Discourse-Aware Neural Machine Translation

Longyue Wang

B.Sc., M.Sc.

A dissertation submitted in fulfilment of the requirements for the award of

Doctor of Philosophy (Ph.D.)

to the

School of Computing

Dublin City University

Supervised by

Prof. Andy Way and Prof. Qun Liu

January 2019

Declaration

I hereby certify that this material, which I now submit for assessment on the programme of study leading to the award of Ph.D is entirely my own work, and that I have exercised reasonable care to ensure that the work is original, and does not to the best of my knowledge breach any law of copyright, and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of my work.

Signed:

(Candidate) ID No.: 13116045

Date: January 1, 2019

Acknowledgements

Today is the last day of 2018. At this very moment, I am sitting alone in my office and writing the last part of my doctoral thesis: acknowledgments. At the age of thirty, I finished my doctoral studies. Just as Confucius said he became independent at thirty, this age is very significant to every Chinese people, the same to me. Looking back, I appreciate all people who help me in the past thirty years.

First of all, I want to say thanks to my parents, who give birth to me and bring me up. My parents are workers in a small town in the north of China. They worked hard and stinted themselves to give me best educations. I really understand how difficult for such ordinary family to produce a "sea turtle" doctor. I hope my current small success can make them happy and proud.

Secondly, I would like to express my sincere gratitude to my elementary school teacher, Ms. Yang. She not only opened a door to "computer world" for me, but also taught me how to find the right way whenever I lose my mind. Besides, I sincerely thank my Youth League teacher, Ms. Zhang. She used her wisdom to develop my capabilities of leadership, organizing and cooperating. Eventually, I became President of the Student's Union at my university. This matter gave me great confidence and kept me successful in the future. My teacher is like a lighthouse in the sea, guiding me to reach the coast of success. I sincerely wish them always healthy and happy.

Thirdly, I would like to thank Dr. Siyou Liu. We met each other in NLP2CT lab at University of Macau in 2012. She is the most kind-hearted, wise and beautiful girl I have ever met. During the master's, we learned from each other and made progress together. In 2014, we enrolled in PhDs in different countries and started long-distance relationship. Just like the happy ending of the fairy tale, we got married in the summer of 2017 and completed our doctoral studies in the end of 2018. In the past six years, she always stands by me and supports me wordlessly. I hope we can start our new lives.

Foremost, I would like to specifically thank my supervisors, Prof. Qun Liu and Prof. Andy Way. During my PhD's studies, I received tremendous help and great encouragement

from them. In scientific research, they tirelessly taught me how to find valuable ideas, how to overcome difficulties, how to write a perfect research papers and so on. Above all, I learned from them how to be a rigorous and responsible scientific researcher. I really appreciated them and I will continue efforts to make high-level academic achievements.

Last but not least, I sincerely thank all my friends who always help and support me: my best classmates at primary and middle schools (e.g., Zhen Huang and Lu Liu), my great partners at students' union (e.g., Xiaoxue Lv and Nan Li), my kind seniors at universities (e.g., Tianshu Wu and Quan Wen), my lovely classmates at NLP2CT lab (e.g., Liang Tian and Xiaodong Zeng), my former supervisors at University of Macau (e.g., Prof. DDerek F. Wong and Prof. Lidia S. Chao), my nice flatmates in Macau (e.g., Li Li and Dr. Zhibo Wang), my PhD classmates at Dublin City University (e.g., Dr. Liangyou Li and Dr. Tengqi Ye), my great colleagues at ADAPT Centre (e.g., Eva Vanmassenhove and Dr. Jinhua Du), my mentors in industry (e.g., Dr. Zhaopeng Tu) and my big families. I can not list all of your names here but I always remember all of your invaluable help. My most sincere thank you to all of you.

Longyue Wang has received funding from Science Foundation Ireland in the ADAPT Centre for Digital Content Technology at Dublin City University funded under the SFI Research Centres Programme (Grant 13/RC/2106) cofunded under the European Regional Development Fund and the European Union Horizon 2020 research and innovation programme under grant agreement 645452 (QT21). Some work in the thesis is also partly supported by two DCU-Huawei Joint Projects: 2015-2016 (201504032-A/YB2015090061) and 2017-2018 (YBN2017080040). Some works were done when Longyue Wang was interning at Tencent AI Lab.

Contents

List of Figures

List of Tables

Abbreviations

BLEU Bilingual Evaluation Understudy. 31, 75, 77, 90, 102, 111, 112, 115

CL Computational Linguistics. 16

CR Coreference Resolution. 46

CRF Conditional Random Field. 46

CWMT China Workshop on Machine Translation. 81

DNMT Document-level Neural Machine Translation. 11, 48, 51, 68, 70–72, 122, 141, 143, 145

DP Dropped Pronoun. 8–12, 35, 45, 46, 74–77, 80–82, 84–92, 97–103, 105, 107–118, 121–123, 125, 126, 132, 134, 142–144

DPP Dropped Pronoun Position. 83–87, 91, 126, 132

DPS Dropped Pronoun Surface. 85, 91

EC Empty Category. 45, 46

GRU Gated Recurrent Unit. 24, 25

HRED Hierarchical Recurrent Encoder-Decoder. 8, 43, 44

LM Language Model. 22, 41, 76, 84, 85, 89, 90, 97, 141

LSTM Long Short-Term Memory. 24, 41

MERT Minimum Error Rate Training. 90

MLP Multi-Layer Perceptron. 9, 77, 85, 86, 90, 91

MT Machine Translation. xi, 1, 2, 4, 7–9, 11, 12, 16, 17, 22, 31, 33–36, 39, 40, 42, 44, 46, 49, 60–63, 73–75, 81, 90, 99, 140–143, 145

NLP Natural Language Processing. 1, 16, 17, 22, 32–34, 37

NMT Neural Machine Translation. xiv, 1–12, 16, 17, 22, 26, 30, 40–44, 46–51, 53–55, 57–60, 63–66, 69, 71, 99–104, 106, 110–112, 114, 117, 121–123, 131, 132, 140–145

NN Neural Network. 9, 22, 43, 44, 77, 85, 90, 100, 109

NNs Neural Networks. 3

PBSMT Phrase-based Statistical Machine Translation. 3, 18, 19, 22, 42, 43, 90

PDTB Penn Discourse Tree Bank. 37, 38

POS Part-Of-Speech. 86

pro-drop Pronoun Dropping. 8, 11, 12, 35, 75–77, 80, 81, 99–101, 121, 142

RBMT Rule-based Machine Translation. 1, 2, 16

RNN Recurrent Neural Network. xi, 8, 9, 11, 17, 22–27, 51, 52, 57, 77, 85, 86, 90, 91, 123, 134, 141

RNNLM Recurrent Neural Network Language Model. 24–26

RST Rhetorical Structure Theory. 37, 38

SMT Statistical Machine Translation. xiv, 1–9, 11, 12, 16–18, 22, 41, 42, 45, 46, 49, 63–65, 74–77, 82, 88, 92, 99–101, 141, 142, 144

SOV Subject-Object-Verb. 97

SVM Support Vector Machine. 45

SVO Subject-Verb-Object. 91, 97

ZP Zero Pronoun. 45

Discourse-Aware Neural Machine Translation

Longyue Wang

Abstract

Machine translation (MT) models usually translate a text by considering isolated sentences based on a strict assumption that the sentences in a text are independent of one another. However, it is a truism that texts have properties of connectedness that go beyond those of their individual sentences. Disregarding dependencies across sentences will harm translation quality especially in terms of coherence, cohesion, and consistency. Previously, some discourse-aware approaches have been investigated for conventional statistical machine translation (SMT). However, this is a serious obstacle for the state-of-the-art neural machine translation (NMT), which recently has surpassed the performance of SMT.

In this thesis, we try to incorporate useful discourse information for enhancing NMT models. More specifically, we conduct research on two main parts: 1) exploiting novel document-level NMT architecture; and 2) dealing with a specific discourse phenomenon for translation models.

Firstly, we investigate the influence of historical contextual information on the performance of NMT models. A cross-sentence context-aware NMT model is proposed to consider the influence of previous sentences in the same document. Specifically, this history is summarized using an additional hierarchical encoder. The historical representations are then integrated into the standard NMT model in different strategies. Experimental results on a Chinese–English document-level translation task show that the approach significantly improves upon a strong attention-based NMT system by up to +2.1 BLEU points. In addition, analysis and comparison also give insightful discussions and conclusions for this research direction.

Secondly, we explore the impact of discourse phenomena on the performance of MT. In this thesis, we focus on the phenomenon of pronoun-dropping (pro-drop), where, in pro-

drop languages, pronouns can be omitted when it is possible to infer the referent from the context. As the data for training a dropped pronoun (DP) generator is scarce, we propose to automatically annotate DPs using alignment information from a large parallel corpus. We then introduce a hybrid approach: building a neural-based DP generator and integrating it into the SMT model. Experimental results on both Chinese–English and Japanese–English translation tasks demonstrate that our approach achieves a significant improvement of up to +1.58 BLEU points with 66% F-score for DP generation accuracy.

Motivated by this promising result, we further exploit the DP translation approach for advanced NMT models. A novel reconstruction-based model is proposed to reconstruct the DP-annotated source sentence from the hidden states of either encoder or decoder, or both components. Experimental results on the same translation tasks show that the proposed approach significantly and consistently improves translation performance over a strong NMT baseline, which is trained on DP-annotated parallel data.

To avoid the errors propagated from an external DP prediction model, we finally investigate an end-to-end DP translation model. Specifically, we improve the reconstruction-based model from three perspectives. We first employ a shared reconstructor to better exploit encoder and decoder representations. Secondly, we propose to jointly learn to translate and predict DPs. In order to capture discourse information for DP prediction, we finally combine the hierarchical encoder with the DP translation model. Experimental results on the same translation tasks show that our approach significantly improves both translation performance and DP prediction accuracy.

Chapter 1

Introduction

As an active research field in Natural Language Processing (NLP), the task of Machine Translation (MT) is to translate texts from one language to another language. It is a challenging task for MT to generate high-quality translation, because computers need to not only thoroughly understand the text in the source language, but also have good knowledge of the target language (Hardmeier 2014). In the last several decades, the scientific research in the field of MT has experienced three main historical periods including Rule-based Machine Translation (RBMT) (Nirenburg et al. 1986), Statistical Machine Translation (SMT) (Koehn 2009) and Neural Machine Translation (NMT) (Kalchbrenner and Blunsom 2013, Sutskever et al. 2014), and each of these models has significantly improved the performance of MT systems.

Despite the success in both research and practical scenarios, MT systems (either RBMT, SMT or NMT) usually translate a text sentence-by-sentence based on an assumption that the sentences in a text are independent of one another. However, it is a truism that documents have the property of connectedness that goes beyond those of their individual sentences (Webber 2014). Disregarding these dependencies across sentences will negatively affect the translation quality of MT when translating a text.

