, linguistic typology either explicitly or implicitly has widely used in many of the current popular solutions to multilingual NLP, such as: transfer learning from high-resource to low-resource languages (Padó & Lapata, 2005; Khapra, Joshi, Chatterjee, & Bhattacharyya, 2011; Das & Petrov, 2011; Täckström, McDonald, & Uszkoreit, 2012), joint multilingual learning (Snyder et al., 2010; S. B. Cohen, Das, & Smith, 2011; Navigli & Ponzetto, 2012), and development of

Accessible Databases in Linguistic Typology			
Name	Type	Coverage	Notes
WALS	Phonology Morphosyntax Lexico Semantics	2676 languages; 192 features; 17% of features have values	Defines language features and provides values for a large set of languages; originally intended for study of a real distribution of features.
SSWL	Morphosyntax	262 languages; 148features; 45% of features have values	Similar to WALS, but differs in being fully open to public editing (Wikipedia-style), and by the addition of numerous example sentences for each feature
APiCS	Phonology Morphosyntax Lexicosemantics	76 languages; 130 features; 18526 examples	Designed to allow comparison with WALS
LAPSyD	Phonology	422 languages	Documents a broader range of features than PHOIBLE, including syllable structures and tone systems; provides bibliographic information and links to recorded samples
PHOIBLE	Phonology	1672 languages; 2160 segments	Collates and standardises several phonological segmentation databases, in addition to new data
URIEL	Phonology Morphosyntax Lexicosemantics	8070 languages/dialects; 284 features; 439000 feature values	Collates features from WALS, SSWL, PHOIBLE, and 'geodata' (e.g. language names, ISO codes,etc.) from sources such as Glottolog and Ethnologue; includes cross-lingual distance measures based on typological features; provides estimates for empty feature values

Table 5.1: An overview of the most commonly used publicly available databases of typological information suitable for application in NLP (O'Horan et al., 2016). The full name and references of the abbreviations are as follow: the World Atlas of Language Structures (WALS) (Dryer & Haspelmath, 2013), the Syntactic Structures of the World's Languages (SSWL) (C. Collins & Kayne, 2009), the Atlas of Pidgin and Creole Language Structures (APiCS) (Michaelis et al., 2013), the Lyon-Albuquerque Phonological Systems Database (LAPSyD) (Maddieson et al., 2013), the Phonetics Information Base and Lexicon (PHOIBLE) (Moran & McCloy, 2019), and the URIEL Typological Compendium (Littell et al., 2017a).

universal models (De Marneffe et al., 2014; Nivre et al., 2016). On the other hand, the neural language models trained on a multilingual text corpus can be used to induce distributed representations of languages and capture the abstract features of languages which are not available in KBs. This is a case which shows how NLP and linguistic typology can interact and benefit one another and lead the future of NLP and linguistic-related sciences.

5.3.2 Language Clustering

The goal of language clustering and language ranking is to choose similar and dominant languages to perform the best transfer learning in different tasks, such as multilingual NMT. Language vectors are essential properties to do the language clustering. There are three general ways to define language vectors in NLP:

Spars language vectors from typological knowledge bases (KB) database. Linguistic typology does support and investigate language variation based on their functional and structural features. Categorical language features are obtainable from several typological knowledge bases (KB) that have been created and publicly available (Littell et al., 2017b) (See table 5.1). Nevertheless, the sparsity and heterogeneity in those KB databases in format, semantics, language, and feature naming is still an open challenge for integrating them to end-to-end NLP algorithms. For example, the World Atlas of Language Structure (Dryer & Haspelmath, 2013) which is currently the most commonly-used typological resource in NLP due to its broad coverage of features and languages, has only a mean coverage of 14% per language.

Dense learned language embedding vectors from multilingual NLP tasks. To overcome the challenges of using sparse language vectors, dense data-driven language representations have emerged. These language representations are computed from multilingual NLP tasks like language modelling (Östling & Tiedemann, 2016) or neural machine translation (NMT) (Malaviya, Neubig, & Littell, 2017). Unlike the sparse language vectors, which treat languages as discrete categories, dense language embedding vectors allow the interpolation between languages and capture the linguistic relations. However, due to the limitation of available corpora, language diversity in the task-learned representations is limited.

The mix of KB and task-learned language vectors. To leverage the best of both views (KB and task-learned) with minimal information loss, (Oncevay et al., 2020) fuse both views using singular vector canonical correlation analysis and investigate how the sparse and dense

language vectors from both views can benefit from each other. They project a shared space of discrete and continuous features using a variant of canonical correlation analysis (Raghu, Gilmer, Yosinski, & Sohl-Dickstein, 2017).

In this work, we take advantage of all of the above language vectors for clustering similar languages. We train the multilingual expert student in our knowledge distillation approach while distilling knowledge from similar languages in the clusters. Using four different clustering approaches obtained from different language vectors allows us to access to four clusters for each language representing different features per language.

5.4 Hierarchical Knowledge Distillation

We address the problem of *data scarcity* and *negative transfer* in MNMT with a *Hierarchical Knowledge Distillation* (HKD) approach. The hierarchy in HKD is constructed in such a way that the node structure captures the similarity structure and the relatedness of the languages. Specifically, in an inverse pyramidal structure as shown in Figure 5.1, the root node corresponds to the ultimate MNMT model that we aim to train, the leave nodes correspond to each individual bilingual NMT models, and the non-terminal nodes represent the language clusters. Our hypothesis is that leveraging common characteristics of languages in the same language group, which is formed using clustering algorithms based on the typological properties of languages (O'Horan et al., 2016), the HKD method can train a high quality MNMT model by distilling knowledge from related languages, rather than diverse ones.

Our HKD approach consists of two knowledge distillation mechanisms, providing two levels of supervision for training the ultimate MNMT model (illustrated in Figure 5.1), including: (i) **selective distillation** of knowledge from individual bilingual teachers to the multilingual intermediate teacher-assistants, each of which corresponds to one language group; and (ii) **adaptive distillation** of knowledge from all related cluster-wise teacher-assistants to the super-multilingual ultimate student model in each mini-batch of training

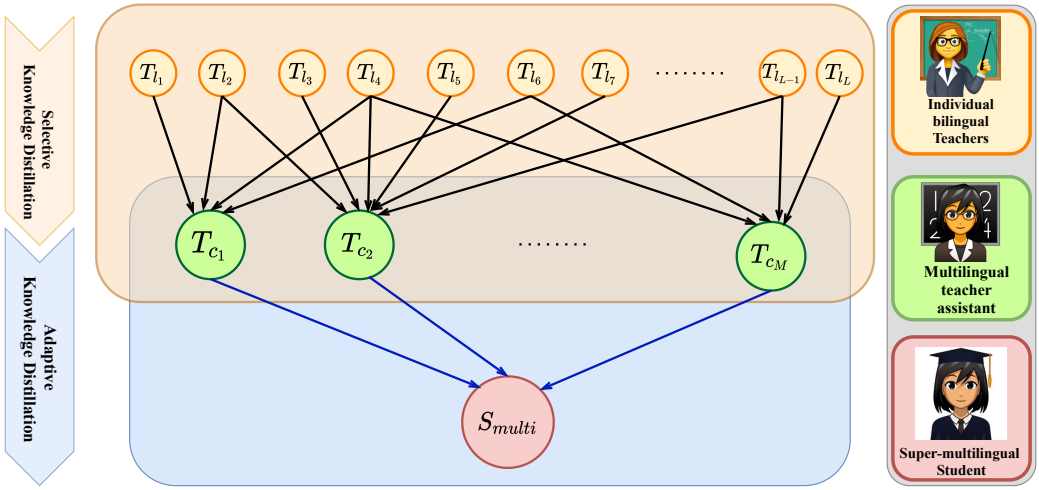

Figure 5.1: **HKD** approach: In the first phase of knowledge distillation, aka "Selective KD", the knowledge is transferred from bilingual teacher models per clusters (orange circles) to the multilingual teacher-assistant models (green circles). For example T_{l_1}, T_{l_2}, T_{l_4}, and T_{l_6} are belonged to one cluster and distilled to teacher-assistant model T_{c_1}. In the second KD phase, aka "Adaptive KD", knowledge is transferred from ensemble of intermediate related teacher-assistant models to the ultimate student (red circle) adaptively.

per language pair. Note that we do not utilize multilingual adaptive KD in both distillation phases as we need to have the predictions of all the *relevant experts* in adaptive KD. Using adaptive KD for both stages is particularly impractical when there is a huge set of diverse teachers as in the first phase. Hence, in the first distillation phase, we aim to generate the cluster-wise teacher assistants using selective KD as the pre-requisites for the adaptive KD phase. The main steps of HKD are elaborated as follows:

Clustering: Clustering can be conducted using different language vectors such as: i) sparse language vectors from typological knowledge base (KB) databases, ii) dense learned language embedding vectors from multilingual NLP tasks, and iii) the combination of KB and task-learned language vectors. The implicit causal relationships between languages are usually learned from translation tasks; the genetic, the geographical, and the structural similarities between languages are extracted from typlogical KBs (Bjerva et al., 2019). Thus, the language groups created by different language vectors can contribute differently to the translation and it is not quite clear which types of language features are more helpful in MNMT systems (Oncevay et al., 2020). For example, *"Greek"* can be clustered with *"Ara-*

bic" and *"Hebrew"* based on the mix of KB and task-learned language vectors. Meanwhile, it can be clustered with *"Macedonian"* and *"Bulgarian"* based on NMT-learned language vectors. Therefor, we cluster the languages based on all types of language representations and propose to explore a mixture of linguistic features by utilizing all clusters in training the ultimate MNMT student. So, given a training dataset consisting of L languages and K clustering approaches, where each clustering approach creates n clusters, we are interested in training a many-to-one MNMT model (ultimate student) by hierarchically distilling knowledge from all M clusters to the ultimate student, where $M := \sum_{k=1}^{K} n_k$.

Multilingual selective knowledge distillation: Assume we have a language cluster that consists of L' languages, where $l \in \{1, 2, \ldots, L'\}$. Given a collection of pretrained individual teacher models $\{\theta^l\}_{l=1}^{L'}$, each handling one language pair in $\{\mathcal{D}^l\}_{l=1}^{L'}$, and inspired by (Tan, Ren, et al., 2019), we use the following knowledge distillation objective for each language l in the cluster.

$$\begin{gathered}\mathcal{L}_{KD}^{selective}(\mathcal{D}^l, \theta^c, \theta^l) := \\ -\sum_{\boldsymbol{x},\boldsymbol{y}\in\mathcal{D}^l} \sum_{t=1}^{|\boldsymbol{y}|} \sum_{v\in V} Q(v|\boldsymbol{y}_{<t}, \boldsymbol{x}, \theta^l) \log P(v|\boldsymbol{y}_{<t}, \boldsymbol{x}, \theta^c)\end{gathered} \tag{5.1}$$

where θ^c is the teacher assistant model, $|V|$ is the vocabulary set, $P(\cdot \mid \cdot)$ is the conditional probability of the teacher assistant model, and $Q(\cdot \mid \cdot)$ denotes the output distribution of the bilingual teacher model. According to Eq. (5.1), knowledge distillation regularises the predictive probabilities generated by a cluster-wise multilingual model with those generated by each individual bilingual models. Together with the translation loss ($\mathcal{L}_{NLL}$), we have the following selective KD loss to generate the intermediate teacher-assistant model:

$$\begin{gathered}\mathcal{L}_{ALL}^{selective}(\mathcal{D}^l, \theta^c, \theta^l) := \\ (1-\lambda)\mathcal{L}_{NLL}(\mathcal{D}^l, \theta^c) + \lambda\mathcal{L}_{KD}^{selective}(\mathcal{D}^l, \theta^c, \theta^l)\end{gathered} \tag{5.2}$$

where λ is a tuning parameter that balances the contribution of the two losses. Instead of using all language pairs in Eq (5.1), we used a deterministic but dynamic approach to exclude language pairs from the loss function if the multilingual student surpasses the individual

Algorithm 5.1: Multilingual Selective Knowledge Distillation (Tan, Ren, et al., 2019)

Input : Training corpora: $\{\mathcal{D}^l\}_{l=1}^{L}$; where $\mathcal{D}^l := \{(\boldsymbol{x}_1^l, \boldsymbol{y}_1), .., (\boldsymbol{x}_n^l, \boldsymbol{y}_n)\}$;
List of all languages: L;
Individual models $\{\theta^l\}_{l=1}^{L'}$;
List of language pairs per cluster: L' ;
Total training epochs: N;
Distillation check step: $\mathcal{N}_{check}$;
Threshold of distillation accuracy: $\mathcal{T}$

Output : θ^c: multilingual model for each cluster,

Randomly initialize multilingual model θ^c, accumulated gradient $g = 0$, distillation flag $f^l = True$ for $l \in L'$;
$n = 0$;
while $n < N$ **do**
  $g = 0$;
  for $l \in L'$ **do**
    $D^l = random\_permute(\mathcal{D}^l)$;
    $\boldsymbol{b}_1^l, .., \boldsymbol{b}_J^l = create\_minibatches(\mathcal{D}^l)$,where $b^l = (x^l, y)$;
    $j = 1$;
    while $j \leq J$ **do**
      if $f^l == True$ **then**
        //compute and accumulate the gradient on loss $\mathcal{L}_{ALL}^{selective}$;
        $\boldsymbol{g} = \nabla_{\theta^c} \mathcal{L}_{ALL}^{selective}(\boldsymbol{b}_j^l, \theta^c, \theta^l)$;
        // updates the parameters using the optimiser ADAM ;
        θ^c = update_param($\theta^c, \boldsymbol{g}$) ;
      else
        //compute and accumulate the gradient on loss $\mathcal{L}_{NLL}$;
        $\boldsymbol{g} = \nabla_{\theta^c} \mathcal{L}_{NLL}(\boldsymbol{b}_j^l, \theta^c, \theta^l)$;
        // updates the parameters using the optimiser ADAM ;
        θ^c = update_param($\theta^c, \boldsymbol{g}$) ;
    $j = j + 1$;
  if $N\%\mathcal{N}_{check} == 0$ **then**
    for $l \in L'$ **do**
      if $Accuracy(\theta^c) < Accuracy(\theta^l) + \mathcal{T}$ **then**
        $f^l = True$
      else
        $f^l = False$
  $n = n + 1$;

models on some language pairs during the training, which makes the training selective.

This selective distillation process[2] is applied to all clusters obtained from different clustering approaches. It is noteworthy that (i) the selective knowledge distillation generates a teacher-assistant model for each cluster, i.e., $c \in \{1, 2, \ldots, M\}$; (ii) each language can be in multiple clusters due to the use of different language representations, thus there can be more than one effective teacher-assistant model for any given language pair (illustrated in Fig-

[2]The training algorithm of selective knowledge distillation is summarized in Alg. 5.1, which is similar to the one used in (Tan, Ren, et al., 2019).

ure 5.2). So for each language pair, we have a set of effective clusters: $c \in \{1, 2, \ldots, C_{sim}\}$.