Natural languages, from bottom to top, are composed of several linguistic units including word, phrase, clause, sentence, paragraph, and discourse (Asher and Lascarides 2003,

Longacre 2013). A discourse is an instance of language use whose type can be classified on the basis of such factors as grammatical and lexical choices and their distribution in main versus supportive materials, theme, style, and the framework of knowledge and expectations within which the addressee interprets the discourse (Elson and Pickett 1983, Crystal 1985, Hanks 1987, Longacre 1990). Like words in a sentence, sentences in a text are closely related to one another. In general, considering discourse information enables the building MT models which not only better understand the semantics on the source side, but also generate more coherent and consistent translations in the target langauge.

During the history of MT, researchers have integrated various discourse-aware approaches to address problems caused by the loss of extra-sentential context. The 1990s saw an intensification of research efforts aimed at endowing RBMT-translated texts with the same document and discourse properties as their source texts (Webber 2014). This included work on stylistics for MT (DiMarco and Mah 1994), target language realization of source-language discourse relations (Mitkov 1993) and of referring forms (Wada 1990, Bond and Ogura 1998) as well as anaphora resolution for generating appropriate target-language pronouns (Chan and T'sou 1999, Ferrández et al. 1999, Nakaiwa 1999). In the era of SMT, discourse was widely investigated in different aspects such as language modelling (Foster et al. 2010), discourse connectives (Meyer and Poláková 2013, Meyer and Webber 2013), lexical cohesion (Xiong et al. 2013), anaphora resolution (Le Nagard and Koehn 2010, Taira et al. 2012) and topic adaption (Su et al. 2012, Hasler et al. 2014). This research demonstrated promising results and motivated us to continue to explore discourse-aware approaches to improve the performance of MT systems in this thesis.

In recent years, NMT (Kalchbrenner and Blunsom 2013, Sutskever et al. 2014, Bahdanau et al. 2015) has made significant progress towards to constructing and utilizing a single large neural network to handle the entire translation task. Subsequently the encoder-decoder architecture was proposed by Cho et al. (2014) and Sutskever et al. (2014), in which the encoder summarizes the source sentence into a vector representation, and the decoder generates the target sentence word-by-word from the vector representation. Due to these

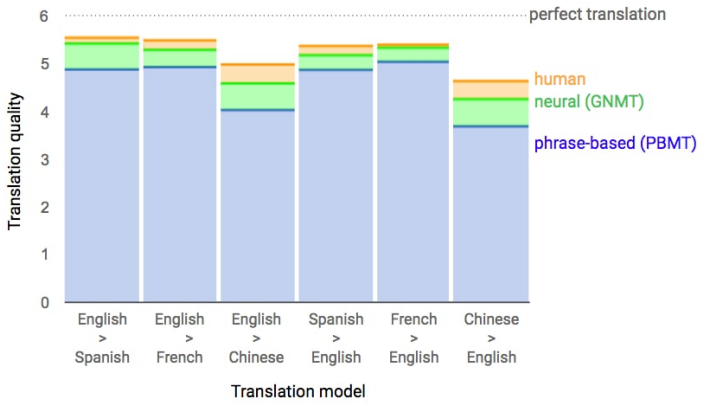


Figure 1.1: Google Research: human raters compare the quality of translations. Scores range from 0 to 6, with 0 meaning "completely nonsense translation", and 6 meaning "perfect translation".

advances in NMT, the performance of NMT has surpassed the performance of traditional SMT in various language pairs (Luong et al. 2015a). More recently, Google Research announced that they overcame many challenges to make NMT work on very large data sets and built a system that is sufficiently fast and accurate enough to provide better translations for users. They also conducted quantitative analysis on machine-translated outputs using human-rated side-by-side comparison as a metric. As shown in Figure 1.1, NMT system produces translations that are vastly improved compared to the previous Phrase-based Statistical Machine Translation (PBSMT) system (*i.e.,* the best model in SMT) in various language pairs, but especially for English–Chinese.

Motivated by the power of Neural Networks (NNs), in this thesis we integrate discourse information into state-of-the-art NMT models. However, discourse-aware approaches to NMT have received relatively little attention from the research community.[1] Through our studies in this thesis, we demonstrate that discourse-aware NMT models can improve translation quality.

[1]Most of our work was conducted from the end of 2014 to the beginning of 2018. Early in our research period, there was almost no related work in NMT.

1.1 Why does Discourse Matter to Machine Translation?

Although deep learning has significantly improved translation quality in terms of BLEU score (Papineni et al. 2002), NMT still models a text by considering isolated sentences, disregarding discourse properties (Webber 2014) including:

- document-wide properties, such as topic mix, style, register, reading level, and genre, all of which are apparent in the frequency and distribution of words, word senses, referential forms and syntactic structures;
- patterns of topical or functional sub-structure that show up in localized differences in the frequency and distribution of these elements within documents;
- patterns of discourse coherence, manifest in explicit and implicit relations between sentences (clauses), or between sentences (clauses) and referring forms, or between referring forms themselves;
- common use of reduced expressions that rely on context to convey a lot of information in very few words.

We are interested in the extent to which discourse affects the performance of MT systems when translating a text. In order to answer this question, we use an example to analyze discourse-related problems in the outputs translated by NMT and SMT models, respectively. We employed two strong NMT (Wu et al. 2016) and SMT (Koehn 2009) models to respectively translate the story *The Farmer and the Snake* in the book *Aesop's Fables* from Chinese into English.

The translation outputs are shown in Table 1.1. Overall, NMT translations are more fluent than that of SMT. For instance, "When the farmer dies, he said ..." is much better than "Farmer before his death, said ..." and the connective phrase "so that" can clearly show a causal relation between preceding and following clauses. In addition, the lexical choice of NMT model is better than the SMT model. For example, the verb "pity" is better than the adjective "poor" and the clause 'I will die'" is better than "I damn". Qualitatively, this

I/O	Text
Input	冬天[1]，农夫[2] 发现一条蛇冻僵了，[3] 他很可怜它，便[6] 把蛇放在自己怀里。蛇温暖后，[DP][4] 苏醒了过来，恢复了它的本性，咬[5] 了它的恩人一口，使他受到了致命的伤害。农夫[2] 临死前说："我该死，我怜悯恶人，应该受恶报。"
Reference	One winter a farmer found a snake stiff and frozen with cold. He had compassion on it, and taking it up, placed it in his bosom. The snake was quickly revived by the warmth, and resuming to its natural instincts, bit its benefactor, inflicting on him a mortal wound. "Oh," cried the farmer with his last breath, "I am rightly served for pitying a scoundrel."
SMT	In winter, the farmer found a snake frozen, he was very poor it, put the snake in his arms. After the snake warm, waking, [DP] resumed its nature, bite its benefactor, that he received fatal injuries. Farmer before his death, said: "I damn, I pity the wicked, should be subject to roost."
NMT	In the winter, the farmer found a snake frozen, he was very pity it, put the snake in his arms. After the snake warm, [DP] wake up, restore its nature, bite it benefactor, so that he suffered a fatal injury. When the farmer dies, he said, "I will die, and I will have mercy on the wicked.

Table 1.1: Example of a text translated by SMT and NMT systems.

illustrates why NMT is better than SMT in a quite general sense. However, we found more interesting cases from the perspective of discourse, and we summarize them as follows:

1. The Chinese word "冬天" (winter) has different translations according to its context. In general usage, it can be translated into "in winter". For more specific instances, we need to further consider whether it means "an uncertain winter" or "a known winter". If "uncertain", it should be translated into "once in a winter" or "one winter", otherwise, "in the winter". Obviously, at the beginning of the story, it means "one unknown winter". However, neither the SMT system nor the NMT system can correctly translate this word.

2. The Chinese word "农夫" (farmer) should be translated into "a farmer" when appearing for the first time. If the noun occurs again in the following sentences and refers to the same person, it should be translated into "the farmer" instead. However, the two words "农夫" are all translated incorrectly by the SMT model. Although NMT

generates the correct translation for the second one, it seems to achieve success by chance because NMT always translate the word in the same way.

3. As the first sentence in the source text, it is common to write Chinese sentences in chronicle style.[2] However, this usually makes one sentence rather long, which should be translated into multiple sentences in English. However, NMT and SMT models both translate the Chinese chronicle sentence into a long English sentence with two complete parts with incorrect connections (commas).

4. Chinese is a pro-drop language, in which certain classes of words can be omitted to make the sentence compact yet comprehensible when the identity of the pronouns can be inferred from the context. Taking the second sentence in the source text, for example, the subject "蛇" (snake) has already occurred in the subordinate clause ("蛇温暖后" (after the snake is warm)). Thus, in Chinese, the corresponding pronoun is usually dropped in the main clause ("[DP] 苏醒了过来" (the snake woke up)). However, on the English side, NMT and SMT fail to recover them and translate the missing pronouns. Note that, this an example of intra-sentential context, while Table 5.1 shows an example of inter-sentential context.

5. As seen, NMT and SMT also have tense inconsistency problem. The story happened in the past, so it should be described in the past tense. However, the words "restore" and "bite" in translation outputs are generated in the infinitive form. NMT performs even worse than SMT in terms of consistency.

6. The Chinese word "便" (and) is a discourse connective, which shows a continuation or causality relation between its preceding and following parts. Without considering the discourse structure and relation, NMT and SMT systems generate grammatically incorrect translations in terms of coherence.

In general, problems [1], [2] and [4] severely harm the translations in terms of cohesion, while problems [3] and [6] negatively affect the coherence of translations. Besides, problem

[2] In Chinese, a sentence is usually very long with multiple clauses.

[5] shows a tense inconsistency problem. Note that, discourse information exists in different kinds of contexts: 1) either the source- or target-side history sentences, or both; 2) either the preceding or following sentences, or both. For instance, to address the problem [2], we need to consider the number of "农夫" in the source-side context. However, about tense inconsistency in problem [5], target-side history information is more useful due to the lack of tense information in the source language. Besides, useful discourse information not only exists in preceding sentences such as problems [2] and [4], but also may subsist in following sentences such as problem [1]. At the beginning of the story, the first word "冬天" has no history context. It is impossible to disambiguate "uncertain" and "known" situations unless we read the following sentences or the whole document.

Via qualitative analysis, we can see that discourse is a big challenge for both NMT and SMT, although the output of NMT is more fluent than that of SMT. From the perspective of discourse, the translation quality is quite far away from a human-acceptable level. That is why we exploit effective approaches to alleviate the discourse-related problems for NMT in terms of coherence, cohesion, and consistency.

1.2 Research Questions

As discourse exists beyond sentence boundaries, some important information may be lost in a sentence-level MT system (*i.e.,* translating each sentence in a document in isolation). Previous work in SMT has shown that considering the document as a whole is helpful to resolve certain ambiguities and inconsistencies (Sammer et al. 2006, Xiao et al. 2011). Therefore, we would like to know whether we can improve translation quality by modelling cross-sentence context for NMT models. This leads to our first research question:

> **RQ 1** *What is the influence of historical contextual information on the performance of neural machine translation? Can a document-level NMT architecture alleviate inconsistency and ambiguity problems?*

However, at the beginning of studying **RQ1**, there were almost no document-level ap-

proaches for NMT models. We review temporal tasks from other research communities such as video modelling (Fakoor et al. 2016), query suggestion (Sordoni et al. 2015) as well as dialogue generation (Serban et al. 2016). Inspired by their success, we investigate a Hierarchical Recurrent Encoder-Decoder (HRED) model that can be used to model previous source sentences in order to generate each target word in the current sentence. Another aspect which motivates us to adopt this architecture for NMT is that sentence-level and document-level Recurrent Neural Network (RNN) layers can respectively model lexical dependencies and discourse relations across sentences. Another issue is how to integrate this across sentence boundaries with the standard NMT model. We explore different strategies to enhance the understanding of source sentence, aiding generating target words or controlling the historical information. Finally, we can observe that the translation quality of NMT can generally improve by considering a larger context.

After investigating a document-level architecture, we move our focus to the specific discourse phenomena. We observed that Pronoun Dropping (pro-drop) poses difficulties for MT: missing translations of pronouns, harming sentence structure and even destroying the semantics of output. Therefore, we focus on Dropped Pronoun (DP) translation problems. A number of related works investigated the use of a pipeline strategy, in which they first recover zero pronouns or empty categories in a source sentence, and then feed the preprocessed input into SMT models (Le Nagard and Koehn 2010, Taira et al. 2012, Xiang et al. 2013). However, their results were not stable, which motivates us to form our second research question:

> **RQ 2** *How do dropped pronouns affect the performance of machine translation? Is it possible to build a robust drop pronoun recovery model for statistical machine translation?*

We found that the primary challenge here is that the training data for DP recovery is small-scale. Some researchers apply manual annotation methods, which are very expensive (Yang et al. 2015). Others employ existing corpora such as empty categories in the Penn

Treebank (Chung and Gildea 2010, Xiang et al. 2013) and dropped subjects in OntoNotes (Chen and Ng 2013). However, performance is not reliable when using recovery systems trained on these small-scale or closed-domain corpora for open-domain translation tasks. Our first concern is how to automatically build a large-scale training data set for our DP recovery model. We are inspired by a point of the view "two languages are more informative than one" (Dagan et al. 1991). Considering that there exists a large amount of parallel data, we can automatically annotated DPs using alignment information.

Due to the powerful capacity of Neural Network (NN), we model DP position (DPP) detection as a sequential labelling task using RNN, and DP word (DPW) prediction as a classification task using a Multi-Layer Perceptron (MLP) model. Similarly, we regard the task of DP recovery as a pre-processing stage for MT. Although their parameters are tuned independently, this direct idea is still worth investigating. Thus, we aim to integrate the DP recovery results into SMT with different strategies.

Modeling DP for NMT has received substantially less attention, resulting in low performance in this respect even for state-of-the-art approaches. Following **RQ2**, we need to consider how to integrate our DP recovery model into the NMT framework. This leads to our third research question:

> **RQ 3** *Does neural machine translation still suffer from dropped pronoun problems? If so, how should we embed DP information into neural network models?*

In **RQ3**, we try to integrate DPs into the training phase, instead of simply using them only in the decoding step in **RQ2**. Recently, Tu et al. (2017b) proposed a novel encoder-decoder-reconstructor model to alleviate the adequacy problem, where NMT tends to repeatedly translate some source words while mistakenly ignoring other words. For example, given a Chinese sentence, the standard encoder-decoder model translates it into an English sentence and assigns a likelihood score. Then, the newly added component reconstructs the translation back to the source sentence and calculates the corresponding reconstruction

score. Linear interpolation of the two scores produces an overall score for the translation. As seen, the added reconstructor imposes a constraint that an NMT model should be able to reconstruct the input source sentence from the target-side hidden layers, which encourages the decoder to embed complete information from the source side.

Reconstruction is a standard concept in auto-encoder model, that guides the system towards learning representations that capture the underlying explanatory factors for the observed input. Therefore, we can adapt this "auto-encoder" concept to the DP translation task by embedding DP information into NMT. The built training corpus in **RQ2** can be used to construct a new triple corpus ($\mathbf{x}$, $\mathbf{y}$, $\hat{\mathbf{x}}$), where $\mathbf{x}$ and $\mathbf{y}$ are source and target sentences, and $\hat{\mathbf{x}}$ is the annotated source sentence. Then we can apply a standard encoder-decoder NMT model to translate $\mathbf{x}$, and obtain two sequences of hidden states from both the encoder and decoder. This is followed by introducing an additional *reconstructor* (Tu et al. 2017b) to reconstruct the annotated source sentence $\hat{\mathbf{x}}$ with hidden states from either the encoder or decoder, or both components.

Although we demonstrate that the model in **RQ3** can achieve significant improvements, there is still a severe drawback behind. The *testing phase* is still a pipeline method, where the DP annotation is automatically done by an external DP prediction model. However, the DP predictor only has a low accuracy, which propagates numerous errors to the translation model. Accordingly, our final research question is:

> **RQ 4** *Can we build a fully end-to-end neural model for dropped pronoun translation? Is cross-sentence context useful for dropped pronoun prediction?*

To answer this question, we need to not only consider multi-task learning approaches, but also take a new look at our document-level architecture in **RQ1**. Inspired by recent success in multi-task learning (Dong et al. 2015, Luong et al. 2016), we explore jointly learning to translate and predict DPs. We expect that the auxiliary objectives can guide the related part of the parameter matrix to learn better latent representations for both translation and DP prediction.

Another concern is that the standard NMT model is just a sentence-level model, while the DP prediction model needs document-wide information. Thus, the idea in **RQ1** inspires us to focus on discourse phenomena under a novel document-level NMT architecture. Finally, these two approaches make it possible to build a fully end-to-end DP translation model.

1.3 Thesis Structure

In this thesis, we investigate discourse-aware MT in three parts: *document-level architecture*, *specific discourse phenomena* and *combination*. Overall, the goal is to improve MT quality by considering/modelling the knowledge of discourse. This thesis comprises seven chapters including the current introductory chapter. We now introduce the topics discussed in each chapter.

In Chapter 2, we will review the background about MT models and algorithms, including both the state-of-the-art NMT and the conventional SMT. Without loss of generality, we will also provide the fundamental information on discourse and its linguistic phenomena. Finally, we will present related work on discourse-aware approaches for MT.

Part I: Document-Level Architecture

In Chapter 3, we will describe our early attempt at investigating the potential for implicitly incorporating discourse information into NMT. In our work, we propose a hierarchical RNN architecture to model cross-sentence context for NMT models. We show that our approach can significantly improve translation quality over the NMT and SMT baseline models. We also analyze the effect of global context and provide examples generated by our model. Furthermore, we compare our approach with other recent Document-level Neural Machine Translation (DNMT) models using various domains of data.

Part II: Targeting Specific Discourse Phenomena

In Chapter 4, we will move our attention to one specific discourse phenomenon: pro-

drop, which significantly affects the cohesion of MT output. First of all, we study the DP problem in pro-drop languages. Secondly, we explore an unsupervised approach to automatically build a large-scale, high-quality DP training corpus. Then we investigate NN-based approaches to recall missing pronouns for SMT. We present our experimental results on Chinese–English to show the efficiency of the proposed approach. Case studies illustrate how our approach alleviates DP problems for translation models. To validate the effect of the proposed approach, we also adopt our approach to Japanese–English translation.

In Chapter 5, we investigate alleviating DP translation problems for NMT models. We present how our reconstruction-based approach guides the hidden states (either encoder-side or decoder-side) of NMT to embed the missing DP information. Experiments on Chinese–English and Japanese–English show that the proposed approach significantly outperforms a strong NMT baseline system. In our analysis, we demonstrate that our models can produce better translations by addressing the DP translation problem.

Part III: Combination

In Chapter 6, we explore DP translation under our proposed document-level NMT architecture. We propose an end-to-end model to jointly learn translate and predict DPs with document-level context. Experimental results on the Chinese–English corpus show that our approach can accumulatively improve translation performance. In addition, the jointly learned DP prediction model significantly outperforms its external counterpart by 9% in terms of F1 score.

Chapter 7 concludes the thesis with general conclusions we draw from our experiments. We also provide some future avenues for research.

In summary, all of our proposed discourse-aware approaches and experiments are presented in Chapter 3, 4, 5 and 6, which are related to **RQ1**, **RQ2**, **RQ3** and **RQ4**, respectively. **RQ1** is studied in *document-level architecture* and **RQ2–RQ3** in *specific discourse phenomena* scenario. **RQ4** is studied in combining these two scenarios into a new one.

1.4 Publications

The published papers which are related to this thesis are as follows:

1. Longyue Wang, Zhaopeng Tu, Andy Way, Qun Liu. (2018). Learning to Jointly Translate and Predict Dropped Pronouns with a Shared Reconstruction Mechanism. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1–5, Brussels, Belgium.

2. Longyue Wang, Zhaopeng Tu, Shuming Shi, Tong Zhang, Yvette Graham, Qun Liu. (2018). Translating Pro-Drop Languages with Reconstruction Models. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, pages 4937–4945, New Orleans, Louisiana, USA.

3. Longyue Wang, Zhaopeng Tu, Andy Way, Qun Liu. (2017). Exploiting Cross-Sentence Context for Neural Machine Translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2816–2821, Copenhagen, Denmark.

4. Longyue Wang, Zhaopeng Tu, Xiaojun Zhang, Siyou Liu, Hang Li, Andy Way and Qun Liu. (2017). A Novel and Robust Approach for Pro-Drop Language Translation. In *Machine Translation*, 31(1-2), 65-87, Springer.

5. Longyue Wang, Zhaopeng Tu, Xiaojun Zhang, Hang Li, Andy Way and Qun Liu. (2016). A Novel Approach for Dropped Pronoun Translation. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 983–993, San Diego, California, USA.

6. Longyue Wang, Xiaojun Zhang, Zhaopeng Tu, Hang Li, Qun Liu. (2016). Dropped Pronoun Generation for Dialogue Machine Translation. In *Proceedings of the IEEE International Conference of Acoustics, Speech and Signal Processing*, pages 6110–6114, Shanghai, China.

7. Longyue Wang, Chris Hokamp, Tsuyoshi Okita, Xiaojun Zhang, Qun Liu. (2015). The DCU Discourse Parser for Connective, Argument Identification and Explicit Sense Classification. In *Proceedings of the 19th Conference on Computational Natural Language Learning (Shared Task)*, pages 89–94, Beijing, China.