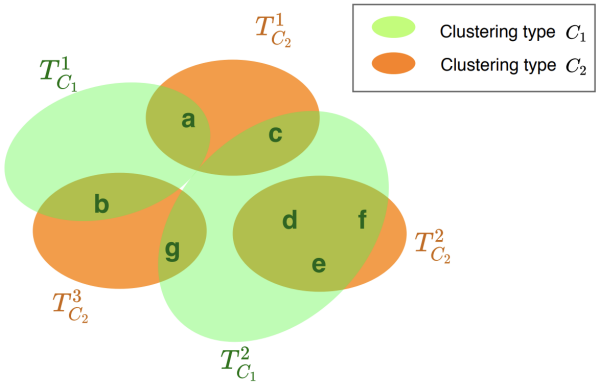

Figure 5.2: Effective teachers for each language after clustering. C refers to the clustering type and T refers to the Teacher. For language ***a***, we have two ***effective*** teachers: $T^1_{C_1}$ and $T^1_{C_2}$.

Multilingual adaptive knowledge distillation: Given a collection of effective teacher-assistant models $\{\theta^c\}_{c=1}^{C_{sim}}$, where C_{sim} is the number of effective clusters per language, we devise the following KD objective for each language pair,

$$\begin{aligned} &\mathcal{L}_{KD}^{adaptive}(\mathcal{D}^l, \theta_s, \{\theta^c\}_1^{C_{sim}}, \boldsymbol{\alpha}) := \\ &-\sum_{c=1}^{C_{sim}} \sum_{\boldsymbol{x},\boldsymbol{y}\in\mathcal{D}^l} \alpha_c \sum_{t=1}^{|\boldsymbol{y}|} \sum_{v\in V} Q(v|\boldsymbol{y}_{<t}, \boldsymbol{x}, \theta^c) \log P(v|\boldsymbol{y}_{<t}, \boldsymbol{x}, \theta_s) \end{aligned} \tag{5.3}$$

where $\boldsymbol{\alpha}$ dynamically weigh the contribution of the teacher-assistants/clusters. $\boldsymbol{\alpha}$ is computed via an attention mechanism based on the rewards (negative perplexity) attained by the teachers on the data, where these values are passed through a softmax transformation to turn into a distribution (Saleh et al., 2020). This adaptive distillation of knowledge allows the student model to get the best of teacher-assistants (which are representative of different linguistic features) based on their effectiveness to improve the knowledge gap of the student. The total loss function then becomes a weighted combination of losses coming from the ensemble of teachers and the data,

$$\mathcal{L}_{ALL}^{adaptive}(\mathcal{D}^l\,\theta_s, \{\theta^c\}_1^{C_{sim}}, \boldsymbol{\alpha}) := \\ \lambda_1 \mathcal{L}_{NLL}(\mathcal{D}^l, \theta_s) + \lambda_2 \mathcal{L}_{KD}^{adaptive}(\mathcal{D}^l, \theta_s, \{\theta^c\}_1^{C_{sim}}, \boldsymbol{\alpha}) \tag{5.4}$$

The training process is summarized in Alg. 5.2.

Algorithm 5.2: Multilingual Adaptive KD

Input : Training corpora: $\{\mathcal{D}^l\}_{l=1}^{L}$, where $\mathcal{D}^l := \{(\boldsymbol{x}_1^l, \boldsymbol{y}_1), .., (\boldsymbol{x}_n^l, \boldsymbol{y}_n)\}$;
List of languages:L;
List of language clusters: $\{C^m\}_{m=1}^{M}$;
Cluster-based MNMT models: $\{\theta^c\}_{c=1}^{M}$;
Total training epochs: N;

Output : Ultimate multilingual student model: θ_s;

Randomly initialize multilingual model θ_s, accumulated gradient $g = 0$, distillation flag $f^l = True$ for $l \in L$;
$n = 0$;
while $n < N$ **do**
  $g = 0$;
  $C_{sim} = []$;
  for $l \in L$ **do**
    // find the effective clusters with similar languages;
    for $c \in \{C\}_1^M$ **do**
      if $l \in c$ **then**
        C_{sim}.append(c)
    $D^l = random\_permute(\mathcal{D}^l)$;
    $\boldsymbol{b}_1^l, .., \boldsymbol{b}_J^l = create\_minibatches(\mathcal{D}^l)$
    //where $b^l = (x^l, y)$;
    $j = 1$;
    while $j \leq J$ **do**
      // compute contribution weights;
      for $c \in C_{sim}$ **do**
        $\Delta_c = -ppl(\theta^c(b_j^l))$;
      $\boldsymbol{\alpha} = \text{softmax}(\Delta_1, .., \Delta_c)$;
      //compute the gradient on loss $\mathcal{L}_{ALL}^{adapt.}$;
      $\boldsymbol{g} = \nabla_{\theta_s} \mathcal{L}_{ALL}^{adapt.}(\boldsymbol{b}_j^l, \theta_s, \{\theta^c\}_1^C, \boldsymbol{\alpha})$;
      // updates the parameters ;
      $\theta_s = \text{update\_param}(\theta_s, \boldsymbol{g})$;
      $j = j + 1$;
  $n = n + 1$;

5.5 Experiment Settings

In this section, we study the efficacy of our HKD approach equipped with language clusters generated using different language features.

Data: We conducted extensive experiments on a parallel corpus (53 languages→English)

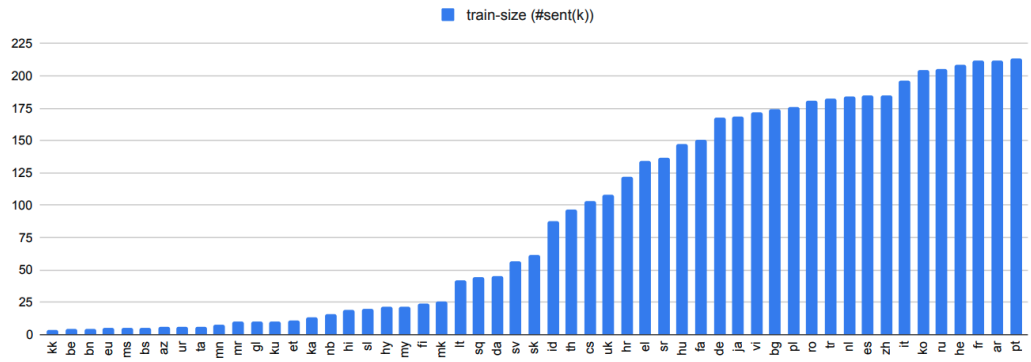

Figure 5.3: The size of the training data (based on the number of sentences) for TED-53 bilingual resources (Language→English)

from TED talks transcripts [3] created and tokenized by Qi et al. (2018). This corpus has 26% of language pairs having less than or equal to 10k sentences (extremely low-resource), and 33% of language pairs having less than 20k sentences (low-resource). All the sentences were segmented with BPE segmentation (Sennrich, Haddow, & Birch, 2016c). We used a shared vocabulary across all languages used for teachers and student models to make the knowledge distillation feasible. Detail about the size of training data and language codes based on ISO 639-1 standard[4] are listed in Table 5.2 and visualised in Figure 5.3. We concatenated all data which have the Portuguese-related languages in the source (pt→en, pt-br→en). We also concatenated all data with French-related languages in the source (fr→en, fr-ca→en). We removed any sentences in the training data which has overlap with any of the test sets. For multilingual training, as a standard practice (X. Wang, Tsvetkov, & Neubig, 2020), we up-sampled the data of low-resource language pairs to make all language pairs having roughly the same size and adjust the distribution of training data.

Clustering: We clustered all the languages based on the three different types of representations discussed in Section 5.3.2 in order to take advantage of a mixture of linguistic features while training the ultimate student. Following Oncevay et al. (2020), we adopted their multi-view language representation approach that uses Singular Vector Canonical Cor-

[3] https://github.com/neulab/word-embeddings-for-nmt
[4] http://www.loc.gov/standards/iso639-2/php/English_list.php

TED-53 Languages

Language name	Kazakh	Belarusian	Bengali	Basque	Malay	Bosnian
Code	kk	be	bn	eu	ms	bs
train-size (#sent(k))	3.3	4.5	4.6	5.1	5.2	5.6
Language name	Azerbaijani	Urdu	Tamli	Mongolian	Marathi	Galician
Code	az	ur	ta	mn	mr	gl
train-size (#sent(k))	5.9	5.9	6.2	7.6	9.8	10
Language name	Kurdish	Estonian	Georgian	Bokmal	Hindi	Slovenian
Code	ku	et	ka	nb	hi	sl
train-size (#sent(k))	10.3	10.7	13.1	15.8	18.7	19.8
Language name	Kurdish	Estonian	Georgian	Bokmal	Hindi	Slovenian
Code	ku	et	ka	nb	hi	sl
train-size (#sent(k))	10.3	10.7	13.1	15.8	18.7	19.8
Language name	Armenian	Burmese	Finnish	Macedonian	Lithuanian	Albanian
Code	hy	my	fi	mk	lt	sq
train-size (#sent(k))	21.3	21.4	24.2	25.3	41.9	44.4
Language name	Danish	Swedish	Slovak	Indonesian	Thai	Czech
Code	da	sv	sk	id	th	cs
train-size (#sent(k))	44.9	56.6	61.4	87.4	96.9	103
Language name	Ukrainian	Croatian	Greek	Serbian	Hungarian	Persian
Code	uk	hr	el	sr	hu	fa
train-size (#sent(k))	108.4	122	134.3	136.8	147.1	150.8
Language name	German	Japanese	Vietnamese	Bulgarian	Polish	Romanian
Code	de	ja	vi	bg	pl	ro
train-size (#sent(k))	167.8	168.2	171.9	174.4	176.1	180.4
Language name	Turkish	Dutch	Chinese	Spanish	Italian	Korean
Code	tr	nl	zh	es	it	ko
train-size (#sent(k))	182.3	183.7	184.8	195.9	204.4	205.4
Language name	Russian	Hebrew	French	Arabic	Portuguese	
Code	ru	he	fr	ar	pt	
train-size (#sent(k))	208.4	211.7	212	213.8	236.4	

Table 5.2: Bilingual resources of 53 Languages $\rightarrow$ English from TED dataset. Language names, language codes based on ISO 639-1 standard[5], and training size based on the number of sentences in bilingual resources are shown in this table.

relation Analysis – SVCCA (Raghu et al., 2017) to fuse the one-hot encoded KB representation obtained from syntactic features of WALS (Dryer & Haspelmath, 2013) and a dense NMT-learned view obtained from MNMT (Tan, Chen, et al., 2019). Specifically, *SVCCA-53* uses 53 languages of TED dataset to build the language representations and generates

Clustering type (1)									
cluster 1	cluster 2	cluster 3	cluster 4	cluster 5	cluster 6	cluster 7	cluster 8	cluster 9	cluster 10
Japanese Korean Mongolian Burmese	Malay Thai Vietnamese Chinese Indonesian	Marathi Tamli Bengali Georgian	Kurdish Persian Kazakh Basque Hindi Urdu	Greek Arabic Hebrew	Turkish Azerbaijani Finnish Hungarian Armenian	Ukrainian Polish Russian Macedonian Lithuanian Belarusian Slovak	Czech Estonian Albanian Croatian Bosnian Slovenian Serbian	Bulgarian Romanian Spanish Galician Italian Portuguese	French Danish Swedish Dutch German Bokmal

Table 5.3: **Clustering type (1)**: SVCCA-53 (Oncevay et al., 2020), clustering based on multi-view representation using both syntax features of WALS and language vectors learned by multilingual NMT model trained with TED-53.

Clustering type (2)									
cluster 1	cluster 2	cluster 3	cluster 4	cluster 5	cluster 6	cluster 7	cluster 8	cluster 9	cluster 10
Korean Bengali Marathi Hindi Urdu	Basque Arabic Hebrew	Armenian Persian Kurdish	Hungarian Turkish Azerbaijani Japanese Mongolian	Georgian Tamli	Kazakh Burmese	Macedonian Albanian Polish Slovak Croatian Bosnian Belarusian Estonian	Russian Ukrainian Slovenian Serbian Finnish Czech Lithuanian	Chinese Thai Indonesian Vietnamese Malay	Bulgarian Swedish Danish Bokmal German Dutch Romanian Greek Spanish Italian Galician French Portuguese

Table 5.4: **Clustering type (2)**: SVCCA-23 (Oncevay et al., 2020), clustering based on multi-view representation using both syntax features of WALS and language vectors learned by multilingual NMT model trained with WIT-23.

10 clusters, the languages within each of which usually have the same *phylogenetic* or *geographical* features. *SVCCA-23* instead uses 23 languages of WIT-23 (Cettolo, Girardi, & Federico, 2012) to compute the shared space. We also generated language clusters based on either KB-based representation using syntax features of WALS (Dryer & Haspelmath, 2013) and NMT-learned representation alone. Tables 5.3-5.6 show the generated language clusters.

Training Configuration: All models are trained with Transformer architecture (Vaswani et al., 2017a), implemented in the Fairseq framework (Ott et al., 2019). The individual models are trained with the model hidden size of 256, feed-forward hidden size of 1024,

Clustering type (3)										
cluster 1	cluster 2	cluster 3	cluster 4	cluster 5	cluster 6	cluster 7	cluster 8	cluster 9	cluster 10	cluster 11
Estonian Finnish	Hindi Burmese Armenian Georgian	Basque Azerbaijani Kazakh Mongolian Urdu Marathi Bengali Tamli Kurdish Bosnian Belarusian Malay	Galician French Italian Spanish Portuguese	Bokmal Danish Swedish	German Dutch	Chinese Japanese Korean Hungarian Turkish	Lithuanian Slovenian Croatian Serbian Czech Slovak Polish Russian Ukrainian	Persian Indonesian Arabic Hebrew Thai Vietnamese	Romanain Albanian	Macedonian Bulgarian Greek

Table 5.5: **Clustering type (3)**: clustering based on NMT-learned representation using a set of 53 factored language embeddings (Oncevay et al., 2020; Tan, Chen, et al., 2019).

Clustering type (4)								
cluster 1				cluster 2	cluster 3			
Bengali Marathi Chinese Burmese Korean Japanese Mongolian	Kazakh Azerbaijani Turkish Hindi Urdu Tamli			Thai Vietnamese Indonesian Malay	Georgian Arabic Hebrew French Czech Lithuanian German Russian Belarusian	Slovenian Serbian Ukrainian Croatian Macedonian Slovak Romanian Albanian Greek	Hungarian Dutch Swedish Danish Bokmal Galician Portuguese Spanish Italian	Estonian Finnish Armenian Polish Bulgarian Bosnian Persian Kurdish Basque

Table 5.6: **Clustering type (4)**: clustering based on KB-based representation using syntax features of WALS (Oncevay et al., 2020).

and 2 layers. All multilingual models either cluster-based or universal MNMT models with or without knowledge distillation were trained with the model hidden size of 512, feed-forward hidden size of 1024, and 6 layers. We use the Adam optimizer (Kingma & Ba, 2015) and an inverse square root schedule with warmup (maximum LR 0.0005). We apply dropout and label smoothing with a rate of 0.3 and 0.1 for bilingual and multilingual models respectively.