8. Tsuyoshi Okita, Longyue Wang, Qun Liu. (2015). The DCU Discourse Parser: A Sense Classification Task. In *Proceedings of the 19th Conference on Computational Natural Language Learning (Shared Task)*, pages 71–77, Beijing, China.

Other publications I have published during my PhD are:

1. Siyou Liu, Longyue Wang, Chao-Hong Liu. (2018). Chinese–Portuguese Machine Translation: A Study on Building Parallel Corpora from Comparable Texts. In *Proceedings of the 11th Language Resources and Evaluation Conference*, pages 1–8, Miyazaki, Japan.

2. Longyue Wang, Jinhua Du, Liangyou Li, Zhaopeng Tu, Andy Way, Qun Liu. (2017). Semantics-Enhanced Task-Oriented Dialogue Translation: A Case Study on Hotel Booking. In *Proceedings of the 8th International Joint Conference on Natural Language Processing (System Demonstrations)*, pages 33–36, Taiwan, China.

3. Longyue Wang, Xiaojun Zhang, Zhaopeng Tu, Andy Way, Qun Liu. (2016). The Automatic Construction of Discourse Corpus for Dialogue Translation. In *Proceedings of the 10th Language Resources and Evaluation Conference*, pages 33–36, Portorož, Slovenia.

1.5 Open Source

We released a number of corpora and code repositories on our work, which are summarized as follows:

1. **Chinese-English Dialogue Corpus**

Tvsub[3]: More than two million sentence pairs extracted from the subtitles of television episodes.

MVsub[4]: About one million sentence pairs extracted from the movie subtitles.

2. **Document-Level NMT Codes**

   LC-NMT[5]: our proposed cross-sentence context-aware NMT model built on the top of Nematus.[6]

   CSNMT[7]: our proposed cross-sentence context-aware NMT model re-implemented on the top of ZPTU-NMT.[8]

3. **Reconstruction NMT Codes (Joint Work)**

   NMT-Coverage[9]: the reconstruction model re-implemented on the top of ZPTU-NMT.

[3]Available at: `https://github.com/longyuewangdcu/tvsub`.
[4]Available at: `https://www.computing.dcu.ie/~lwang/corpora/resource.html`.
[5]Available at: `https://github.com/tuzhaopeng/LC-NMT`.
[6]Available at: `https://github.com/longyuewangdcu/nematus`.
[7]Available at: `https://github.com/longyuewangdcu/Cross-Sentence-NMT`.
[8]Available at: `https://github.com/tuzhaopeng/NMT`.
[9]Available at: `https://github.com/tuzhaopeng/NMT-Coverage`.

Chapter 2

Machine Translation and Discourse

MT is an active research field in NLP and many models and algorithms have been intensively studied in the literature. Since 1954, MT has been developed through RBMT (Nirenburg et al. 1986), SMT (Koehn 2009) and NMT (Kalchbrenner and Blunsom 2013, Sutskever et al. 2014) models. In this chapter, we review the background on both traditional SMT as well as the state-of-the-art NMT for two main reasons:

- Discourse for SMT has been previously discussed while discourse-aware NMT received relatively little attention from the research community. Following prior studies, we first explored a hybrid approach *i.e.,* neural component for SMT model (as discussed in Chapter 4). For better understanding related work and our proposed approach, we thus provide basic information on SMT;

- Recently, it is reported that the performance of NMT has surpassed the performance of SMT on various language pairs (Luong et al. 2015a). In the thesis, we mainly investigate discourse-aware approaches for NMT models (as discussed in Chapter 3, 5 and 6). Thus, we also introduce related models of NMT.

Furthermore, discourse is a concept from linguistics while our work is related to Computational Linguistics (CL). We do not try to exhaustively cover all aspects, but mainly introduce "discourse" from the respective of CL. In other words, we focus on discourse

architecture and phenomena which are related to NLP and MT.

This chapter is organized as follows: we first introduce the MT (NMT and SMT) including the models, frameworks, and evaluation metrics in Section 2.1. We then provide the basic information on discourse in Section 2.2, including related theory, structures, linguistic phenomena. In Section 2.3, we present related work on discourse-aware MT including document-level MT and pronominal anaphora in MT. Finally, we summarize the content of this chapter in Section 2.4.

2.1 Machine Translation

MT is a sequence-to-sequence prediction task, which aims to find for a source language sentence the most probable target language sentence that shares the most similar meaning. In following sections, we first introduce the framework of SMT and its components such as translation model and language model. We then describe basic knowledge on word vector models, RNN and neural language models for NMT. Finally, we review the framework of NMT and its related mechanisms.

2.1.1 Statistical Machine Translation

SMT is a data-driven approach towards MT that aims to frame translation as a statistical optimization problem (Koehn 2009, Koehn et al. 2007). The training of an SMT system is a data-driven process, where large amounts of training data are required in order to sufficiently cover the linguistic phenomena for the desired language pair. The training data requires to be parallel, where each sentence in the target language is the translation of the corresponding sentence in the source language.

Assume that a sentence pair $\mathbf{x} = \{x_1, ..., x_i, ..., x_I\}$ and $\mathbf{y} = \{y_1, ..., y_j, ..., y_J\}$ are in source and target side, respectively. x_i is the i-th word of $\mathbf{x}$ and y_j is the j-th word of $\mathbf{y}$. I and J are lengths of $\mathbf{x}$ and $\mathbf{y}$, which can be different. Based on Bayes decision theory, we

can formulate SMT as in Equation (2.1) (Brown et al. 1993):

$$\begin{aligned}\hat{\mathbf{y}} &= \arg\max_{\mathbf{y}} p(\mathbf{y}|\mathbf{x}) \\ &= \arg\max_{\mathbf{y}} \frac{p(\mathbf{x}|\mathbf{y})p(\mathbf{y})}{p(\mathbf{x})} \\ &\propto \arg\max_{\mathbf{y}} p(\mathbf{x}|\mathbf{y})p(\mathbf{y})\end{aligned} \tag{2.1}$$

where $\hat{\mathbf{y}}$ denotes the translation output with the highest translation probability. The translation problem is factored into $p(\mathbf{x}|\mathbf{y})$ and $p(\mathbf{y})$, which respectively represent the inverse translation probability and language model probability. The denominator $p(\mathbf{x})$ is ignored since it remains constant for a given source sentence $\mathbf{x}$. The advantage of this decomposition is that we can learn separate probabilities in order to compute $\hat{\mathbf{y}}$.

One important theoretical development was the log-linear model proposed by Och and Ney (2002), which incorporated different features containing information from the source and target sentences in the model, in addition to the language and translation models of the original noisy channel (Weaver 1949) approach. Thus, $p(\mathbf{y}|\mathbf{x})$ can be decomposed using the log-linear model as presented in Equation (2.2):

$$p(\mathbf{y}|\mathbf{x}) = \sum_{i=1}^{I} \exp \lambda_i \cdot h_i(\mathbf{x}, \mathbf{y}) \tag{2.2}$$

where $h_i(\cdot)$ indicates a translation feature and λ_i is its corresponding optimal weight, which is learned by maximizing the translation probability of a development set. I indicates the total feature number, which can be increased with the advantages of the log-linear framework.

Framework A refinement of word-based models (Brown et al. 1993) is the influential PBSMT model (Koehn et al. 2003), in which a model learns to translate not word-by-word but on the basis of contiguous sets of words, *i.e.,* phrases, which are not necessarily linguistically motivated.

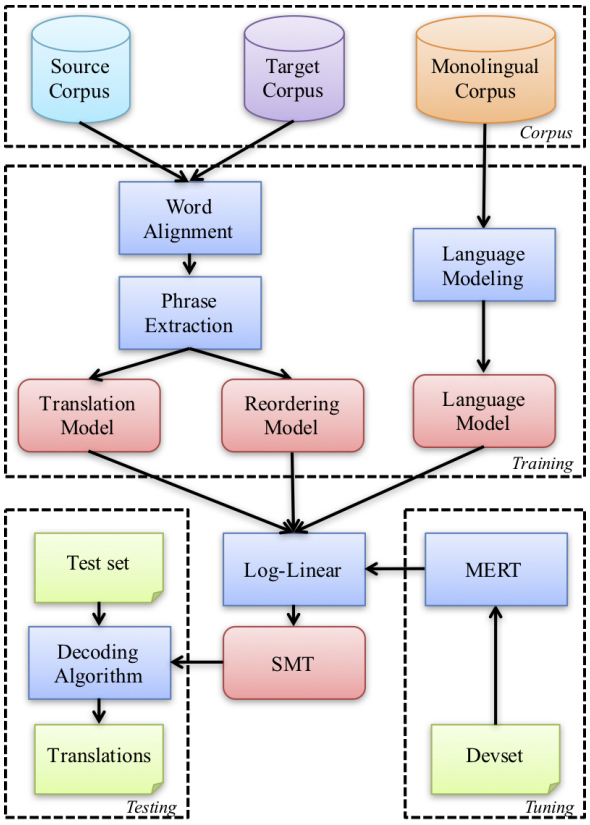


Figure 2.1: Architectures of phrase-based statistical machine translation. Training, tuning and testing phases are illustrated in one framework.

Figure 2.1 describes the architectures of PBSMT, which of several components: 1) words within the parallel corpus are aligned and phrase pairs are then extracted based on word-alignment results; 2) the translation model and the lexicalized reordering model can be learned using aligned phrases; 3) an N-gram language model can be built using a large amount of monolingual sentences in the target language; 4) These models are optimized under the log-linear framework in order to maximize the performance using a development set; 5) with the optimized weight parameters of the features in the models, we can finally translate the test set and the evaluation score indicates the performance of the whole system.

Translation Model A translation model is used to translate text from a source language to a target language. It provides translation segments for input sentences and consists of a direct/inverse phrase translation probability and a direct/inverse lexical weight. The bilin-

gual phrase pairs are extracted based on word alignments (Och and Ney 2003, 2004). Given a collection of phrase pairs, the direct phrase translation probability can be estimated as in Equation (2.3):

$$p(\bar{x}|\bar{y}) = \frac{count(\bar{x}, \bar{y})}{\sum_{\bar{x}'} count(\bar{x}', \bar{y})} \tag{2.3}$$

where $\bar{x}$ and $\bar{y}$ are the source and target phrase pairs, respectively. $count(\cdot)$ is relative frequency. To overcome problems of unreliable probability estimations due to low-frequency phrase pairs, the lexical translation probability is introduced and is computed as in Equation (2.4) (Koehn et al. 2003):

$$p(\bar{x}|\bar{y}, a) = \prod_{m=0}^{M'} \frac{1}{\{n|(m,n) \in a\}} \sum_{\forall(m,n)\in a} w(x_m|y_n) \tag{2.4}$$

where a is the word alignment; M' is the length of $\bar{x}$. m and n indicate a word position in $\bar{x}$ and $\bar{y}$, respectively. $w(x_m|y_n)$ is the lexical weights. The inverse phrase translation probability and lexical translation probability can also be computed accordingly.

Reordering Model A reordering model learns a distribution of location changes for each word of translation from alignment. Taking the lexicalized reordering model (Koehn et al. 2005, Galley and Manning 2008) for example, it estimates three types of orientations (o) – monotone, swap and discontinuous – of a phrase pair based on previous adjacent target phrase. The *monotone* (m) predicts if the current source phrase is located immediately to the right of the previous source; the *swap* (s) predicts if the current source phrase is located immediately to the left of the previous source and the *discontinuous* (d) predicts if the current source phrase is located anywhere else. The reordering probabilities are computed as shown in Equation (2.5):

$$p(o|\bar{x}, \bar{y}) = \frac{count(o, \bar{x}, \bar{y})}{\sum_{o'} count(o', \bar{x}, \bar{y})} \tag{2.5}$$

where we count how often each extracted phrase pair ($\bar{x}$ and $\bar{y}$) is found with each of the three orientation types $o \in \{m, s, d\}$.

N-gram Language Model A language model is used to evaluate the fluency for the translations with respect to target language. It estimates the likelihood of a word appearing next in a sequence of target words. The probability of $\mathbf{y}$ can be computed according to the chain rule as in Equation (2.6):

$$\begin{aligned} p(\mathbf{y}) &= p(y_1, y_2, \ldots, y_{N-1}, y_N) \\ &= p(y_1) \times p(y_2|y_1) \times \cdots \times p(y_N|y_1, y_2, ..., y_{N-1}) \end{aligned} \tag{2.6}$$

Due to high computation costs and sparsity for longer sentences, we only consider a limited number of historical words according to the Markov assumption (Stolcke 2002). For example, a bigram LM can be computed as $p(y_1) \times p(y_2|y_1) \times ... \times p(y_n|y_{n-1}) \times ... \times p(y_N|y_{N-1})$, where $p(y_n|y_{n-1})$ is bigram relative frequency as in Equation (2.7):

$$p(y_n|y_{n-1}) = \frac{count(y_{n-1}, y_n)}{\sum_{y'} count(y_{n-1}, y')} \tag{2.7}$$

Decoding Finding the best translation for a sentence which corresponds to a search problem is called decoding and is an NP-complete problem, possibly exponential in the length of the sentence to be translated (Knight 1999). Normally, one or more heuristics are used in order to make decoding computationally feasible, such as beam-search (Och and Ney 2004). The decoder generates hypotheses (translation segments) from left to right using the beam search algorithm. Each hypothesis maintains a coverage vector to indicate which source words have been translated so far. The translation process ends when all source words have been translated.

2.1.2 Neural Machine Translation

As presented in Section 2.1.1, PBSMT consists of a translation model, a reordering model and an Language Model (LM), which are linearly integrated using the log-linear framework. NMT, being a new paradigm for MT, employs an individual large NN to model the entire translation process. The advantages of NMT over SMT can be summarized as follows:

1. Distributed word representations can facilitate the computation of semantic distance (Bengio et al. 2003);

2. Different from SMT, there is no need to explicitly design features to capture translation regularities (Tu et al. 2016);

3. RNN are better at capturing long-distance reordering, which is a significant challenge for SMT (Zhang et al. 2015).

In the following parts, we review the knowledge related to NMT, including word embedding, RNN, RNN LM, encoder-decoder framework, attention mechanism, and bidirectional RNN mechanism.

Word Vector Model It is distributed representations of words, which are important building blocks in NN. In NLP tasks, words are treated as discrete atomic symbols. For instance, "cat" may be represented as "$id0537$" and "dog" as "$id0143$", which are arbitrary, and provide no useful information to the system regarding the relationships that may exist between the individual symbols. Representing words as unique, discrete IDs furthermore leads to data sparsity. Mikolov et al. (2013b) shows that distributed word representations can capture the linguistic regularities and similarities in the training corpus. In NLP, distributed word representations have the advantage that similar words are represented closely in the vector space. For example, given the word vectors of words "king", "man" and "woman", we can apply vector operations on them: $vec(king) - vec(man) + vec(woman) = vec(queen)$.

There are different ways to represent words and one prominent approach uses vectors as their representations. A word vector model is a $V \times E$ matrix which can map a word in a

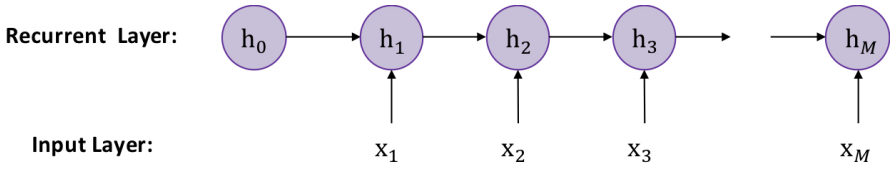


Figure 2.2: A simple unfold RNN which maintains a context vector covering previous sequential information. For example, h_1 is computed using x_1 and h_0. Later, h_1 is involved in the computation of h_2. h_0 is the initial state (a vector of zeros or random numbers) of the network. The context vector is also referred as the hidden state of an RNN and x_t is the input at time step t.

vocabulary to a real-value word vector, where V is the size of vocabulary and E is the size of word vectors. Besides, word vector models can be trained together with the other tasks (*e.g.,* RNN language modeling) and word embeddings can be updated during the training process.

Recurrent Neural Networks RNN are neural networks on sequential inputs and assume that the hidden states within the network are dependent, which is true in many sequence prediction tasks. The hidden states can be thought of a "memory" to maintain the previous history.

A simple RNN, as seen in Figure 2.2, consists of two layers: an input layer and a recurrent layer. The recurrent layer maintains a context vector (hidden state) covering previous sequential information. Each h_t in an RNN is computed by the input x_t at time step t and previous hidden state h_{t-1}. Formally, given the history representation h_{t-1} encoding all the preceding words and the input x_t at time step t, each hidden state h_t in a RNN is computed as in Equation (2.8):

$$\mathbf{h}_t = f(W\mathbf{x}_t + U\mathbf{h}_{t-1} + b) \tag{2.8}$$

where $f(\cdot)$ is an active function such as sigmoid, tanh etc. b is a bias value. W is the weight matrix between the input and the hidden state. U is the weight matrix between the context and the hidden state. W and U can be obtained using Back Propagation Through Time (Rumelhart et al. 1995).

However, simple RNNs suffer from the vanishing gradient problem (Bengio et al. 1994),

where for long sequence inputs, the early contexts are often forgotten and overwritten by the later contexts. The Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber 1997) or the more recent Gated Recurrent Unit (GRU) (Chung et al. 2014) use gates to control the information flow from previous words, which are better at capturing long-term dependencies. GRU is a simplified variation of LSTM with fewer gates but comparable performance. As illustrated in Figure 2.3, GRU consists of an update gate and a reset gate, as in Equation (2.9):

$$\begin{aligned} u_t &= f(W_u x_t + U_u h_{t-1} + b_u) \\ r_t &= f(W_r x_r + U_r h_{t-1} + b_r) \\ \tilde{h}_t &= g(W x_t + U(r_t \odot h_{t-1}) + b) \\ h_t &= u_t \odot h_{t-1} + (1 - u_t) \odot \tilde{h}_t \end{aligned} \tag{2.9}$$

where $f(\cdot)$ is a sigmoid function and $g(\cdot)$ is a tanh function. u_t is the update gate and r_t is the reset gate. $\tilde{h}_t$ is the candidate activation and $\odot$ is the element-wise multiplication operation. h_t is the linear-interpolated output between the previous hidden state h_{t-1} and the candidate activation. Intuitively, the update gate determines the interpolation weights between the previous hidden state h_{t-1} and the candidate activation, and the reset gate determines the information flow from previous hidden states. If the reset gate is set to 1 and the update gate is set to 0, the GRU is equivalent to the simple RNN. W_u, U_u, W_r, U_r, W and U are the weight parameters, and b_u, b_r and b are the bias values of the corresponding gates.

RNN Language Model Recurrent Neural Network Language Model (RNNLM) models the probability of the next word given the previous words. It is known to be better at generalization as word embeddings are used in training. The simplest RNNLM has an input layer, a recurrent layer, and an output layer, as seen in Figure 2.4. Compared with RNN (as in Figure 2.2), RNNLM additionally has an output layer, which operates a softmax function to compute probability distributions over all words in the vocabulary. If the recurrent layer

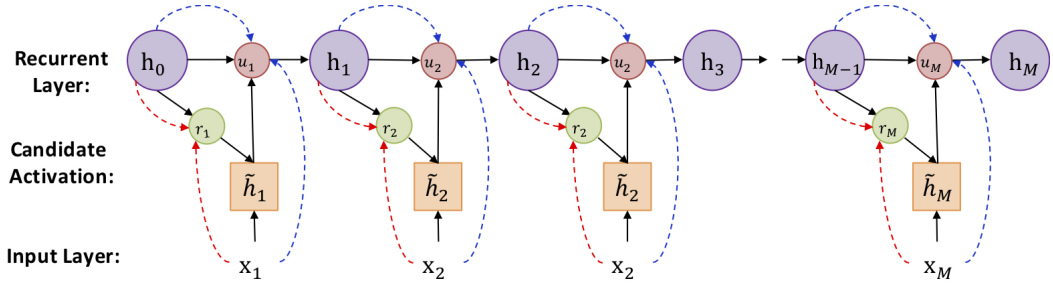


Figure 2.3: Illustration of a GRU network, which consists of an update gate u and a reset gate r. Dashed lines indicate the computations for u and r and h_0 is the initial state (a vector of zeros) of the network.

is a GRU, we can formally define the RNNLM based on Equation (2.9) (Chung et al. 2014):

$$p(t) = softmax(S(h_t)) \tag{2.10}$$

where $S(\cdot)$ is a transform function which can convert h_t into a vector with dimensions equal to the size of vocabulary. Words are sequentially fed to the model and each word is assigned a probability to indicate the likelihood of being the next word. At each training step, we use cross-entropy to compute the error vectors, model weights are updated with BPTT. For example, we can define the cross-entropy error function as in Equation (2.11):

$$C(y, \hat{y}) = -\sum_i y_i \log(\hat{y}_i) \tag{2.11}$$

where y is the predicted probability distribution and $\hat{y}$ is the true distribution. The RNN is unfolded into a flat architecture through time for a certain amount of time-steps and the errors are summed up for all unfolded time-steps. Then gradients of the error are computed and model parameters are updated.