For the first phase of distillation, i.e., the multilingual selective KD, the distillation coefficient λ is equal to 0.6. In the second phase of distillation, i.e., the multilingual adaptive KD, we applied $\lambda_1 = 0.5$ and λ_2 is started from 0.5 and increased to 3 using the annealing

function of (Bowman et al., 2016; Saleh et al., 2020). We train our final multilingual student with mixed-precision floats on up to 8 V100 GPUs for maximum 100 epochs ($\approx$ 3 days), with at most 8192 tokens per batch and early stopping at 20 validation steps based on the BLEU score. The translation quality is also evaluated and reported based on the BLEU (Papineni, Roukos, Ward, & Zhu, 2002c) score[6]. We did not carry out intense parameter tuning to search for the best parameter settings of each model for the sake of simplicity. It is noteworthy that the purpose of our experiments is rather to demonstrate the benefit of considering different type of language clusters through a unified hierarchical KD.

5.6 Findings

The translation results of (53 languages $\rightarrow$ English) for all cluster-based approaches are summarised in Table 5.7. The language pairs are sorted based on the size of training data in an ascending order. The translation quality is evaluated and reported based on the BLEU score (Papineni et al., 2002c).

5.6.1 Studies of cluster-based MNMT models

In this section, we discuss the cluster-based MNMTs' results in different data-size situations and for different language families through the following observations.

Low-resource vs high-resource languages: All cluster-based MNMT and baseline approaches are ranked based on the number of the times they got the first or second-best score in different resource-size scenarios in Table 5.9. Based on this result and also the result represented in Table 5.7, the massive MNMT models with all languages (second column under baseline and first column under selective KD in Tables 5.7, 5.9) outperform the cluster-based MNMT models (columns (2-5) under selective KD in Tables 5.7, 5.9) in extremely low-resource scenarios (e.g., bn-en, ta-en, eu-en). This result shows that having more data either from related languages or distant languages has the most impact on train-

[6]SacreBLEU signature:BLEU+case.mixed+numrefs.1+smooth.exp+tok.none+version.1.3.1

Resource	MT task [$lang$]-en	size # sent.	Baseline		Multilingual Selective KD				
			Individ.	Multi.	All languages	Clus. type1	Clus. type2	Clus. type3	Clus. type4
Extremely low-resource	kk	3314	3.42	5.05	4.66	7.00	3.51	3.10	**8.13**
	be	4508	5.13	12.51	12.36	12.78	10.81	8.46	**15.18**
	bn	4647	5.06	12.50	**12.58**	9.13	10.11	12.16	10.13
	eu	5180	4.40	**13.12**	12.00	9.08	9.70	8.14	11.03
	ms	5219	3.78	13.88	**14.61**	12.93	12.94	7.63	12.98
	bs	5661	7.92	14.82	15.46	18.05	16.89	9.03	**19.02**
	az	5944	5.79	**10.32**	9.91	9.59	9.17	8.64	9.23
	ur	5965	8.98	12.76	**16.50**	13.35	13.17	12.02	13.39
	ta	6223	4.57	5.86	**6.19**	4.02	5.76	3.96	3.49
	mn	7604	3.54	6.11	5.82	5.75	**6.60**	5.39	6.20
	mr	9837	6.92	10.53	**10.72**	8.70	9.00	8.39	9.04
	gl	10009	13.5	22.04	22.44	25.53	26.81	**26.93**	25.90
	ku	10308	6.30	10.32	12.12	9.43	6.98	9.22	**12.93**
	et	10738	8.24	12.21	13.19	13.47	13.21	10.44	**13.94**
Low-resource	ka	13177	8.64	8.18	8.66	9.28	**10.85**	9.14	8.88
	nb	15819	26.36	28.49	29.08	33.55	**34.31**	28.79	30.87
	hi	18789	10.66	16.03	**17.93**	13.27	12.09	12.16	12.80
	sl	19824	11.45	15.12	15.39	16.48	16.54	17.75	**18.31**
Enough resource	hy	21337	11.14	14.07	15.12	13.76	12.77	10.81	**17.17**
	my	21495	4.91	10.70	**11.11**	9.65	6.35	8.48	9.54
	fi	24219	8.16	11.69	12.23	11.36	12.57	10.59	**12.76**
	mk	25326	18.32	20.63	21.09	21.48	20.06	**24.65**	23.8
	lt	41910	14.78	15.44	16.76	16.98	16.9	17.96	**18.24**
	sq	44489	22.62	24.44	25.22	24.74	23.22	**26.89**	26.42
	da	44925	31.85	30.39	30.61	35.02	**39.76**	30.58	32.04
	sv	56629	27.20	27.18	26.84	31.36	**34.52**	26.81	28.65
	sk	61454	19.36	22.04	22.58	22.49	21.18	**24.08**	23.77
	id	87401	20.51	20.89	20.69	21.11	21.13	**22.56**	21.12
	th	96954	20.46	21.34	21.72	22.94	22.94	**23.09**	22.87
	cs	103062	20.13	22.01	22.07	21.72	22.49	**23.12**	22.86
	uk	108478	21.32	22.11	23.07	23.06	22.91	23.58	**23.66**
	hr	122074	25.89	26.51	27.17	27.56	25.62	**28.66**	28.34
	el	134311	26.82	26.07	28.51	30.05	**31.35**	29.13	29.66
	sr	136891	26.94	25.43	25.88	27.48	25.75	27.69	27.12
	hu	147190	18.46	17.61	18.55	19.08	18.41	**20.16**	19.82
	fa	150813	**23.60**	21.7	21.29	21.31	22.44	23.51	22.24
	de	167864	15.23	14.83	16.69	16.88	**17.79**	15.44	16.67
	ja	168289	10.11	8.61	8.93	10.14	10.14	**10.17**	8.69
	vi	171971	18.97	19.19	20.58	**21.60**	**21.60**	21.30	20.33
	bg	174428	28.85	27.66	29.14	31.67	**32.18**	29.86	30.48
	pl	176134	17.23	18.62	19.45	19.45	18.37	19.93	**20.26**
	ro	180460	25.21	25.97	26.53	28.03	28.43	**28.90**	27.35
	tr	182387	17.72	10.2	10.01	18.66	18.19	**19.85**	18.27
	nl	183737	27.65	26.91	26.82	28.05	29.07	**29.17**	28.03
	zh	184821	20.44	22.10	22.71	22.19	22.11	**23.91**	22.84
	es	195993	30.17	29.55	30.00	29.06	**33.45**	31.46	31.82
	it	204438	26.84	25.13	27.99	30.57	**30.85**	30.36	29.45
	ko	205436	15.98	15.71	16.41	15.17	15.18	**17.45**	16.00
	ru	208413	19.76	19.83	20.86	20.85	20.80	21.25	**21.49**
	he	211761	29.35	28.03	28.27	**32.82**	32.18	31.32	30.02
	fr	212078	30.08	30.55	30.28	**32.25**	32.19	31.14	31.34
	ar	213880	25.36	23.89	24.48	**28.53**	28.03	27.46	25.85
	pt	236498	30.99	31.12	30.85	33.36	**33.84**	33.25	32.56

Table 5.7: Comparison of the clustering approaches with the baselines based on the BLEU score.

MT task $[lang]$-en	**Baseline**		**Multilingual Selective KD**				
	Individ.	Multi.	All languages	Clus. type1	Clus. type2	Clus. type3	Clus. type4
Avg.	18.50	18.64	19.24	19.84	19.87	19.35	**20.05**

Table 5.8: The average BLEU scores of the translation tasks for 53 Languages → English.

Resource-size	**size** (# sent.)	**Baseline**		**Multilingual Selective KD**				
		Individ.	Multi.	All langs	Clus. type1	Clus. type2	Clus. type3	Clus. type4
Extremely low-resource	<= 10k	0%	21.43%	**28.57%**	14.28%	7.14%	3.58%	25.00%
Low-resource	> 10k and <= 20k	0%	12.50%	12.50%	**25.00%**	**25.00%**	12.50%	12.50%
Enough resource	> 20k	1.39%	1.39%	2.78%	20.83%	25.00%	**27.78%**	20.83%

Table 5.9: The translation ranking ablation study for all approaches excluding the HKD approach based on the percentage of the times they got the 1^{st} or 2^{nd} best results. Sum of percentages in each row = 100%.

ing a better MNMT model for under-resourced languages. Furthermore, clustering type (4) is dominant among other clustering approaches for under-resourced situations. This result is also explainable based on the size of the clusters in clustering type (4). The translations of extremely low-resource languages are significantly improved when they have been clustered in the third cluster of clustering type (4) with 35 languages (shown in Table 5.6). However, for languages with enough resources, the multilingual baselines with all languages underperformed other cluster-based MNMT models.

Related vs isolated languages: A group of languages that originated from a similar ancestor is known as a *language family*; and a language that does not have any relationship with another languages is called a *language isolate*. The language families (Eberhard et al., 2019) are shown in Table 5.10.

According to the results shown in Table 5.9, clustering approaches usually have the same behaviour and less diversity for clustering languages belonged to IE/Germanic, IE/Italic, Afroasiatic, and Austronesian families. In comparison, there is more diversity, and less consensus for clustering languages belonged to IE/Balto-Slavic, IE/Indo-Iranian, Turkic, Uralic, and Sino-Tibetan families. Moreover, the isolated languages (shown in the last 11 columns of Table 5.10) generally have the same behaviour and less variance in BLEU scores in dif-

IE/Balto-Slavic	IE/Italic	IE/Indo-Iranian	IE/Germanic	Turkic	Uralic	Afroasiatic	Sino-Tibetan	Austronesian	Koreanic	Japonic	Austroasiatic	IE/Hellenic	Kra-Dai	IE/Albanian	IE/Armenian	Kartvelian	Mongolic	Dravidian	Isolate (Basque)
be, bs, sl, mk, lt, sk cs, uk, hr, sr, bg, pl, ru	bn, ur, ku hi, fa, mr	gl, pt, ro fr, es, it	nb, da sv, de, nl	kk, az, tr	et, fi, hu	he, ar	my, zh	ms, id	ko	ja	vi	el	th	sq	hy	ka	mn	ta	eu

Table 5.10: Language families (Eberhard et al., 2019). IE refers to Indo European.

ferent clustering approaches. This observation shows that cluster-based MNMT models (regardless of the clustering type) do not significantly improve the translation of isolated languages unless the isolated languages have extremely low-resources and have been clustered in a huge cluster (e.g., eu-en, hy-en). The results of cluster-based MNMT are presented based on the language families in Tables 5.16, 5.17, 5.18, and 5.19.

Random clustering vs Actual clustering: We conducted an ablation study by using clusters with randomly chosen languages for two translation tasks (el→en and gl→en). We kept the number of languages per cluster the same as the actual clustering to make a fair comparison. According to the result represented in Table 5.11, for both translation tasks, random clusters underperform the actual clusters in all clustering types. However, notably, the average BLEU score's difference between the random and actual clusters for gl→en is considerably higher than el→en (gl→en, Δ = -8.14 **vs** el→en, Δ = -1.9). This observation is inline with the previous observation that *Greek* (el) is an isolated language categorised in IE/Hellenic family and clustering approaches have less impact on this language due to its lower similarity to most languages. In comparison, *Galician* (gl) is highly similar to the languages in IE/Italic family and clustering improves the translation of gl→en remarkably.

5.6.2 Studies of HKD

According to the results in Tables of 5.12 and 5.13, the HKD approach outperforms massive and cluster-based MNMT models in average by 1.11 BLEU score. We discuss the HKD's results in the following observations:

Ranking based on data size: We ranked all approaches, including HKD, based on the

Model	Contributed Languages	BLEU	Contributed Languages	BLEU
Individual	gl	13.50	el	26.82
Multi. (uniform)	All langs.	22.04	All langs.	26.07
Multi. (SKD)	All langs.	22.44	All langs.	28.51
Clus. type1	gl, bg, ro, es, it, pt	25.53	el, ar, he	30.05
Clus. type2	gl, bg, sv, da, nb, de, nl, ro, el, es, it, fr, pt	26.81	el, bg, sv, da, nb, de, nl, es, it, gl, fr, pt, ro	31.35
Clus. type3	gl, fr, it, es, pt	26.93	el, mk, bg	29.13
Clus. type4	gl, fa, ku, eu, hu, et, fi, hy, ka, ar, he, fr, cs, lt, de, nl, it, sv, da, nb, ru, be, pl, bg, sl, pt sr, uk, hr, mk, sk, ro, sq, el, es	25.90	el, fa, ku, eu, hu, et, fi, hy, ka, ar, he, fr, cs, lt, de, nl, it, sv, da, nb, ru, be, pl, bg, sl, pt, sr, uk, hr, mk, sk, ro, sq, es, gl	29.66
Avg.	-	**26.29**	-	**30.04**
Clus. Rand.1	gl, nb, uk, hr, se, ja	16.53	el, id, be	28.01
Clus. Rand.2	gl, ta, mk, be, id, sq, pt, fr, ur, az, ku, bs, fa	20.27	el, cs, lt, id, sk, th, it, hy, ms, hu, mk, my, bn	27.78
Clus. Rand.3	gl, az, ja, nb, kk	13.61	el, sq, th	27.97
Clus. Rand.4	gl, zh, pt, fa, ar, kk, sr, bg, nl, cs, th, ko, vi, hu, mk, fi, ru, mn, de, sl, el, ka, pl, et, ta, fr, ur, ro, sv, mr, be, bs, uk, sq, az	22.20	el, zh, pt, fa, ar, kk, sr, bg, nl, cs,th, ko, vi, hu, mk, fi, ru, mn, de, sl, gl, ka, pl, et, ta, fr, ur, ro, sv, mr, be, bs, uk, sq, az	28.82
Avg.	-	**18.15**$_{\Delta-8.14}$	-	**28.14**$_{\Delta-1.9}$

Table 5.11: Ablation study on using random clusters. Comparison of the (gl→en) and (el→en) translation tasks between individual, massive multilingual, and clustering-based multilingual (for actual and random clusters) baselines.

number of times they got the first or second-best score (shown in Table 5.14). According to this result, the HKD approach in three different situations, i.e., extremely low-resource, low-resource, and enough resource, has the best rank among other approaches. This observation proves that the HKD approach is robust in different *data-size* situations by leveraging the best of both multilingual NMT and language-relatedness guidance in a systematic HKD setting.