Encoder-Decoder Framework The encoder-decoder architecture is widely employed, in which the encoder summarizes the source sentence into a vector representation, and the decoder generates the target sentence word by word from the vector representation. The encoder can be implemented using an RNN model and the decoder can be implemented

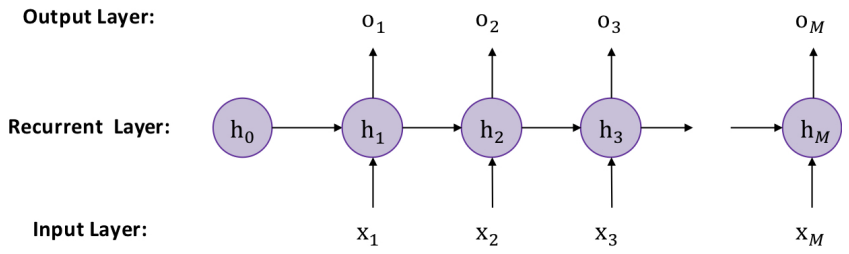


Figure 2.4: Illustration of RNNLM, which has an input layer, a recurrent layer, and an output layer. The recurrent layer uses a GRU network. h_0 is the initial state (a vector of zeros) of the network. For example, if the current input word is w_1, we first learn the word vector x_1 in the input layer, then compute the context vector h_1 in recurrent layer using the GRU network. In the output layer, we compute the probability of the current output o_1 using a softmax function..

using an RNNLM model, thus the framework can be regarded as being composed of an RNN and an RNNLM. Formally, the NMT directly models the probability of translation from the source sentence to the target sentence word by word as in Equation (2.12):

$$P(\mathbf{y}|\mathbf{x}) = \prod_{j=0}^{N} P(y_j|y_{<j}, \mathbf{x}) \tag{2.12}$$

in which given source sentence $\mathbf{x}$ and previous target translations $y_{<j}$ $(y_1, ..., y_{j-1})$, we need to compute the probability of the next word y_j $(j \in \{1, ..., N\})$. This can be interpreted as the translation probability of a target sentence $\mathbf{y}$ given a source sentence $\mathbf{x}$ is computed by multiplying the translation probabilities of each target word; and the translation probability of each target word, *e.g.,* y_j, is computed as the conditional probability of given source sentence $\mathbf{x}$ and previous target translations $y_{<j}$.

Firstly, x_i is represented as a 1-of-K vector $w_i \in R^{|V|}$, where the dimension of the vector $|V|$ equals the size of the vocabulary. The vector consists of 0s in all cells with the exception of a single 1 in a cell used uniquely to identify the word. We then map each w_i to a low dimension semantic space using word embedding s_i. The source sentence is encoded into a sequence of hidden states $h = h_1, ..., h_N$, in which h_j is the hidden state of the i-th source word vector s_i and the last hidden state h_N $(c \equiv h_N)$ is the representation of the whole sentence. The decoder internal hidden state z_j is computed based on source sentence

c, previously generated word u_{j-1} and previous hidden state z_{j-1} as in Equation (2.13):

$$z_j = f(u_{j-1}, z_{j-1}, c) \tag{2.13}$$

where $f(\cdot)$ is a function to compute the current decoding state given all the related inputs. It can be either a vanilla RNN unite using tanh function, or a sophisticated gated RNN unit such as GRU or LSTM. Given the source context c, current decoder hidden state z_j and previously generated words y_{j-1}, the probability of generating next word y_j is computed as in Equation 2.1:

$$p(y_j|y_{<j}, \mathbf{x}) = softmax(g(u_{j-1}, z_j, c)) \tag{2.14}$$

where $g(\cdot)$ is a non-linear function that can transform the inputs into a vector. The decoder uses the softmax function to output the probability distribution over the target words, which can be used to select a word u_j by sampling the distribution.

All the network parameters are trained to maximize the probability in the bilingual training data. The NMT model can be trained with the mini-batch Stochastic Gradient Descent algorithm (Robbins and Monro 1985) together with Adadelta (Zeiler 2012) and is validated based on cross-entropy error.

Bidirectional RNN In the encoder, words can also be fed into RNNs in both directions, using a bidirectional RNN (Schuster and Paliwal 1997). As RNNs can represent recent inputs better, the hidden states of a bidirectional RNN are representing the context word in both sides. A bidirectional RNN used in NMT "contains the summaries of both the preceding words and the following words" (Bahdanau et al. 2015) for source inputs. Sutskever et al. (2014) also claim that it is "extremely valuable" and can "greatly boost the performance" by using the reversed source sentences information NMT.

Figure 2.5 is a graphical illustration of the bidirectional RNN, which consists of a forward ($\overrightarrow{h_1}$, ..., $\overrightarrow{h_M}$) and a backward ($\overleftarrow{h_1}$, ..., $\overleftarrow{h_M}$) RNNs, where $(f_1, \ldots . f_M)$ are the input

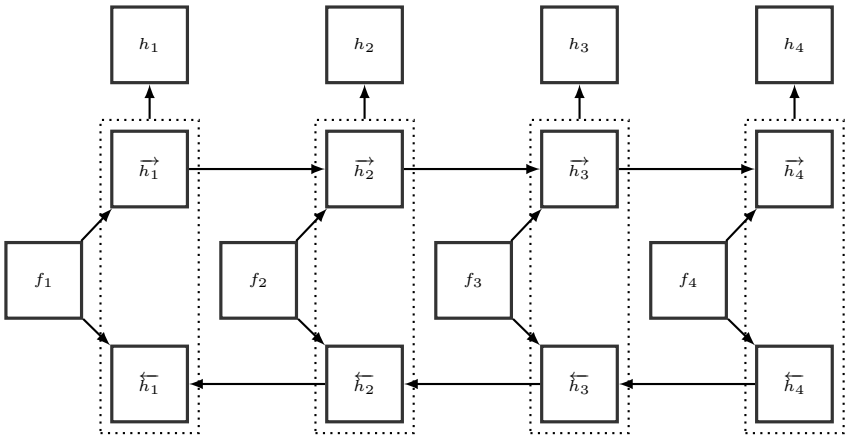


Figure 2.5: The graphical illustration of the bidirectional RNN, which consists of forward a forward ($\overrightarrow{h_1}$, ..., $\overrightarrow{h_M}$) and a backward ($\overleftarrow{h_1}$, ..., $\overleftarrow{h_M}$) RNNs, where $(f_1, \ldots . f_M)$ are the input sequences. The outputs are the concatenations of the annotation vector at a corresponding time step, i.e. $h_i = [\overrightarrow{h_i}, \overleftarrow{h_i}]$.

sequences. The outputs are the concatenations of the annotation vector at a corresponding time step, i.e. $h_i = [\overrightarrow{h_i}, \overleftarrow{h_i}]$.

Attention Mechanism The original encoder-decoder framework uses a fixed-size vector to represent the whole source input. The attention mechanism is proposed by Bahdanau et al. (2015) to learn dynamic soft-alignment during network training. Although an RNN is known to be better at capturing long-range dependencies, Bahdanau et al. (2015) reported that translation quality decreases for long input sentences. With the attention model, source information can be spread across the source context vector, and the decoder can selectively pay attention to different parts of the source context during decoding.

Figure 2.6 illustrates this mechanism. The attention model computes weights ($\alpha_{1,j}, \alpha_{2,j}$ $, \ldots, \alpha_{M,j}$) for each source context h_i and outputs a weighted sum of h_i – a distinct source context vector, c_j. Note that the original encoder-decoder NMT regards the source context vector c as a static vector that summarizes the whole sentence (*i.e.,* $c \equiv h_N$ as shown in Equation 2.14), while an attention-based NMT regards context as a dynamic vector that selectively summarizes certain parts of the source sentence at each decoding step. The alignment model that scores the alignment at position i and j in $\mathbf{x}$ and $\mathbf{y}$ respectively, is

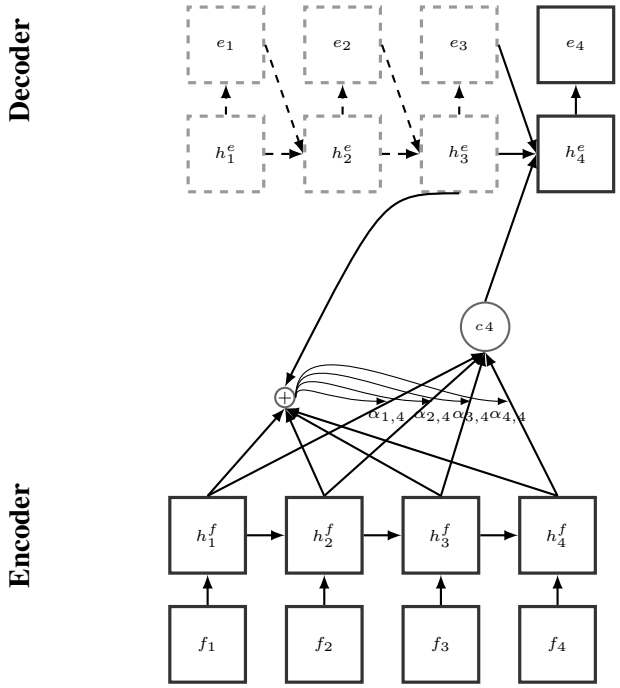


Figure 2.6: The graphical illustration of the attention-based NMT, where h^f indicates the source context vectors and h^e indicates the target context vectors. Suppose there are 4 source input words $\{f_1, \ldots, f_4\}$ and the current predicting word is e_4 in the target. The encoder reads the source input words and produces the source context vectors for each source input word. Next, the attention model $\oplus$ computes weights ($\alpha_{1,4}$, $\alpha_{2,4}$, $\alpha_{3,4}$ and $\alpha_{4,4}$) for each h^f and outputs a weighted sum of h^f – a distinct source context vector c_4. Then, the distinct source context vector c_4, previous translation e_3 and previous target context vector h_3^e are used to obtaining the current target context vector h_4^e, which is used to output translation probability for all target words.

computed as in Equation (2.15):

$$e_{ij} = v^T a(z_{j-1}, h_i) \tag{2.15}$$

where z_{j-1} is the target hidden state and h_i is the source hidden state at time i, and a is a non-linear function, such as the $tanh$ function. $v \in \mathbb{R}^n$ is a weight matrix. Thus, a distinct source context vector c_j can be computed for each word in the target side, and the source context vector c is rewritten as in Equation (2.16):

$$c_j = \sum_{i=1}^{M} \alpha_{ij} h_j \tag{2.16}$$

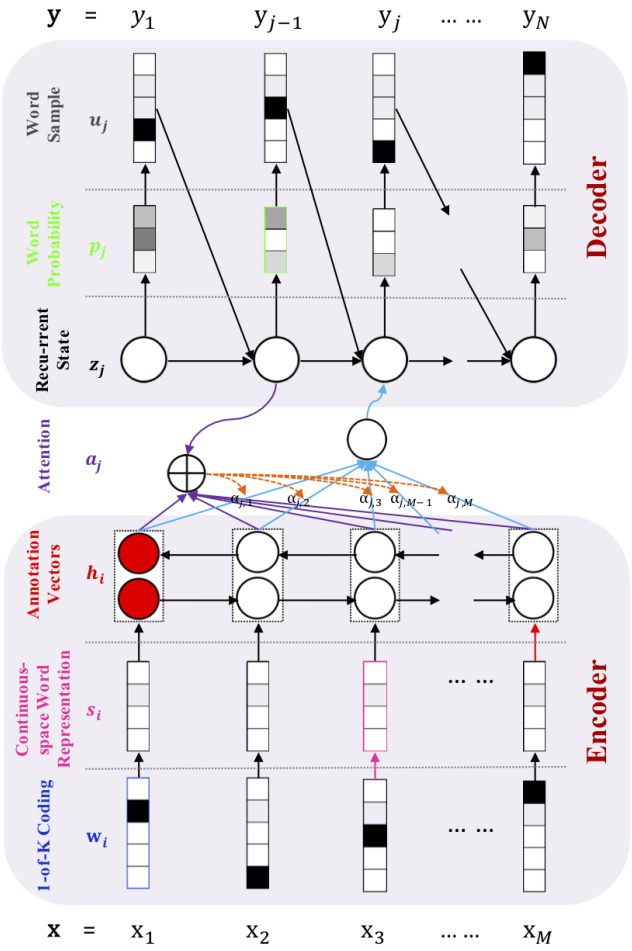


Figure 2.7: Architectures of NMT equipped with bidirectional RNN and attention mechanism.

where $\alpha_{i,j}$ is normalized weight for each context vector of source input in $\{0, \ldots, i\}$, computed as in Equation (2.17):

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{M} \exp(e_{ij})} \tag{2.17}$$

Thus, the c in Equation (2.13) and (2.14) is updated accordingly.

Finally, Figure 2.7 shows the overall framework of NMT equipped with bidirectional RNN and attention mechanism. We employ this model as a strong baseline in our thesis.

2.1.3 Machine Translation Evaluation

To evaluate MT translation quality, we use automatic evaluation metrics. Compared to human evaluation, automatic evaluation metrics are faster and more consistent. Many automatic evaluation metrics have been proposed in the field, e.g. Sentence Error Rate (SER), Word Error Rate (WER) (Stolcke et al. 1997), Bilingual Evaluation Understudy (BLEU) (Papineni et al. 2002), METEOR (Banerjee and Lavie 2005) and Translation Edit Rate (TER) (Snover et al. 2006). In this thesis, we choose to use BLEU to estimate the machine translation quality as it is the most commonly used one in MT.

BLEU is a reference-based MT evaluation metric, so reference translations are essential when computing the evaluation scores. It is language-independent. The output of BLEU is a score between 0 and 100% indicating the similarity between the MT outputs and the reference translations. BLEU is computed over the entire test set. The higher the scores are, the better the translations are. BLEU scores are computed based on a modified *n*-gram precision, as in Equation (2.18):

$$BLEU = BP * \exp \sum_{n=1}^{N} \frac{1}{N} \log(\frac{|m_n \cap m_r|}{|m_n|}) \tag{2.18}$$

where n represents the order of the *n*-grams compared between the translations and references. Typically, n is from 1 to 4. m_n and m_r indicate the *n*-grams occurring in the MT outputs and the corresponding references, respectively. $|m_n \cap m_r|$ is the number of *n*-grams occurring in both translations and references. In the case of multiple occurrences *n*-grams, we clip $|m_n \cap m_r|$ to the maximum number of times that an *n*-gram occurs in the reference. The motivation is that MT systems can overgenerate improbable translations and "a reference word should be considered exhausted after a matching candidate word is identified" (Papineni et al. 2002). A high BLEU score candidate translation should also match the reference translations in length, therefore, BP is introduced. BP is the brevity penalty to

penalize shorter translations than the references, which is computed as in Equation (2.19):

$$BP = \exp^{max(1-\frac{length(r)}{length(n)},0)} \tag{2.19}$$

where n and r indicate the translation output and reference translation, respectively.

2.2 Discourse

Natural languages, from bottom to top, can be divided into several linguistic units including word, phrase, clause, sentence, paragraph, and discourse (Longacre 2013). A discourse is an instance of language use whose type can be classified on the basis of such factors as grammatical and lexical choices and their distribution in main versus supportive materials, theme, style, and the framework of knowledge and expectations within which the addressee interprets the discourse (Elson and Pickett 1983, Crystal 1985, Hanks 1987, Longacre 1990). In this subsection, we mainly introduce the knowledge of discourse from the perspective of NLP instead of linguistic theory.

As shown in Table 2.1, we use examples to illustrate "what is discourse". Example 1 is a paragraph, which consists of four complete sentences. Although the text is grammatically correct, it is not a discourse. Because each sentence is independent and they bear no relation to each other. In contrast, the dialogue in Example 2 is a discourse. First, Speaker A makes a request for Speaker B to perform an action (*i.e.,* answering the phone). Speaker B then states a reason why he/she cannot comply with the request. Finally, Speaker A undertakes to perform the action. Although some information is implicit, these utterances are closely related to each other under a clear topic: who can answer the phone. From the example, we can make a summary on discourse and its properties: 1) it is a continuous stretch of language longer than a sentence; 2) it involves conversation (*e.g.,* dialogue) or text (*e.g.,* document); 3) it is meaningful, coherent, unified and purposive.

A discourse contains seven fundamental properties including cohesion, coherence, intentionality, acceptability, informatively, situationality and intertextuality (De Beaugrande

No.	Example
1	It is very hot today. Cohen comes from Germany. HK launches first sightseeing restaurant bus to promote tourism. Natural language processing has been rapidly developed in recent years.
2	A: That's the telephone. B: I'm in the bath. A: O.K.

Table 2.1: An examples of discourse.

and Dressler 1981). Among them, *cohesion* and *coherence* have often been studied together in discourse analysis (Sanders and Maat 2006, Xiong et al. 2013). Besides, translation *consistency* is an important issue in document-level translation (Xiao et al. 2011). Therefore, in the following contents, we mainly introduce these three properties from the perspectives of MT and NLP.

2.2.1 Cohesion

From a linguistic perspective, cohesion is a well-known means to establish such inter-sentential links within a text. Widdowson (1979) defines cohesion as "the overt structural link between sentences as formal items". Cohesion is a surface property of the text that is realized by explicit clues. It occurs whenever "the interpretation of some element in the discourse is dependent on that of another" (Halliday and Hasan 1976). In other words, cohesion refers to various manifest linguistic links (*e.g.,* references, word repetitions) between sentences within a text that holds the text together.

Halliday and Hasan (1976) identify five general categories of cohesive devices: reference, ellipsis, substitution, lexical cohesion, and conjunction. We mainly introduce referential cohesion and lexical cohesion in the following contents. Besides, referential cohesion is mainly realized by the way of pronominal reference including *anaphora* and *coreference*, as shown in Figure 2.8.

To derive the correct interpretation of a text, or even to estimate the relative importance of various mentioned subjects, pronouns and other referring expressions must be connected

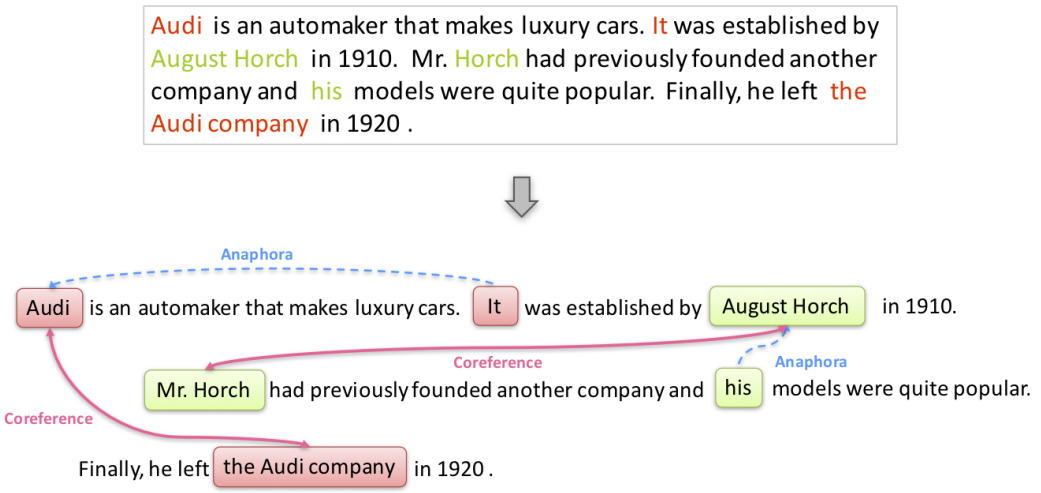


Figure 2.8: An example of referential cohesion.

to the right individuals. In NLP community, researchers investigated various referential cohesion tasks such as anaphora resolution (Yang et al. 2006), coreference resolution (Kong and Zhou 2012, Chen and Ng 2012) and dropped pronoun recovering (Chen and Ng 2013, Xue and Yang 2013) are all well-studied problems of referential cohesion. Furthermore, lexical cohesion knowledge are explored for text summarization (Barzilay and Elhadad 1997), word sense disambiguation (Galley and McKeown 2003), question answering (Novischi and Moldovan 2006) etc.

Anaphora Anaphora is the use of an expression whose interpretation depends specifically upon antecedent expression. The anaphoric (referring) term is called an anaphor. Sometimes anaphor may rely on the postcedent expression, and this phenomenon is called cataphora. Taking Figure 2.9 for example, the pronoun "It" is an anaphor, which points to the left toward its antecedent "Audi". When translating the English sentence into French, the pronoun "it" could be translated into three equivalents according to the properties of its antecedent: 1) "il" (masculine singular subject pronoun); 2) "elle" (feminine singular subject pronoun); 3) "cela" (demonstrative pronoun). It is easy for the human to choose the correct translation, however, sentence-level MT models always make mistakes.

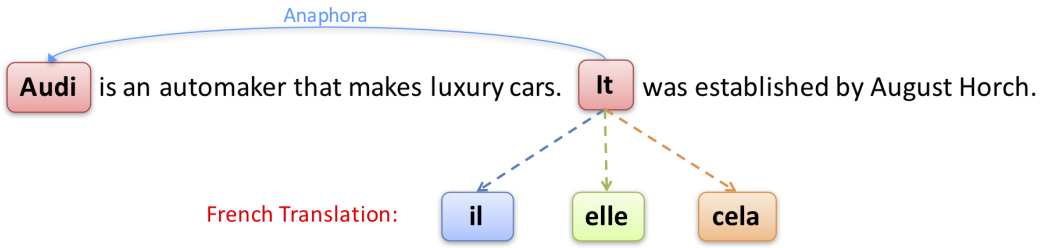


Figure 2.9: An example of anaphora and translation problem. When translating the English sentence into French, the pronoun "it" could be translated into three equivalents according to the properties of its antecedent.