Clustering consistency impact on HKD: Based on the result in Table 5.12, the HKD

Resource	MT task [$lang$]-en	size # sent.	Baseline		Multilingual Selective KD					Multilingual HKD
			Individ.	Multi.	All languages	Clus. type1	Clus. type2	Clus. type3	Clus. type4	
Extremely low-resource	kk	3314	3.42	5.05	4.66	7.00	3.51	3.10	**8.13**	6.61
	be	4508	5.13	12.51	12.36	12.78	10.81	8.46	**15.18**	14.88
	bn	4647	5.06	12.50	12.58	9.13	10.11	12.16	10.13	**12.81**
	eu	5180	4.4	**13.12**	12.00	9.08	9.70	8.14	11.03	12.90
	ms	5219	3.78	13.88	**14.61**	12.93	12.94	7.63	12.98	14.11
	bs	5661	7.92	14.82	15.46	18.05	16.89	9.03	**19.02**	17.51
	az	5944	5.79	10.32	9.91	9.59	9.17	8.64	9.23	**10.40**
	ur	5965	8.98	12.76	**16.50**	13.35	13.17	12.02	13.39	15.95
	ta	6223	4.57	5.86	**6.19**	4.02	5.76	3.96	3.49	5.63
	mn	7604	3.54	6.11	5.82	5.75	6.60	5.39	6.20	**6.81**
	mr	9837	6.92	10.53	10.72	8.70	9.00	8.39	9.04	**10.98**
	gl	10009	13.5	22.04	22.44	25.53	26.81	26.93	25.90	**27.11**
	ku	10308	6.3	10.32	12.12	9.43	6.98	9.22	12.93	**13.03**
	et	10738	8.24	12.21	13.19	13.47	13.21	10.44	13.94	**14.10**
Low-resource	ka	13177	8.64	8.18	8.66	9.28	10.85	9.14	8.88	**11.15**
	nb	15819	26.36	28.49	29.08	33.55	**34.31**	28.79	30.87	33.89
	hi	18789	10.66	16.03	**17.93**	13.27	12.09	12.16	12.80	16.11
	sl	19824	11.45	15.12	15.39	16.48	16.54	17.75	18.31	**18.43**
Enough resource	hy	21337	11.14	14.07	15.12	13.76	12.77	10.81	**17.17**	16.72
	my	21495	4.91	10.70	**11.11**	9.65	6.35	8.48	9.54	8.81
	fi	24219	8.16	11.69	12.23	11.36	12.57	10.59	12.76	**12.90**
	mk	25326	18.32	20.63	21.09	21.48	20.06	24.65	23.8	**25.05**
	lt	41910	14.78	15.44	16.76	16.98	16.9	17.96	**18.24**	18.11
	sq	44489	22.62	24.44	25.22	24.74	23.22	26.89	26.42	**26.93**
	da	44925	31.85	30.39	30.61	35.02	**39.76**	30.58	32.04	36.00
	sv	56629	27.2	27.18	26.84	31.36	**34.52**	26.81	28.65	33.14
	sk	61454	19.36	22.04	22.58	22.49	21.18	24.08	23.77	**24.33**
	id	87401	20.51	20.89	20.69	21.11	21.13	22.56	21.12	**22.76**
	th	96954	20.46	21.34	21.72	22.94	22.94	23.09	22.87	**23.30**
	cs	103062	20.13	22.01	22.07	21.72	22.49	23.12	22.86	**23.62**
	uk	108478	21.32	22.11	23.07	23.06	22.91	23.58	23.66	**24.09**
	hr	122074	25.89	26.51	27.17	27.56	25.62	28.66	28.34	**28.91**
	el	134311	26.82	26.07	28.51	30.05	**31.35**	29.13	29.66	30.10
	sr	136891	26.94	25.43	25.88	27.48	25.75	27.69	27.12	**27.97**
	hu	147190	18.46	17.61	18.55	19.08	18.41	**20.16**	19.82	20.10
	fa	150813	**23.60**	21.7	21.29	21.31	22.44	23.51	22.24	23.19
	de	167864	15.23	14.83	16.69	16.88	17.79	15.44	16.67	**18.04**
	ja	168289	10.11	8.61	8.93	10.14	10.14	10.17	8.69	**10.30**
	vi	171971	18.97	19.19	20.58	21.60	21.60	21.30	20.33	**21.82**
	bg	174428	28.85	27.66	29.14	31.67	32.18	29.86	30.48	**32.33**
	pl	176134	17.23	18.62	19.45	19.45	18.37	19.93	20.26	**20.71**
	ro	180460	25.21	25.97	26.53	28.03	28.43	28.90	27.35	**29.00**
	tr	182387	17.72	10.2	10.01	18.66	18.19	**19.85**	18.27	16.91
	nl	183737	27.65	26.91	26.82	28.05	29.07	29.17	28.03	**29.58**
	zh	184821	20.44	22.10	22.71	22.19	22.11	**23.91**	22.84	23.85
	es	195993	30.17	29.55	30.00	29.06	**33.45**	31.46	31.82	32.76
	it	204438	26.84	25.13	27.99	30.57	30.85	30.36	29.45	**30.93**
	ko	205436	15.98	15.71	16.41	15.17	15.18	17.45	16.00	**17.70**
	ru	208413	19.76	19.83	20.86	20.85	20.80	21.25	21.49	**21.77**
	he	211761	29.35	28.03	28.27	32.82	32.18	31.32	30.02	32.05
	fr	212078	30.08	30.55	30.28	32.25	32.19	31.14	31.34	**32.65**
	ar	213880	25.36	23.89	24.48	28.53	28.03	27.46	25.85	**28.94**
	pt	236498	30.99	31.12	30.85	33.36	33.84	33.25	32.56	**33.90**

Table 5.12: BLEU scores of all the translation tasks for 53 Languages → English.

MT task $[lang]$-en	Baseline		Multilingual Selective KD					Multilingual HKD
	Individ.	Multi.	All languages	Clus. type1	Clus. type2	Clus. type3	Clus. type4	
Avg.	18.50	18.64	19.24	19.84	19.87	19.35	20.05	**21.16**

Table 5.13: The average BLEU scores of all the translation tasks for (53 Languages → English).

Resource-size	size (# sent.)	Baseline		Multilingual Selective KD					HKD
		Individ.	Multi.	All langs	Clus. type1	Clus. type2	Clus. type3	Clus. type4	
Extremely low-resource	<= 10k	0%	13.79%	17.24%	6.90%	3.45%	3.45%	17.24%	**37.93%**
Low-resource	> 10k and < 20k	0%	0%	12.50%	0%	25.00%	0%	12.50%	**50.00%**
Enough resource	>20k	1.43%	1.43%	1.43%	5.71%	12.86%	24.28%	8.57%	**44.29%**

Table 5.14: The ranking of all approaches including the HKD approach based on the percentage of the times they got the 1^{st} or 2^{nd} best results.

approach underperforms other multilingual approaches when the clusters are inconsistent, causing a high variance in teacher-assistants' results. For example for kk→en, bs→en, the variance of the BLEU scores of the teacher-assistant models is 6.92 and 20.81 respectively and HKD underachieved a good result. This observation shows that, although in the second phase of the HKD approach, the cluster-based teacher-assistants adaptively contribute to training the ultimate students, still a weak teacher-assistant deteriorates the collaborative teaching process. One possible solution is excluding the worst teacher-assistant in such heterogeneous situations.

5.6.3 Comparison with other approaches

Following the previous discussions, to highlight the pros and cons of the related baselines (with and without KD), we draw a comparison shown in Table 5.15. Accordingly, our HKD approach is comparable with other approaches based on the following properties:

Multilingual translation: Our approach works in a multilingual setting by sharing resources between high-resource and low-resource languages. This property not only improves the regularisation of the model by avoiding over-fitting to the limited data of the low-resource languages but also decreases the deployment footprint by leveraging the whole training in a single model instead of having individual models per language (Dabre et al.,

	Individual NMT	Uniform MNMT	Selective KD MNMT	Adaptive KD NMT	HKD MNMT
Multilingual	✗	✔	✔	✗	✔
Maximum transfer	●	✔	✗	✗	✔
KD from multiple languages	●	●	✗	✔	✔
Reduced risk of negative transfer	●	✗	✔	✔	✔

Table 5.15: Comparing different properties of HKD with: transformer-based individual and multilingual NMT (Vaswani et al., 2017a), multilingual selective KD (Tan, Ren, et al., 2019), and adaptive KD (Saleh et al., 2020).

2020).

Optimal transfer: In the HKD approach, we have an optimal transfer by transferring knowledge from *all* possible languages *related* to a student in the hierarchical structure which leads to the best average BLEU score (21.16) comparing to the other baselines (shown in Table 5.12). In the universal multilingual NMT without KD, the language transfer stream is maximized when all languages shared their knowledge in a single model during training; however, it is not an optimal transfer due to the lack of any condition on the the relatedness of languages contributing in the multilingual training. The related experiments are shown in the second column under the baseline experiments in Table 5.12. The average BLEU score of this approach is 18.46. In multilingual selective KD (Tan, Ren, et al., 2019), knowledge is distilled from one selected teacher with the same language when training the student multilingually. So, although there is a condition on language relatedness, knowledge transfer is not maximized as the similar languages from the same language family are ignored in the distillation process. The related results are shown in the first column under multilingual selective KD of Table 5.12. Accordingly, this approach got the average BLEU score of 19.24 in our experiments. Adaptive KD (Saleh et al., 2020) is a bilingual approach and also uses a random set of teachers which does not essentially have all the related languages to the student and does not lead to optimal transfer. We did not perform any experiment on adaptive KD (Saleh et al., 2020) since this is a bilingual approach.

Adaptive KD vs Selective KD vs HKD: All KD-based approaches in our comparison re-

duce *negative transfer* in different ways. In multilingual selective KD (Tan, Ren, et al., 2019), the risk of negative transfer is reduced by distilling knowledge from the selected teacher per language in a multilingual setting. In bilingual adaptive KD (Saleh et al., 2020), the contribution weights of different teachers vary based on their effectiveness to improve the student which prevents the negative transfer in bilingual setting. In HKD, the hierarchical grouping based on the language similarity provides a systematic guide to prevent negative transfer as much as possible. This property leads HKD to get the *best* results for 32 language pairs out of total 53 language pairs in our multilingual experiments.

5.7 Conclusion

We presented a Hierarchical Knowledge Distillation (HKD) approach to mitigate the negative transfer effect in MNMT when having a diverse set of languages in training. We put together all languages which behave similarly in the first phase of distillation process and generated the expert teacher-assistants for each group of languages. As we clustered languages based on four different language representations capturing different linguistic features, we then adaptively distill knowledge from all related teacher-assistant models to the ultimate student in each mini-batch of training per language. Experimental results on 53 languages to English show our approach's effectiveness to reduce negative transfer in MNMT. As the future direction, it is interesting to study an end-to-end HKD approach by adding a backward HKD pass compared to the forward HKD pass described in this chapter.

family	lang	Clustering Type 1										Clustering Type 2										Clustering Type 3										Clustering Type 4		
		c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9	c_{10}	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9	c_{10}	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9	c_{10}	c_{11}	c_1	c_2	c_3
		ja, ko, mn, my	ms, th, vi, zh, id	mr, ta, bn, ka	ku, fa, kk, eu, hi, ur	el, ar, he	tr, az, fi, hu, hy	uk, pl, ru, mk, lt, be, sk	cs, et, sq, hr, bs, sl, sr	bg, ro, es, gl, it, pt	fr, da, sv, nl, de, nb	ko, bn, mr, hi, ur	eu, ar, he	hy, fa, ku	hu, tr, az, ja, mn	ka, ta	kk, my	mk, sq, pl, sk, hr, bs, be, et	ru, uk, sl, fi, cs, lt	zh, th, id, vi, ms	bg, sv, da, nb, de, nl, el, es, it, gl, fr, pt, ro	et, fi	hi, my, hy, ka	eu, az, kk, mn, ur, mr, bn, ta, ku, bs, be, ms	gl, fr, it, es, pt	nb, da, sv	de, nl	zh, ja, ko, hu, tr	lt, sl, hr, sr, cs, sk, pl, ru, uk	fa, id, ar, he, th, vi	ro, sq	mk, bg, el	fa, ku, eu, hu, et, fi, hy, ka, ar, he, fr, cs, lt, de, nl, sv, da, nb, ru, be, pl, bg, sl, sr, uk, hr, mk, sk, ro, sq, el, es, gl, pt, it	vi, id, ms, th	kk, az, tr, lt, ur, bn, mr, zh, my, ko, ja, mn, ta
IE/Balto-Slavic	be	-	-	-	-	-	-	12.7	-	-	-	-	-	-	-	-	-	10.8	-	-	-	-	-	8.4	-	-	-	-	-	-	-	**15.1**	-	-
	bs	-	-	-	-	-	-	-	18.0	-	-	-	-	-	-	-	-	16.8	-	-	-	-	-	9.0	-	-	-	-	-	-	-	**19.0**	-	-
	sl	-	-	-	-	-	-	-	16.4	-	-	-	-	-	-	-	-	-	16.5	-	-	-	-	-	-	-	-	-	17.7	-	-	-	**18.3**	-	-
	mk	-	-	-	-	-	-	21.4	-	-	-	-	-	-	-	-	-	20.0	-	-	-	-	-	-	-	-	-	-	-	-	**24.6**	23.8	-	**18.2**
	lt	-	-	-	-	-	-	16.9	-	-	-	-	-	-	-	-	-	-	16.9	-	-	-	-	-	-	-	-	-	17.9	-	-	-	-	-
	sk	-	-	-	-	-	-	22.4	-	-	-	-	-	-	-	-	-	-	21.1	-	-	-	-	-	-	-	-	-	**24.0**	-	-	23.7	-	-
	cs	-	-	-	-	-	-	-	21.7	-	-	-	-	-	-	-	-	-	22.4	-	-	-	-	-	-	-	-	-	**23.1**	-	-	22.8	-	-
	uk	-	-	-	-	-	-	23.0	-	-	-	-	-	-	-	-	-	-	22.9	-	-	-	-	-	-	-	-	-	23.5	-	-	**23.6**	-	-
	hr	-	-	-	-	-	-	-	27.5	-	-	-	-	-	-	-	-	25.62	-	-	-	-	-	-	-	-	-	-	28.6	-	-	28.3	-	-
	sr	-	-	-	-	-	-	-	27.4	-	-	-	-	-	-	-	-	25.7	-	-	-	-	-	-	-	-	-	-	**27.6**	-	-	27.1	-	-
	bg	-	-	-	-	-	-	-	-	31.6	-	-	-	-	-	-	-	-	-	-	**32.1**	-	-	-	-	-	-	-	-	-	29.8	30.4	-	-
	pl	-	-	-	-	-	-	19.4	-	-	-	-	-	-	-	-	-	18.3	-	-	-	-	-	-	-	-	-	19.9	-	-	-	**20.2**	-	-
	ru	-	-	-	-	-	-	20.8	-	-	-	-	-	-	-	-	-	-	20.8	-	-	-	-	-	-	-	-	-	21.2	-	-	**21.4**	-	-
IE/Indo-Iranian	bn	-	-	9.1	-	-	-	-	-	-	-	10.1	-	-	-	-	-	-	-	-	-	-	-	**12.1**	-	-	-	-	-	-	-	-	-	10.1
	ur	-	-	-	13.3	-	-	-	-	-	-	13.1	-	-	-	-	-	-	-	-	-	-	-	12.0	-	-	-	-	-	-	-	-	-	**13.3**
	ku	-	-	-	9.4	-	-	-	-	-	-	6.9	-	-	-	-	-	-	-	-	-	-	-	9.2	-	-	-	-	-	-	-	-	-	**12.9**
	hi	-	-	-	**13.2**	-	-	-	-	-	-	12.0	-	-	-	-	-	-	-	-	-	-	12.1	-	-	-	-	-	-	-	-	12.8	-	-
	fa	-	-	-	21.3	-	-	-	-	-	-	-	-	22.4	-	-	-	-	-	-	-	-	-	-	-	-	-	-	**23.5**	-	-	22.2	-	-
	mr	-	-	8.7	-	-	-	-	-	-	-	9.0	-	-	-	-	-	-	-	-	-	-	-	8.3	-	-	-	-	-	-	-	-	-	**9.0**
IE/Italic	gl	-	-	-	-	-	-	-	-	25.5	-	-	-	-	-	-	-	-	-	-	26.8	-	-	-	**26.9**	-	-	-	-	-	-	25.9	-	-
	pt	-	-	-	-	-	-	-	-	33.3	-	-	-	-	-	-	-	-	-	-	33.8	-	-	-	33.2	-	-	-	-	-	-	32.5	-	-
	ro	-	-	-	-	-	-	-	-	28.0	-	-	-	-	-	-	-	-	-	-	28.4	-	-	-	-	-	-	-	-	**28.9**	-	27.3	-	-
	fr	-	-	-	-	-	-	-	-	-	**32.2**	-	-	-	-	-	-	-	-	-	32.1	-	-	-	31.1	-	-	-	-	-	-	31.3	-	-
	es	-	-	-	-	-	-	-	-	29.0	-	-	-	-	-	-	-	-	-	-	**33.4**	-	-	-	31.4	-	-	-	-	-	-	31.8	-	-
	it	-	-	-	-	-	-	-	-	30.5	-	-	-	-	-	-	-	-	-	-	**30.8**	-	-	-	30.3	-	-	-	-	-	-	29.4	-	-

Table 5.16: The results of clustering-based multilingual NMT with knowledge distillation for languages belong to IE/Balto-Slavic, IE/Indo-Iranian, and IE/Italic.

family	lang	Clustering Type 1										Clustering Type 2										Clustering Type 3										Clustering Type 4		
		c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9	c_{10}	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9	c_{10}	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9	c_{10}	c_{11}	c_1	c_2	c_3
		ja, ko, mn, my	ms, th, vi, zh, id	mr, ta, bn, ka	ku, fa, kk, eu, hi, ur	el, ar, he	tr, az, fi, hu, hy	uk, pl, ru, mk, lt, be, sk	cs, et, sq, hr, bs, sl, sr	bg, ro, es, gl, it, pt	fr, da, sv, nl, de, nb	ko, bn, mr, hi, ur	eu, ar, he	hy, fa, ku	hu, tr, az, ja, mn	ka, ta	kk, my	mk, sq, pl, sk, hr, bs, be, et	ru, uk, sl, fi, cs, lt	zh, th, id, vi, ms	bg, sv, da, nb, de, nl, el, es, it, gl, fr, pt, ro	et, fi	hi, my, hy, ka	eu, az, kk, mn, ur, mr, bn, ta, ku, bs, be, ms	gl, fr, it, es, pt	nb, da, sv	de, nl	zh, ja, ko, hu, tr	lt, sl, hr, sr, cs, sk, pl, ru, uk	fa, id, ar, he, th, vi	ro, sq	mk, bg, el	fa, ku, eu, hu, et, fi, hy, ka, ar, he, fr, cs, lt, de, nl, sv, da, nb, ru, be, pl, bg, sl, sr, uk, hr, mk, sk, ro, sq, el, es, gl, pt, it	vi, id, ms, th	kk, az, tr, lt, ur, bn, mr, zh, my, ko, ja, mn, ta
IE/Germanic	nb	-	-	-	-	-	-	-	-	-	33.5	-	-	-	-	-	-	-	-	-	**34.3**	-	-	-	-	28.7	-	-	-	-	-	30.8	-	-
	da	-	-	-	-	-	-	-	-	-	35.0	-	-	-	-	-	-	-	-	-	**39.7**	-	-	-	-	30.5	-	-	-	-	-	32.0	-	-
	sv	-	-	-	-	-	-	-	-	-	31.3	-	-	-	-	-	-	-	-	-	**34.5**	-	-	-	-	26.8	-	-	-	-	-	28.6	-	-
	de	-	-	-	-	-	-	-	-	-	16.8	-	-	-	-	-	-	-	-	-	**17.7**	-	-	-	-	-	15.4	-	-	-	-	16.6	-	-
	nl	-	-	-	-	-	-	-	-	-	28.0	-	-	-	-	-	-	-	-	-	29.0	-	-	-	-	-	**29.1**	-	-	-	-	28.0	-	-
Turkic	kk	-	-	-	7.0	-	-	-	-	-	-	-	-	-	-	3.5	-	-	-	-	-	-	-	3.1	-	-	-	-	-	-	-	-	-	**8.1**
	az	-	-	-	-	-	**9.5**	-	-	-	-	-	-	-	9.1	-	-	-	-	-	-	-	-	8.6	-	-	-	-	-	-	-	-	-	9.2
	tr	-	-	-	-	-	18.6	-	-	-	-	-	-	-	18.1	-	-	-	-	-	-	-	-	-	-	-	**19.8**	-	-	-	-	-	-	18.2
Uralic	et	-	-	-	-	-	-	-	13.4	-	-	-	-	-	-	-	-	13.2	-	-	-	10.4	-	-	-	-	-	-	-	-	-	13.9	-	-
	fi	-	-	-	-	-	11.3	-	-	-	-	-	-	-	-	-	-	-	12.5	-	-	10.5	-	-	-	-	-	-	-	-	-	**12.7**	-	-
	hu	-	-	-	-	-	19.0	-	-	-	-	-	-	-	18.4	-	-	-	-	-	-	-	-	-	-	-	**20.1**	-	-	-	-	19.8	-	-
Afroasiatic	he	-	-	-	-	**32.8**	-	-	-	-	-	-	-	-	-	-	-	32.1	-	-	-	-	31.3	-	-	-	-	-	-	-	-	30.0	-	-
	ar	-	-	-	-	**28.5**	-	-	-	-	-	-	28.0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	27.4	-	-	25.8	-	-
Sino-Tibetan	my	9.6	-	-	-	-	-	-	-	-	-	-	-	-	-	**6.3**	-	-	-	-	-	-	8.4	-	-	-	-	-	-	-	-	9.5	-	-
	zh	-	22.1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	22.1	-	-	-	-	-	-	23.9	-	-	-	-	-	-	22.8
Austronesian	ms	-	12.9	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	12.9	-	-	7.6	-	-	-	-	-	-	-	-	-	**12.9**	-
	id	-	21.1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	21.1	-	-	-	-	-	-	-	-	22.5	-	-	-	21.1	-

Table 5.17: The results of clustering-based multilingual NMT with knowledge distillation for languages belong to IE/Germanic, Turkic, Uralic, Afroasiatic, Sino-Tibetan, and Austronesian.

family	lang	Clustering Type 1										Clustering Type 2										Clustering Type 3										Clustering Type 4		
		c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9	c_{10}	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9	c_{10}	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9	c_{10}	c_{11}	c_1	c_2	c_3
Koreanic	ko	15.1	-	-	-	-	-	-	-	-	-	15.1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	**17.4**	-	-	-	-	-	16.0
Japonic	ja	10.1	-	-	-	-	-	-	-	-	-	-	-	-	10.1	-	-	-	-	-	-	-	-	-	-	-	-	10.1	-	-	-	-	-	8.6
Austroasiatic	vi	-	21.6	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	21.6	-	-	-	-	-	-	-	-	**21.3**	-	-	-	20.3	-
IE/Hellenic	el	-	-	-	-	30.0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	**31.3**	-	-	-	-	-	-	-	-	-	29.1	29.6	-	-
Kra-Dai	th	-	22.9	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	**22.9**	-	-	-	-	-	-	-	-	23.09	-	-	-	22.8	-
IE/Albanian	sq	-	-	-	-	-	-	24.7	-	-	-	-	-	-	-	-	-	23.2	-	-	-	-	-	-	-	-	-	-	-	-	**26.8**	26.4	-	-

ja, ko, mn, my	
ms, th, vi, zh, id	
mr, ta, bn, ka	
ku, fa, kk, eu, hi, ur	
el, ar, he	
tr, az, fi, hu, hy	
uk, pl, ru, mk, lt, be, sk	
cs, et, sq, hr, bs, sl, sr	
bg, ro, es, gl, it, pt	
fr, da, sv, nl, de, nb	
ko, bn, mr, hi, ur	
eu, ar, he	
hy, fa, ku	
hu, tr, az, ja, mn	
ka, ta	
kk, my	
mk, sq, pl, sk, hr, bs, be, et	
ru, uk, sl, fi, cs, lt	
zh, th, id, vi, ms	
bg, sv, da, nb, de, nl, el, es, it, gl, fr, pt, ro	
et, fi	
hi, my, hy, ka	
eu, az, kk, mn, ur, mr, bn, ta, ku, bs, be, ms	
gl, fr, it, es, pt	
nb, da, sv	
de, nl	
zh, ja, ko, hu, tr	
lt, sl, hr, sr, cs, sk, pl, ru, uk	
fa, id, ar, he, th, vi	
ro, sq	
mk, bg, el	
fa, ku, eu, hu, et, fi, hy, ka, ar, he, fr, cs, lt, de, nl, sv, da, nb, ru, be, pl, bg, sl, sr, uk, hr, mk, sk, ro, sq, el, es, gl, pt, it	
vi, id, ms, th	
kk, az, tr, lt, ur, bn, mr, zh, my, ko, ja, mn, ta	

Table 5.18: The results of clustering-based multilingual NMT with knowledge distillation for languages belong to Koreanic, Japonic, Austroasiatic, IE/Hellenic, Kra-Dai, and IE/Albanian families.

family	lang	Clustering Type 1										Clustering Type 2										Clustering Type 3										Clustering Type 4		
		c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9	c_{10}	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9	c_{10}	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9	c_{10}	c_{11}	c_1	c_2	c_3
IE/Armenian	hy	-	-	-	-	-	13.7	-	-	-	-	-	-	12.7	-	-	-	-	-	-	-	-	10.8	-	-	-	-	-	-	-	-	-	**17.1**	-	-
Kartvelian	ka	-	-	9.2	-	-	-	-	-	-	-	-	-	-	-	**10.8**	-	-	-	-	-	-	9.1	-	-	-	-	-	-	-	-	-	8.8	-	-
Mongolic	mn	5.7	-	-	-	-	-	-	-	-	-	-	-	-	6.6	-	-	-	-	-	-	-	-	5.3	-	-	-	-	-	-	-	-	-	6.2
Dravidian	ta	-	-	4.0	-	-	-	-	-	-	-	-	-	-	-	**5.7**	-	-	-	-	-	-	-	3.9	-	-	-	-	-	-	-	-	-	3.4
Isolate	eu	-	-	-	9.0	-	-	-	-	-	-	-	9.7	-	-	-	-	-	-	-	-	-	-	8.1	-	-	-	-	-	-	-	-	**11.0**	-

ja, ko, mn, my	ko, bn, mr, hi, ur
ms, th, vi, zh, id	eu, ar, he
mr, ta, bn, ka	hy, fa, ku
ku, fa, kk, eu, hi, ur	hu, tr, az, ja, mn
el, ar, he	ka, ta
tr, az, fi, hu, hy	kk, my
uk, pl, ru, mk, lt, be, sk	mk, sq, pl, sk, hr, bs, be, et
cs, et, sq, hr, bs, sl, sr	ru, uk, sl, fi, cs, lt
bg, ro, es, gl, it, pt	zh, th, id, vi, ms
fr, da, sv, nl, de, nb	bg, sv, da, nb, de, nl, el, es, it, gl, fr, pt, ro
	et, fi
	hi, my, hy, ka
	eu, az, kk, mn, ur, mr, bn, ta, ku, bs, be, ms
	gl, fr, it, es, pt
	nb, da, sv
	de, nl
	zh, ja, ko, hu, tr
	lt, sl, hr, sr, cs, sk, pl, ru, uk
	fa, id, ar, he, th, vi
	ro, sq
	mk, bg, el
	fa, ku, eu, hu, et, fi, hy, ka, ar, he, fr, cs, lt, de, nl, sv, da, nb, ru, be, pl, bg, sl, sr, uk, hr, mk, sk, ro, sq, el, es, gl, pt, it
	vi, id, ms, th
	kk, az, tr, lt, ur, bn, mr, zh, my, ko, ja, mn, ta

Table 5.19: The results of clustering-based multilingual NMT with knowledge distillation for languages belong to IE/Armenian, Kartvelian, Mongolic, Dravidian, and Isolate families.

6 | Concluding Remarks

The primary contribution of this thesis is improving the resource-constrained Neural Machine Translation (NMT) and Natural Language Generation (NLG) by transitioning knowledge from high-resource (NMT) models to low-resource NMT or NLG models. In this thesis, we not only used different knowledge transition schemes, i.e., transfer learning and knowledge distillation, to address the research gaps in low-resource text generation scenarios, but also argued the deficiency of conventional transfer learning and knowledge distillation approaches and introduced novel techniques to address those shortcomings. This chapter concludes this thesis via reviewing the contributions and discussing the potential future directions to the findings of this research.

6.1 Summary of the Thesis

This thesis introduced effective knowledge transition approaches for learning NMT and NLG tasks when there are constraints on resources. We have explored three knowledge transition schemes in the low-resource scenario: i) knowledge transition from a high-resource task (NMT) to a different low-resource task (NLG), ii) knowledge transition from an ensemble of high-resource models to a low-resource model from the same task (MT) in a bilingual setting disregarding the relatedness/similarity of contributed languages, and iii) knowledge transition from groups of NMT models to a single multilingual model given the language

similarity in a hierarchy structure. In particular, this thesis has made three contributions to the literature of low-resource neural text generation tasks. The first contribution mainly focused on low-resource NLG, while the second and third contributions focused on low-resource NMT in bilingual and multilingual settings.

In Chapter 3, we proposed a novel approach by transitioning from document-level NMT to NLG which improves both coherency and adequacy of the generated text in low-resource NLG scenario. Inspiring by the similarities between document-level machine translation and data-to-text generation tasks in generating the same target text (e.g., generating English stories from either German stories or structured meta-data), we proposed to encode the structured meta-data as a document-level text sequence. We formulated a compact way to encode the data available in the original database and enrich it with extra facts that can be inferred with minimal knowledge of the task. We then fine-tuned a high-resource pre-trained document-level NMT model with small amounts of document-level NLG meta-data, transitioning from NMT to NLG. To the best of our knowledge, this is the first research which argues that separate data selection and data ordering steps are not necessary for data-to-text generation if the NLG model is transferred from a document-level translation model and is given all meta-data as a document sequence. This approach outperformed the previous state of the art on the Rotowire NLG dataset and obtained the best results on each of the six tasks of Document-level Generation and Translation (DGT) shared task in WNGT 2019. This work was published in EMNLP 2019.

In Chapter 4, we looked at the problem of data scarcity for training a high-quality NMT model. We first discussed the weaknesses of the existed transfer learning and knowledge distillation approaches proposed to improve the low-resource NMT. Then, we introduced our novel adaptive knowledge distillation algorithm which addresses the inefficiency of the original transfer learning by using an ensemble of high-resource pre-trained models in an adaptive collaborative learning manner to train a single low-resource NMT model. The main contribution in this chapter was derived from the intuition that models transferred from different high-resource language pairs may have complementary syntactic and/or se-

mantic strengths on the target low-resource language. Furthermore, it is not generally clear which high-resource language pair offers the best transfer learning for the target MT setting in every mini-batch of data. Thus, high-resource language pairs should not have the same contribution in transfer learning while training the low-resource NMT model. In our Adaptive Knowledge Distillation (AKD) approach, the label smoothing coming from different teachers is combined and regulated based on the loss incurred by the teacher models during the distillation process, and thus the contribution of each teacher is changed based on its effectiveness in improving the student. Experiments on transferring from a collection of six language pairs from IWSLT to five low-resource language pairs from TED Talks demonstrated the effectiveness of our approach, achieving up to +0.9 BLEU score improvement compared to solid baselines. This work was published in COLING 2020.

In Chapter 5, we shifted toward NMT in multilingual settings inspired by our previous findings mentioned above. We presented a Hierarchical Knowledge Distillation (HKD) approach which capitalises on language groups generated based on typological features and phylogeny of languages to mitigate the negative transfer effect in MNMT when having a diverse set of languages in training. We put together all languages which behave similarly in the first phase of the distillation process and generated the expert teacher-assistants for each group of languages. As we clustered languages based on four different language representations capturing different linguistic features, we then adaptively distill knowledge from all related teacher-assistant models to the ultimate student in each mini-batch of training per language. Experimental results on 53 languages to English show our approach's effectiveness to reduce negative transfer in MNMT. This work was accepted for publication in EMNLP 2021.

6.2 Future Directions

In this section, we list a few possible future directions to the findings of this thesis.

- In this thesis, we focused on low-resource NMT and NLG tasks. However, most of the

approaches introduced are general and can be applied to other low-resource NLP applications such as question-answering, grammar and text correction, summarization, and simplification, to name but a few.

- In this thesis, we explored the classic NLG task, i.e., data-to-text generation using transfer learning from NMT to NLG when we have only numerical and textual information as the meta-data. Another possible interesting avenue to explore is transitioning from NMT to NLG when we have joint representations of different modalities (e.g., image and text) as meta-data and document-level text as the output. The other appealing extension can be a cycled transfer learning to fill missing modalities given the observed ones and the generated output to generate more representative meta-data.
- All knowledge distillation approaches used throughout this thesis are based on output logits. Another possible approach could be distillation on encoder outputs by reducing the size of the encoder unit, the number of encoder units, or both (Ganesh et al., 2020). More specifically, if the huge pre-trained models like BERT or mBART are used in the teacher sides for both ADK and HKD, distillation on encoder outputs may lead to more compact representations in the student.
- In our KHD approach, we focused on a forward pass in distillation by distilling the student using teachers' knowledge. It is also interesting to study an end-to-end HKD approach by adding a backward HKD pass compared to the forward HKD pass described in this thesis.

To conclude, the findings presented in this thesis have shed new light on how transitioning knowledge from high-resource NMT models can effectively improve the low-resource NMT and NLG models in both bilingual and multilingual settings. The previous works on low-resource NMT and NLG scenarios usually rely on one-to-one transfer learning approaches on a single task, which cannot effectively exploit models trained with multiple high-resource language pairs in different tasks for the target language pair of interest. Therefore, this work is a step towards transfer learning by adaptively transitioning knowledge

from different high-resource language pairs with complementary syntactic and/or semantic impact on the low-resource models. We hope this work motivates research in this domain with a greater propensity towards the most effective knowledge transition schemes in low-resource NLP domains.

Bibliography

Ahn, S., Hu, S. X., Damianou, A. C., Lawrence, N. D., & Dai, Z. (2019). Variational information distillation for knowledge transfer. CVPR.

Angeli, G., Liang, P., & Klein, D. (2010). A Simple Domain-Independent Probabilistic Approach to Generation. In *Emnlp*.

Arivazhagan, N., Bapna, A., Firat, O., Aharoni, R., Johnson, M., & Macherey, W. (2019). The missing ingredient in zero-shot neural machine translation. *arXiv preprint arXiv:1903.07091*.

Ba, J. L., Kiros, J. R., & Hinton, G. E. (2016). Layer normalization. *stat*, *1050*, 21.

Ba, L. J., & Caruana, R. (2014). Do deep nets really need to be deep? NeurIPS.

Bahdanau, D., Cho, K., & Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Banaee, H., Ahmed, M. U., & Loutfi, A. (2013). Towards nlg for physiological data monitoring with body area networks. In *14th european workshop on natural language generation, sofia, bulgaria, august 8-9, 2013* (pp. 193–197).

Banerjee, S., & Lavie, A. (2005, June). 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* (pp. 65–72). Ann Arbor, MI, USA: ACL.

Bannard, C., & Callison-Burch, C. (2005). Paraphrasing with bilingual parallel corpora. In

Proceedings of the 43rd annual meeting of the association for computational linguistics (acl'05) (pp. 597–604).

Bar-Hillel, Y. (1964). *Language and information: Selected essays on their theory and application*. Addison-Wesley Reading.

Bartoli, A., De Lorenzo, A., Medvet, E., & Tarlao, F. (2016). Your paper has been accepted, rejected, or whatever: Automatic generation of scientific paper reviews. In *International conference on availability, reliability, and security* (pp. 19–28).

Barzilay, R., & Lapata, M. (2005). Collective content selection for concept-to-text generation. In *Proceedings of human language technology conference and conference on empirical methods in natural language processing* (pp. 331–338).

Bender, E. M. (2011). On achieving and evaluating language-independence in nlp. *Linguistic Issues in Language Technology*, *6*(3), 1–26.

Berard, A., Ioan, C., & Roux, C. (2019). NAVER LABS Europe's Systems for the WMT19 Machine Translation Robustness Task. In *Wmt - shared task paper.*

Berg-Kirkpatrick, T., & Klein, D. (2010). Phylogenetic grammar induction. In *Proceedings of the 48th annual meeting of the association for computational linguistics* (pp. 1288–1297).

Besag, J. (1975). Statistical analysis of non-lattice data. *J. of the Royal Statist. Soc. Series D (The Statistician)*, *24*(3), 179–195. doi: 10.2307/2987782

Bickel, B. (2007). Typology in the 21st century: Major current developments. *Linguistic Typology*, *11*(1), 239–251.

Bjerva, J., Östling, R., Veiga, M. H., Tiedemann, J., & Augenstein, I. (2019). What do language representations really represent? *Computational Linguistics*, *45*(2), 381–389.

Bowman, S., Vilnis, L., Vinyals, O., Dai, A., Jozefowicz, R., & Bengio, S. (2016). Generating sentences from a continuous space. In *Proceedings of the 20th signll conference on computational natural language learning* (pp. 10–21).

Brown, J., Frishkoff, G., & Eskenazi, M. (2005). Automatic question generation for vocabulary assessment. In *Proceedings of human language technology conference and*

conference on empirical methods in natural language processing (pp. 819–826).

Brown, P. F., Cocke, J., Della Pietra, S. A., Della Pietra, V. J., Jelinek, F., Lafferty, J., ... Roossin, P. S. (1990). A statistical approach to machine translation. *Computational linguistics*, *16*(2), 79–85.

Bucilua, C., Caruana, R., & Niculescu-Mizil, A. (2006). Model compression. In *Proceedings of the 12th acm sigkdd international conference on knowledge discovery and data mining* (pp. 535–541).

Campbell, L. (2008). *Ethnologue: Languages of the world.* JSTOR.

Campbell, L. (2013). *Historical linguistics*. Edinburgh University Press.

Cettolo, M., Girardi, C., & Federico, M. (2012). Wit3: Web inventory of transcribed and translated talks. In *Conference of european association for machine translation* (pp. 261–268).

Cettolo, M., Niehues, J., Stüker, S., Bentivogli, L., & Federico, M. (2014). Report on the 11th IWSLT evaluation campaign, IWSLT 2014. In *Proceedings of the international workshop on spoken language translation, hanoi, vietnam* (Vol. 57).

Chatzikoumi, E. (2020). How to evaluate machine translation: A review of automated and human metrics. *Natural Language Engineering*, *26*(2), 137–161. doi: 10.1017/S1351324919000469

Chen, D. L., & Mooney, R. J. (2008). Learning to sportscast: a test of grounded language acquisition. In *Proceedings of the 25th international conference on machine learning* (pp. 128–135).

Chen, M. X., Firat, O., Bapna, A., Johnson, M., Macherey, W., Foster, G., ... others (2018). 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)* (pp. 76–86).

Chen, S. F., & Goodman, J. (1999). An empirical study of smoothing techniques for language modeling. *Computer Speech & Language*, *13*(4), 359–394.

Chen, Y., Li, V. O., Cho, K., & Bowman, S. (2018, November). A stable and effective learning strategy for trainable greedy decoding. In *Proceedings of the 2018 conference on*

empirical methods in natural language processing (pp. 380–390). Brussels, Belgium: ACL. doi: 10.18653/v1/D18-1035

Chen, Y., Liu, Y., Cheng, Y., & Li, V. O. (2017). A teacher-student framework for zero-resource neural machine translation. *arXiv preprint arXiv:1705.00753*.

Chen, Y., Liu, Y., & Li, V. (2018). Zero-resource neural machine translation with multi-agent communication game. In *Proceedings of the aaai conference on artificial intelligence* (Vol. 32).

Cheng, Y. (2019). Joint training for pivot-based neural machine translation. In *Joint training for neural machine translation* (pp. 41–54). Springer.

Cheng, Y., Wang, D., Zhou, P., & Zhang, T. (2017). A survey of model compression and acceleration for deep neural networks. *arXiv e-prints*, arXiv–1710.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Chowdhury, K. D., España-Bonet, C., & van Genabith, J. (2020). Understanding translationese in multi-view embedding spaces. In *Proceedings of the 28th international conference on computational linguistics* (pp. 6056–6062).

Clarke, J., & Lapata, M. (2010). Discourse constraints for document compression. *Computational Linguistics*, *36*(3), 411–441.

Clerwall, C. (2014). Enter the robot journalist: Users' perceptions of automated content. *Journalism practice*, *8*(5), 519–531.

Cohen, N. S. (2015). From pink slips to pink slime: Transforming media labor in a digital age. *The Communication Review*, *18*(2), 98–122.

Cohen, S. B., Das, D., & Smith, N. A. (2011). Unsupervised structure prediction with non-parallel multilingual guidance. In *Proceedings of the 2011 conference on empirical methods in natural language processing* (pp. 50–61).

Cohen, S. B., & Smith, N. A. (2009). Shared logistic normal distributions for soft parameter tying in unsupervised grammar induction. In *Proceedings of human language technologies: The 2009 annual conference of the north american chapter of the asso-*

ciation for computational linguistics (pp. 74–82).

Collins, C., & Kayne, R. (2009). *Syntactic structures of the world's language: A cross-linguistic database.*

Collins, M., Koehn, P., & Kučerová, I. (2005). Clause restructuring for statistical machine translation. In *Proceedings of the 43rd annual meeting of the association for computational linguistics (acl'05)* (pp. 531–540).

Dabre, R., Chu, C., & Kunchukuttan, A. (2020). A survey of multilingual neural machine translation. *ACM Computing Surveys (CSUR)*, *53*(5), 1–38.

Dabre, R., Fujita, A., & Chu, C. (2019). Exploiting multilingualism through multistage fine-tuning for low-resource 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)* (pp. 1410–1416).

Dabre, R., Nakagawa, T., & Kazawa, H. (2017). An empirical study of language relatedness for transfer learning in neural machine translation. In *Proceedings of the 31st pacific asia conference on language, information and computation* (pp. 282–286).

Dale, R. (1989). Generating referring expressions in a domain of objects and processes. *Annexe Thesis Digitisation Project 2018 Block 20*.

Dale, R., Anisimoff, I., & Narroway, G. (2012). Hoo 2012: A report on the preposition and determiner error correction shared task. In *Proceedings of the seventh workshop on building educational applications using nlp* (pp. 54–62).

Dale, R., & Reiter, E. (1997). Building applied natural language generation systems. *Journal of Natural Language Engineering*, *3*(1), 57–87.

Daniel, M. (2011). Linguistic typology and the study of language. In *The oxford handbook of linguistic typology.*

Das, D., & Petrov, S. (2011). Unsupervised part-of-speech tagging with bilingual graph-based projections. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies* (pp. 600–609).

Delavenay, E., & Delavenay, K. M. (1960). *An introduction to machine translation*. Thames

and Hudson London.

De Marneffe, M.-C., Dozat, T., Silveira, N., Haverinen, K., Ginter, F., Nivre, J., & Manning, C. D. (2014). Universal stanford dependencies: A cross-linguistic typology. In *Lrec* (Vol. 14, pp. 4585–4592).

Dong, D., Wu, H., He, W., Yu, D., & Wang, H. (2015). Multi-task learning for multiple language translation. 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)* (pp. 1723–1732).

Dryer, M. S., & Haspelmath, M. (2013). *The world atlas of language structures.*

Duboue, P., & McKeown, K. (2001). Empirically estimating order constraints for content planning in generation. In *Proceedings of the 39th annual meeting of the association for computational linguistics* (pp. 172–179).

Duboué, P. A., & McKeown, K. R. (2003). Statistical acquisition of content selection rules for natural language generation. In *Emnlp.*

Eberhard, D., Simons, G., & Fennig, C. (2019). What are the largest language families. *Ethnologue: Languages of the World*.

Edunov, S., Ott, M., Auli, M., & Grangier, D. (2018). Understanding Back-Translation at Scale. In *EMNLP*.

Finn, C., Abbeel, P., & Levine, S. (2017). Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning* (pp. 1126–1135).

Firat, O., Cho, K., Sankaran, B., Vural, F. T. Y., & Bengio, Y. (2017). Multi-way, multilingual neural machine translation. *Computer Speech & Language*, *45*, 236–252.

Ganesh, P., Chen, Y., Lou, X., Khan, M. A., Yang, Y., Chen, D., ... Nakov, P. (2020). Compressing large-scale transformer-based models: A case study on bert. *arXiv preprint arXiv:2002.11985*.

Gatt, A., & Krahmer, E. (2018). Survey of the state of the art in natural language generation: Core tasks, applications and evaluation. *Journal of Artificial Intelligence Research*, *61*, 65–170.

Goldberg, E., Driedger, N., & Kittredge, R. I. (1994). Using natural-language processing to

produce weather forecasts. *IEEE Expert*, *9*(2), 45–53.

Gou, J., Yu, B., Maybank, S. J., & Tao, D. (2020). Knowledge distillation: A survey. *arXiv preprint arXiv:2006.05525*.

Graves, A. (2012). Sequence transduction with recurrent neural networks. In *Representation learning workshop at the international conference on machine learning*. Edinburgh, Scotland, UK: PMLR.

Gu, J., Hassan, H., Devlin, J., & Li, V. O. (2018). Universal neural machine translation for extremely low resource languages. In *Proceedings of the 2018 conference of the north american chapter of the association for computational linguistics: Human language technologies, volume 1 (long papers)* (pp. 344–354).

Gu, J., Wang, Y., Chen, Y., Li, V. O., & Cho, K. (2018). Meta-learning for low-resource neural machine translation. In *Proceedings of the 2018 conference on empirical methods in natural language processing* (pp. 3622–3631).

Ha, T.-L., Niehues, J., & Waibel, A. (n.d.). Toward multilingual neural machine translation with universal encoder and decoder. *Institute for Anthropomatics and Robotics*, *2*(10.12), 16.

Ha, T.-L., Niehues, J., & Waibel, A. (2016). Toward multilingual neural machine translation with universal encoder and decoder. *Institute for Anthropomatics and Robotics*, *2*(17/03), 17.

Hana, J., Feldman, A., & Brew, C. (2004). A resource-light approach to russian morphology: Tagging russian using czech resources. In *Proceedings of the 2004 conference on empirical methods in natural language processing* (pp. 222–229).

Hayashi, H., Oda, Y., Birch, A., Constas, I., Finch, A., Luong, M.-T., ... Sudoh, K. (2019). Findings of the Third Workshop on Neural Generation and Translation. In *Proceedings of the third workshop on neural generation and translation (wngt)*.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 770–778).

Heo, B., Lee, M., Yun, S., & Choi, J. Y. (2019). Knowledge transfer via distillation of acti-

vation boundaries formed by hidden neurons. In *Proceedings of the aaai conference on artificial intelligence* (Vol. 33, pp. 3779–3787).

Hinton, G., Vinyals, O., & Dean, J. (2015). Distilling the knowledge in a neural network. In *Nips deep learning and representation learning workshop.*

Hoang, C. D. V., Haffari, G., & Cohn, T. (2017). Towards decoding as continuous optimisation in neural machine translation. In *Proceedings of the 2017 conference on empirical methods in natural language processing* (pp. 146–156).

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, *9*(8), 1735–1780.

Holmes-Higgin, P. (1994). Text generation—using discourse strategies and focus constraints to generate natural language text by kathleen r. mckeown, cambridge university press, 1992, pp 246,£ 13.95, isbn 0-521-43802-0. *The Knowledge Engineering Review*, *9*(4), 421–422.

Hovy, E. H. (1993). Automated discourse generation using discourse structure relations. *Artificial intelligence*, *63*(1-2), 341–385.

Huang, Z., & Wang, N. (2017). Like what you like: Knowledge distill via neuron selectivity transfer. *arXiv preprint arXiv:1707.01219*.

Hwa, R., Resnik, P., Weinberg, A., Cabezas, C., & Kolak, O. (2005). Bootstrapping parsers via syntactic projection across parallel texts. *Natural language engineering*, *11*(3), 311–326.

Iso, H., Uehara, Y., Ishigaki, T., Noji, H., Aramaki, E., Kobayashi, I., ... Takamura, H. (2020). Learning to select, track, and generate for data-to-text. *Journal of Natural Language Processing*, *27*(3), 599–626.

Johnson, M., Schuster, M., Le, Q. V., Krikun, M., Wu, Y., Chen, Z., ... others (2017). Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, *5*, 339–351.

Junczys-Dowmunt, M. (2019, August). Microsoft translator at WMT 2019: Towards large-scale document-level neural machine translation. In *Proceedings of the 4th conference on machine translation* (pp. 225–233). Florence, Italy: ACL. doi: 10.18653/v1/

W19-5321

Jurafsky, D., & Martin, J. H. (2009). *Speech and language processing* (2nd ed.). Upper Saddle River, NJ, USA: Prentice-Hall, Inc.

Kauchak, D., & Barzilay, R. (2006). Paraphrasing for automatic evaluation. In *Proceedings of the human language technology conference of the naacl, main conference* (pp. 455–462).

Khapra, M. M., Joshi, S., Chatterjee, A., & Bhattacharyya, P. (2011). Together we can: Bilingual bootstrapping for wsd. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies* (pp. 561–569).

Kim, J., & Mooney, R. (2010). Generative alignment and semantic parsing for learning from ambiguous supervision. In *Coling 2010: Posters* (pp. 543–551).

Kim, J., Park, S., & Kwak, N. (2018). Paraphrasing complex network: network compression via factor transfer. In *Proceedings of the 32nd international conference on neural information processing systems* (pp. 2765–2774).

Kim, Y., Gao, Y., & Ney, H. (2019). Effective cross-lingual transfer of neural machine translation models without shared vocabularies. In *Proceedings of the 57th annual meeting of the association for computational linguistics* (pp. 1246–1257).

Kim, Y., Jernite, Y., Sontag, D., & Rush, A. (2016). Character-aware neural language models. In *Proceedings of the aaai conference on artificial intelligence* (Vol. 30).

Kim, Y., Petrov, P., Petrushkov, P., Khadivi, S., & Ney, H. (2019). Pivot-based transfer learning for neural machine translation between non-english languages. 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)* (pp. 865–875).

Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. In Y. Bengio & Y. LeCun (Eds.), *3rd international conference on learning representations, ICLR 2015, san diego, ca, usa, may 7-9, 2015, conference track proceedings.*

Kocmi, T., & Bojar, O. (2018). Trivial transfer learning for low-resource neural machine

translation. In *Proceedings of the third conference on machine translation: Research papers* (pp. 244–252).

Koehn, P. (2009). *Statistical machine translation*. Cambridge University Press.

Koehn, P., & Knowles, R. (2017). Six challenges for neural machine translation. *ACL 2017*, 28.

Koehn, P., Och, F. J., & Marcu, D. (2003). Statistical phrase-based translation. In *Proceedings of the 2003 conference of the north american chapter of the association for computational linguistics on human language technology-volume 1* (pp. 48–54).

Komodakis, N., & Zagoruyko, S. (2017). Paying more attention to attention: improving the performance of convolutional neural networks via attention transfer. In *Iclr*.

Konstas, I., & Lapata, M. (2013). A global model for concept-to-text generation. *Journal of Artificial Intelligence Research (JAIR)*.

Kraus, D. (2003). Text generation in clinical medicine-a review. *Methods Inf. Med*, *42*.

Kudo, T., & Richardson, J. (2018). SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing. In *EMNLP*.

Kudugunta, S., Bapna, A., Caswell, I., & Firat, O. (2019). Investigating multilingual nmt representations at scale. 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)* (pp. 1565–1575).

Kukich, K. (1983). Design of a Knowledge-Based Report Generator. In *Acl*.

Kukich, K. (1992). Techniques for automatically correcting words in text. *Acm Computing Surveys (CSUR)*, *24*(4), 377–439.

Lample, G., Conneau, A., Denoyer, L., & Ranzato, M. (2017). Unsupervised machine translation using monolingual corpora only. *arXiv preprint arXiv:1711.00043*.

Lavie, A., & Agarwal, A. (2007). METEOR: An automatic metric for mt evaluation with high levels of correlation with human judgments. In *Proceedings of the 2nd workshop on statistical machine translation* (pp. 228–231). Prague, Czech Republic: ACL.

Lebret, R., Grangier, D., & Auli, M. (2016). Neural Text Generation from Structured Data with Application to the Biography Domain. In *Emnlp*.

LeCun, Y. (1988). A theoretical framework for back-propagation. In D. Touretzky, G. Hinton, & T. Sejnowski (Eds.), *Proceedings of the 1988 connectionist models summer school* (pp. 21–28). Pittsburg, PA, USA: Morgan Kaufmann.

Lee, J., Cho, K., & Hofmann, T. (2017). Fully character-level neural machine translation without explicit segmentation. *Transactions of the Association for Computational Linguistics*, *5*, 365–378.

Leppänen, L., Munezero, M., Granroth-Wilding, M., & Toivonen, H. (2017). Data-driven news generation for automated journalism. In *Proceedings of the 10th international conference on natural language generation* (pp. 188–197).

Li, G., Crego, J. M., & Senellart, J. (2019). Systran@ wngt 2019: Dgt task. In *Proceedings of the 3rd workshop on neural generation and translation* (pp. 262–267).

Liang, P., Jordan, M. I., & Klein, D. (2009). Learning semantic correspondences with less supervision. 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* (pp. 91–99).

Lin, C.-Y. (2004). Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out* (pp. 74–81).

Littell, P., Mortensen, D. R., Lin, K., Kairis, K., Turner, C., & Levin, L. (2017a). Uriel and lang2vec: Representing languages as typological, geographical, and phylogenetic vectors. In *Proceedings of the 15th conference of the european chapter of the association for computational linguistics: Volume 2, short papers* (pp. 8–14). Association for Computational Linguistics.

Littell, P., Mortensen, D. R., Lin, K., Kairis, K., Turner, C., & Levin, L. (2017b). Uriel and lang2vec: Representing languages as typological, geographical, and phylogenetic vectors. In *Proceedings of the 15th conference of the european chapter of the association for computational linguistics: Volume 2, short papers* (pp. 8–14).

Liu, J., Wen, D., Gao, H., Tao, W., Chen, T.-W., Osa, K., & Kato, M. (2019). Knowledge representing: Efficient, sparse representation of prior knowledge for knowledge distillation. In *2019 ieee/cvf conference on computer vision and pattern recognition*

workshops (cvprw) (pp. 638–646).

Lu, Y., Keung, P., Ladhak, F., Bhardwaj, V., Zhang, S., & Sun, J. (2018). A neural interlingua for multilingual machine translation. In *Proceedings of the third conference on machine translation: Research papers* (pp. 84–92).

Lui, M., & Baldwin, T. (2012). Langid.Py: An Off-the-shelf Language Identification Tool. In *Proceedings of the acl 2012 system demonstrations.*

Luong, M.-T., & Manning, C. D. (2016). Achieving open vocabulary neural machine translation with hybrid word-character models. *arXiv preprint arXiv:1604.00788*.

Maddieson, I., Flavier, S., Marsico, E., Coupé, C., & Pellegrino, F. (2013). Lapsyd: Lyon-albuquerque phonological systems database. In *Proceedings of the annual conference of the international speech communication association, interspeech.*

Maimaiti, M., Liu, Y., Luan, H., & Sun, M. (2019). Multi-round transfer learning for low-resource nmt using multiple high-resource languages. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*, *18*(4), 1–26.

Malaviya, C., Neubig, G., & Littell, P. (2017). Learning language representations for typology prediction. *arXiv preprint arXiv:1707.09569*.

Maruf, S., & Haffari, G. (2018, July). Document context neural machine translation with memory networks. In *Proceedings of the 56th annual meeting of the association for computational linguistics* (pp. 1275–1284). Melbourne, Australia: ACL. doi: 10.18653/v1/P18-1118

Maruf, S., & Haffari, G. (2019, November). Monash University's submissions to the WNGT 2019 document translation task. In *Proceedings of the 3rd workshop on neural generation and translation* (pp. 256–261). Hong Kong, China: ACL. doi: 10.18653/v1/D19-5628

Maruf, S., Saleh, F., & Haffari, G. (2021). A survey on document-level neural machine translation: Methods and evaluation. *ACM Computing Surveys (CSUR)*, *54*(2), 1–36.

McDonald, R., Petrov, S., & Hall, K. (2011). Multi-source transfer of delexicalized dependency parsers. In *Proceedings of the 2011 conference on empirical methods in natural language processing* (pp. 62–72).

McKeown, K. R. (1985). *Text Generation: Using Discourse Strategies and Focus Constraints to Generate Natural Language Text*. Cambridge University Press.

Mei, H., Bansal, M., & Walter, M. R. (2016). What to talk about and how? Selective Generation using LSTMs with Coarse-to-Fine Alignment. In *Naacl-hlt.*

Michaelis, S. M., Maurer, P., Haspelmath, M., & Huber, M. (2013). *Atlas of pidgin and creole language structures online. leipzig: Max planck institute for evolutionary anthropology.*

Mikolov, T., Karafiát, M., Burget, L., Černockỳ, J., & Khudanpur, S. (2010). Recurrent neural network based language model. In *Eleventh annual conference of the international speech communication association.*

Mirzadeh, S. I., Farajtabar, M., Li, A., Levine, N., Matsukawa, A., & Ghasemzadeh, H. (2020). Improved knowledge distillation via teacher assistant. In *Proceedings of the aaai conference on artificial intelligence* (Vol. 34, pp. 5191–5198).

Molina, M., Stent, A., & Parodi, E. (2011). Generating automated news to explain the meaning of sensor data. In *International symposium on intelligent data analysis* (pp. 282–293).

Montal, T., & Reich, Z. (2017). I, robot. you, journalist. who is the author? authorship, bylines and full disclosure in automated journalism. *Digital journalism*, *5*(7), 829–849.

Mooney, R. J., & Wong, Y. W. (2007). Generation by Inverting a Semantic Parser that Uses Statistical Machine Translation. In *Naacl-hlt.*

Moran, S., & McCloy, D. (Eds.). (2019). *Phoible 2.0.* Jena: Max Planck Institute for the Science of Human History.

Müller, R., Kornblith, S., & Hinton, G. E. (2019). When does label smoothing help? In H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. B. Fox, & R. Garnett (Eds.), *Advances in neural information processing systems 32: Annual conference on neural information processing systems 2019, neurips 2019, 8-14 december 2019, vancouver, bc, canada* (pp. 4696–4705).

Murthy, R., Kunchukuttan, A., & Bhattacharyya, P. (2019). Addressing word-order di-

vergence in multilingual neural machine translation for extremely low resource languages. 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)* (pp. 3868–3873).

Naseem, T., Chen, H., Barzilay, R., & Johnson, M. (2010). Using universal linguistic knowledge to guide grammar induction. In *Proceedings of the 2010 conference on empirical methods in natural language processing* (pp. 1234–1244).

Navigli, R., & Ponzetto, S. P. (2012). Joining forces pays off: Multilingual joint word sense disambiguation. In *Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning* (pp. 1399–1410).

Neubig, G. (2017). Neural machine translation and sequence-to-sequence models: A tutorial. *arXiv preprint arXiv:1703.01619*.

Neubig, G., & Hu, J. (2018). Rapid adaptation of neural machine translation to new languages. In *Proceedings of the 2018 conference on empirical methods in natural language processing* (pp. 875–880).

Ng, N., Yee, K., Baevski, A., Ott, M., Auli, M., & Edunov, S. (2019a). Facebook FAIR's WMT19 News Translation Task Submission. In *Wmt - shared task papers.*

Ng, N., Yee, K., Baevski, A., Ott, M., Auli, M., & Edunov, S. (2019b, August). Facebook FAIR's WMT19 news translation task submission. In *Proceedings of the 4th conference on machine translation* (pp. 314–319). Florence, Italy: ACL. doi: 10.18653/v1/W19-5333

Nivre, J., De Marneffe, M.-C., Ginter, F., Goldberg, Y., Hajic, J., Manning, C. D., ... others (2016). Universal dependencies v1: A multilingual treebank collection. In *Proceedings of the tenth international conference on language resources and evaluation (lrec'16)* (pp. 1659–1666).

Och, F. J., & Ney, H. (2003). A systematic comparison of various statistical alignment models. *Computational linguistics*, *29*(1), 19–51.

O'Horan, H., Berzak, Y., Vulić, I., Reichart, R., & Korhonen, A. (2016). Survey on the

use of typological information in natural language processing. In *Proceedings of coling 2016, the 26th international conference on computational linguistics: Technical papers* (pp. 1297–1308).

Oncevay, A., Haddow, B., & Birch, A. (2020). Bridging linguistic typology and multilingual machine translation with multi-view language representations. *arXiv preprint arXiv:2004.14923*.

Östling, R., & Tiedemann, J. (2016). Continuous multilinguality with language vectors. *arXiv preprint arXiv:1612.07486*.

Ott, M., Edunov, S., Baevski, A., Fan, A., Gross, S., Ng, N., ... Auli, M. (2019). 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)* (pp. 48–53).

Ott, M., Edunov, S., Grangier, D., & Auli, M. (2018). Scaling Neural Machine Translation. In *WMT*.

Padó, S., & Lapata, M. (2005). Cross-linguistic projection of role-semantic information. In *Proceedings of human language technology conference and conference on empirical methods in natural language processing* (pp. 859–866).

Pan, S. J., & Yang, Q. (2009). A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, *22*(10), 1345–1359.

Paolillo, J. C., & Das, A. (2006). Evaluating language statistics: The ethnologue and beyond. *Contract report for UNESCO Institute for Statistics*.

Papineni, K., Roukos, S., Ward, T., & Zhu, W. (2002c). BLEU: a method for automatic evaluation of machine translation. In *ACL*.

Papineni, K., Roukos, S., Ward, T., & Zhu, W.-J. (2002a). BLEU: A method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics* (pp. 311–318). Philadelphia, PA, USA: ACL. doi: 10.3115/1073083.1073135

Papineni, K., Roukos, S., Ward, T., & Zhu, W.-J. (2002b). BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the*

association for computational linguistics (pp. 311–318).

Petrov, S., Das, D., & McDonald, R. (2012). A universal part-of-speech tagset. In *Proceedings of the eighth international conference on language resources and evaluation (lrec'12)* (pp. 2089–2096).

Pires, T., Schlinger, E., & Garrette, D. (2019). How multilingual is multilingual bert? In *Proceedings of the 57th annual meeting of the association for computational linguistics* (pp. 4996–5001).

Plachouras, V., Smiley, C., Bretz, H., Taylor, O., Leidner, J. L., Song, D., & Schilder, F. (2016). Interacting with financial data using natural language. In *Proceedings of the 39th international acm sigir conference on research and development in information retrieval* (pp. 1121–1124).

Puduppully, R., Dong, L., & Lapata, M. (2019a). Data-to-Text Generation with Content Selection and Planning. In *Proceedings of the aaai conference on artificial intelligence.*

Puduppully, R., Dong, L., & Lapata, M. (2019b). Data-to-text generation with entity modeling. In *Proceedings of the 57th annual meeting of the association for computational linguistics* (pp. 2023–2035).

Puduppully, R., & Lapata, M. (2021). Data-to-text generation with macro planning. *Transactions of the Association for Computational Linguistics*, *9*, 510–527.

Puduppully, R., Mallinson, J., & Lapata, M. (2019). University of edinburgh's submission to the document-level generation and translation shared task. In *Proceedings of the 3rd workshop on neural generation and translation* (pp. 268–272).

Qi, Y., Sachan, D., Felix, M., Padmanabhan, S., & Neubig, G. (2018, June). When and why are pre-trained word embeddings useful for neural machine translation? In *Proc. of the 2018 conference of the north American chapter of the association for computational linguistics: Human language technologies, volume 2 (short papers).* New Orleans, Louisiana: ACL.

Raghu, M., Gilmer, J., Yosinski, J., & Sohl-Dickstein, J. (2017). SVCCA: singular vector canonical correlation analysis for deep learning dynamics and interpretability. In *Proceedings of the 31st international conference on neural information processing*

systems (pp. 6078–6087).

Reiter, E., & Dale, R. (1997). Building applied natural language generation systems. *Natural Language Engineering*, *3*(1), 57–87.

Reiter, E., & Dale, R. (2000a). *Building natural language generation systems*. Cambridge University Press.

Reiter, E., & Dale, R. (2000b). *Building natural language generation systems*. USA: Cambridge University Press.

Reiter, E., Mellish, C., & Levine, J. (1995). Automatic generation of technical documentation. *Applied Artificial Intelligence an International Journal*, *9*(3), 259–287.

Rijsbergen, C. J. V. (1979). *Information retrieval* (2nd ed.). Newton, MA, USA: Butterworth-Heinemann.

Robins, A. (1995). Catastrophic forgetting, rehearsal and pseudorehearsal. *Connection Science*, *7*(2), 123–146.

Romero, A., Ballas, N., Kahou, S. E., Chassang, A., Gatta, C., & Bengio, Y. (2015). Fitnets: Hints for thin deep nets. ICLR.

Rosenstein, M. T. (2005). To transfer or not to transfer. In *Nips-2005 workshop on transfer learning* (Vol. 898, p. 1-4).

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning internal representations by error propagation. In *Parallel distributed processing: Explorations in the microstructure of cognition, vol. 1: Foundations* (p. 318–362). Cambridge, MA, USA: MIT Press.

Russell, S., & Norvig, P. (2010). *Artificial intelligence: A modern approach* (3rd ed.). Prentice Hall.

Saleh, F., Bérard, A., Calapodescu, I., & Besacier, L. (2019). Naver labs europe's systems for the document-level generation and translation task at wngt 2019. *EMNLP-IJCNLP 2019*, 273.

Saleh, F., Buntine, W., & Haffari, G. (2020). Collective wisdom: Improving low-resource neural machine translation using adaptive knowledge distillation. In *Proceedings of the 28th international conference on computational linguistics* (pp. 3413–3421).

Saleh, F., Buntine, W., Haffari, G., & Du, L. (2021). Multilingual neural machine translation: Can linguistic hierarchies help? In *Findings of the association for computational linguistics: Emnlp 2021* (pp. 1313–1330).

Sarkar, D., Bali, R., & Ghosh, T. (2018). *Hands-on transfer learning with python: Implement advanced deep learning and neural network models using tensorflow and keras*. Packt Publishing Ltd.

Sennrich, R., Haddow, B., & Birch, A. (2016a). Neural Machine Translation of Rare Words with Subword Units. In *ACL*.

Sennrich, R., Haddow, B., & Birch, A. (2016b, August). Neural machine translation of rare words with subword units. In *Proceedings of the 54th annual meeting of the association for computational linguistics* (pp. 1715–1725). Berlin, Germany: ACL. doi: 10.18653/v1/P16-1162

Sennrich, R., Haddow, B., & Birch, A. (2016c, August). Neural machine translation of rare words with subword units. In *Proc. of the 54th annual meeting of the association for computational linguistics (volume 1: Long papers)* (pp. 1715–1725). Berlin, Germany: ACL.

Siddharthan, A. (2014). A survey of research on text simplification. *ITL-International Journal of Applied Linguistics*, *165*(2), 259–298.

Siddharthan, A., & MacDonald, I. (2016). Summarising news stories for children. In *Proceedings of the 9th international conference on natural language generation (inlg 2016), edinburgh, scotland.*

Snyder, B., et al. (2010). *Unsupervised multilingual learning* (Unpublished doctoral dissertation). Massachusetts Institute of Technology.

Søgaard, A., Ruder, S., & Vulić, I. (2018). On the limitations of unsupervised bilingual dictionary induction. In *Proceedings of the 56th annual meeting of the association for computational linguistics (volume 1: Long papers)* (pp. 778–788).

Stahlberg, F. (2019). Neural machine translation: A review. *arXiv preprint arXiv:1912.02047*.

Sutskever, I., Martens, J., & Hinton, G. (2011). Generating text with recurrent neural

networks. In *Proceedings of the 28th international conference on international conference on machine learning* (pp. 1017–1024).

Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *Advances in neural information processing systems* (pp. 3104–3112).

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In *2016 IEEE conference on computer vision and pattern recognition, CVPR 2016, las vegas, nv, usa, june 27-30, 2016* (pp. 2818–2826). IEEE Computer Society.

Täckström, O., McDonald, R., & Nivre, J. (2013). Target language adaptation of discriminative transfer parsers. In *The 2013 conference of the north american chapter of the association for computational linguistics: Human language technologies* (pp. 1061–1071).

Täckström, O., McDonald, R., & Uszkoreit, J. (2012). Cross-lingual word clusters for direct transfer of linguistic structure. In *The 2012 conference of the north american chapter of the association for computational linguistics: Human language technologies (naacl-hlt 2012).*

Tan, X., Chen, J., He, D., Xia, Y., Tao, Q., & Liu, T.-Y. (2019). Multilingual neural machine translation with language clustering. 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)* (pp. 962–972).

Tan, X., Ren, Y., He, D., Qin, T., Zhao, Z., & Liu, T.-Y. (2019). Multilingual neural machine translation with knowledge distillation. *arXiv preprint arXiv:1902.10461*.

Theune, M., Klabbers, E., de Pijper, J.-R., Krahmer, E., & Odijk, J. (2001). From data to speech: a general approach. *Natural Language Engineering*, *7*(1), 47–86.

Toma, P. (1977). Systran as a multilingual machine translation system. In *Proceedings of the third european congress on information systems and networks, overcoming the language barrier* (pp. 569–581).

Torrey, L., & Shavlik, J. (2010). Transfer learning. In *Handbook of research on machine learning applications and trends: algorithms, methods, and techniques* (pp. 242–264).

IGI global.

Turner, R., Sripada, S., Reiter, E., & Davy, I. P. (2007). Selecting the content of textual descriptions of geographically located events in spatio-temporal weather data. In *International conference on innovative techniques and applications of artificial intelligence* (pp. 75–88).

Van Dalen, A. (2012). The algorithms behind the headlines: How machine-written news redefines the core skills of human journalists. *Journalism practice*, *6*(5-6), 648–658.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... Polosukhin, I. (2017a). Attention is all you need. In I. Guyon et al. (Eds.), *Advances in neural information processing systems 30* (pp. 5998–6008). Curran Associates, Inc.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... Polosukhin, I. (2017b). Attention Is All You Need. In *NIPS*.

Wang, X., Tsvetkov, Y., & Neubig, G. (2020). Balancing training for multilingual neural machine translation. In *Proceedings of the 58th annual meeting of the association for computational linguistics* (pp. 8526–8537).

Wang, Z., Dai, Z., Póczos, B., & Carbonell, J. (2019). Characterizing and avoiding negative transfer. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition* (pp. 11293–11302).

Werlen, L. M., Marone, M., & Hassan, H. (2019). Selecting, planning, and rewriting: A modular approach for data-to-document generation and translation. In *Proceedings of the 3rd workshop on neural generation and translation* (pp. 289–296).

Williams, S., & Reiter, E. (2008). Generating basic skills reports for low-skilled readers. *Natural Language Engineering*, *14*(4), 495–525.

Wiseman, S., Shieber, S., & Rush, A. (2017). Challenges in Data-to-Document Generation. In *Emnlp*.

Wisniewski, G., Pécheux, N., Gahbiche-Braham, S., & Yvon, F. (2014). Cross-lingual part-of-speech tagging through ambiguous learning. In *Proceedings of the 2014 conference on empirical methods in natural language processing (emnlp)* (pp. 1779–1785).

WNGT. (2019). *The 3rd workshop on neural generation and translation (wngt 2019), held*

at at emnlp-ijcnlp 2019. Retrieved from https://sites.google.com/view/wngt19/home?authuser=0

Xu, C., Qin, T., Wang, G., & Liu, T.-Y. (2019). Polygon-net: A general framework for jointly boosting multiple unsupervised neural machine translation models. In *Ijcai* (pp. 5320–5326).

Yosinski, J., Clune, J., Bengio, Y., & Lipson, H. (2014). How transferable are features in deep neural networks? *Advances in Neural Information Processing Systems*, *27*, 3320–3328.

Young, M. L., & Hermida, A. (2015). From mr. and mrs. outlier to central tendencies: Computational journalism and crime reporting at the los angeles times. *Digital Journalism*, *3*(3), 381–397.

Zhang*, T., Kishore*, V., Wu*, F., Weinberger, K. Q., & Artzi, Y. (2020). BERTScore: Evaluating text generation with BERT. In *Proceedings of the 8th international conference on learning representations.*

Zhang, Y., & Barzilay, R. (2015). Hierarchical low-rank tensors for multilingual transfer parsing. In *Proceedings of the 2015 conference on empirical methods in natural language processing* (pp. 1857–1867).

Zhao, Y., Zhang, J., & Zong, C. (2018). Exploiting pre-ordering for neural machine translation. In *Proceedings of the eleventh international conference on language resources and evaluation (lrec 2018).*

Zoph, B., Yuret, D., May, J., & Knight, K. (2016, November). Transfer learning for low-resource neural machine translation. In *Proceedings of the 2016 conference on empirical methods in natural language processing* (pp. 1568–1575). Austin, Texas: Association for Computational Linguistics. doi: 10.18653/v1/D16-1163