Zero Anaphora (pronoun-dropping) is a more complex case of anaphora. In pro-drop languages such as Chinese, pronouns can be omitted to make the sentence compact yet comprehensible when the identity of the pronouns can be inferred from the context. These omissions may not be problems for our humans since we can easily recall the missing pronouns from the context. However, this poses difficulties for MT from pro-drop languages to non-pro-drop languages (*e.g.,* English), since the translation of such missing pronouns cannot be normally reproduced. Taking Figure 2.10 as an example, all the pronouns (in purple blocks) are omitted in the conversation between Speaker A and B, however, speakers can still recall the missing pronouns from the context. As seen, the omitted object pronouns "它" (it) refers to the noun "工作" (job) while the others pronouns refer to the speaker themselves. When translating the Chinese sentence into English, humans can easily recover these DPs and then translate the "complete" sentence. However, the sentence-level MT models make two severe mistakes: 1) harming the syntax structure (*e.g.,* interrogative sentence); and 2) missing translations of corresponding elements (*e.g.,* subject-verb-object).

Coreference Two or more expressions (*e.g.,* nouns) in a text refer to the same referent. As the referents point to persons or things in the real world, the coreference relation can exist independently of the context. As shown in Figure 2.11, the noun phrases "HK Chief Executive" and "Mr. Tung Chee-hwa" point to the same person, although their surfaces are totally different. It may not result in severe problems for MT. As seen, even though MT is not aware of the coreference relation, it still can translate them well.

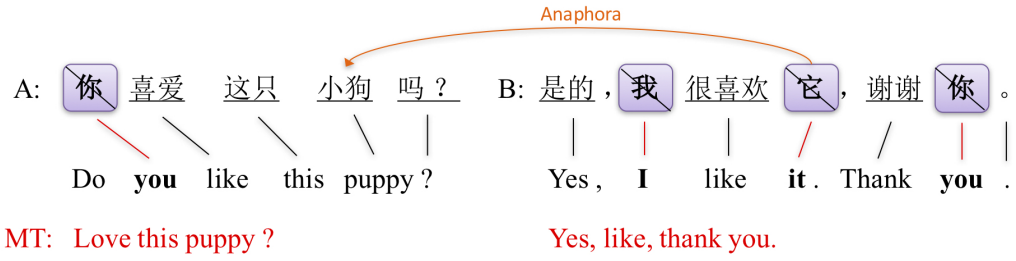


Figure 2.10: An example of zero anaphora and translation problem. The sentence-level MT models make two severe mistakes: 1) harming the syntax structure (*e.g.,* interrogative sentence); and 2) missing translations of corresponding elements (*e.g.,* subject-verb-object).

Figure 2.11: An example of coreference.

Lexical Cohesion Lexical cohesion refers to the way related words are chosen to link elements of a text. I can divided into two forms: repetition and collocation. The "repetition" indicates the linking between the same word, or synonyms, antonyms, etc. As shown in Figure 2.12 (a), the synonyms "dress" and "frock" across two sentences are the repetition case. In the "collocation" form, related words are typically put together or tend to repeat the same meaning. For example, the phrase "once upon a time" in Figure 2.12 (b) is a collocation case. As seen, MT outputs are insufficiently perfect without considering repetition while there are no effects on collocation translation.

2.2.2 Coherence

To make a text semantically meaningful, coherence is related to the connectedness of the "mental representation of the text rather than of the text itself" (Sanders and Maat 2006). Coherence is created referentially, when different parts of a text refer to the same entities, and relationally, by means of coherence relations such as "Cause–Consequence" between different discourse segments. Therefore, *discourse structure* (sequencing subparts of the

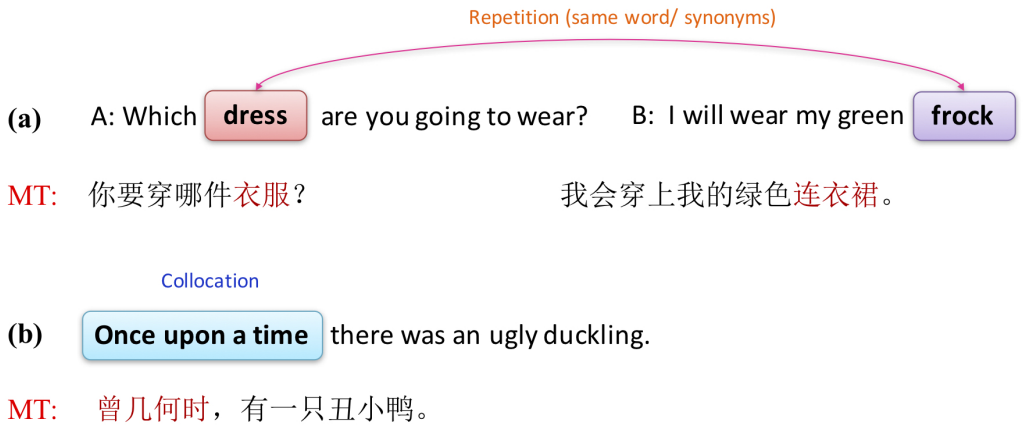


Figure 2.12: An example of lexical cohesion.

discourse as well as relations between them) can be used to analyze the coherence of a text. The commonly-used discourse structures are Rhetorical Structure Theory (RST) (Mann and Thompson 1988) and Penn Discourse Tree Bank (PDTB) (Marcu 2000).

In NLP, discourse parsing is one of the fundamental tasks, which automatically parses a text into the relational structure like RST and PDTB trees. Researchers have explored various approaches for discourse parsing (Soricut and Marcu 2003, Feng and Hirst 2012, Xue et al. 2015, 2016). For instance, Lin et al. (2014) introduce a pipeline framework including several sub-tasks (connective classifier, argument labeler, explicit classifier, and non-explicit classifier) to handle both explicit and non-explicit relations based on the PDTB using maximum entropy.

Rhetorical Structure Theory RST relations are applied recursively in a text until all units in that text are constituents in a predefined relation. As shown in Figure 2.13, the result of such analysis is that RST structure is typically represented as a tree, with one top-level relation that encompasses other relations at lower levels. There are a number of predefined relations such as "ATTRIBUTION" (causality) and "Contrast" (adversative relation), and the leaves are presented as segments/parts of the text.

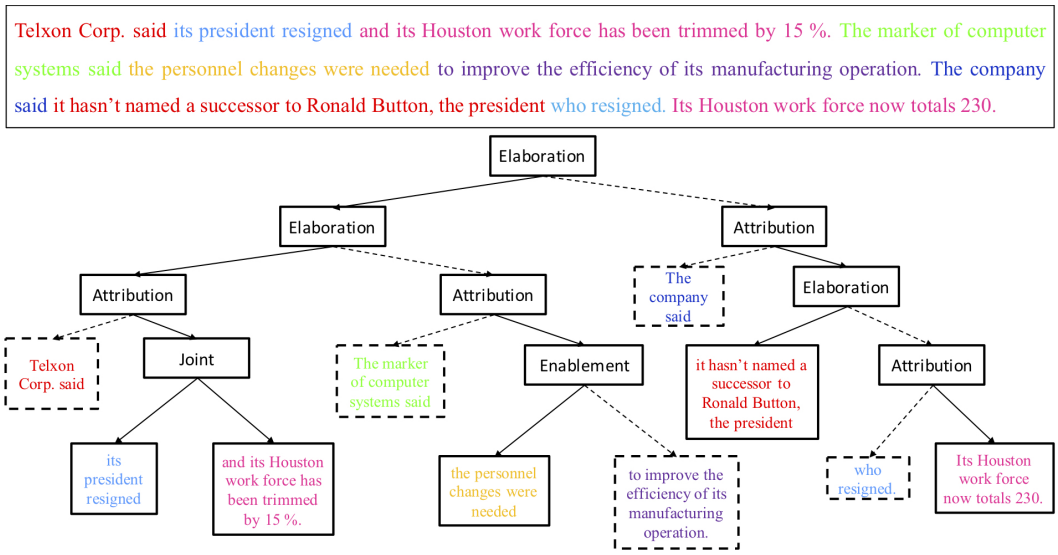


Figure 2.13: An example of RST Tree.

Penn Discourse Tree Bank The PDTB annotation methodology is proposed based on RST, but highlights the role of the connectives. According to whether containing a connective or not, discourse relations can be divided into two categories: explicit and implicit. In Table 2.2, Example 1 shows an explicit discourse, which uses a coordinating conjunction "But" to bridge two text spans (*i.e.,* arguments), and the relationship between them is "Comparison.Concession" (two-level relation category). However, Example 2 omitted discourse connective "however", and the implicit relation between the two arguments is "Comparison.Contrast". The CoNLL Shared Task has organized a series of tasks on discourse parsing based on PDTB, focusing on identifying individual discourse relations that are present in a natural language text.[1]

Cohesion is related to the surface structure link while coherence concerns the underlying connectedness in a text (Vasconcellos 1989). Compared with cohesion, coherence is not easy to be detected. The Chinese sentence in Figure 2.14 is in a "Cause–Consequence" order, in which "他没有上过学 (he did not go to school)" is the "cause" and "只能写成到这种水平 (he can only write to this level)" is the "Consequence" with a connective

[1]The CoNLL-2015 Shared Task: `http://www.cs.brandeis.edu/~clp/conll15st/` and the CoNLL-2016 Shared Task: `http://www.cs.brandeis.edu/~clp/conll16st/`

No.	Example (Argument 1 – Connective – Argument 2)
1	According to Lawrence Eckenfelder, "Kemper is the first firm to make a major statement with program trading." He added that "having just one firm do this isn't going to mean a hill of beans. But if this prompts others to consider the same thing, then it may become much more important."
2	According to Lawrence Eckenfelder, "Kemper is the first firm to make a major statement with program trading." He added that "having just one firm do this is not going to mean a hill of beans. However, if this prompts others to consider the same thing, then it may become much more important."

Table 2.2: An examples of explicit and implicit relations in PDTB.

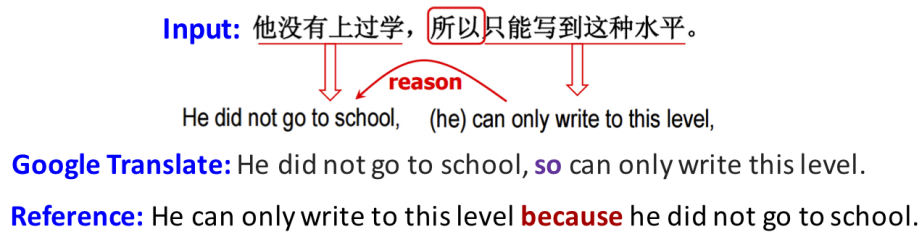


Figure 2.14: An example of coherence and translation problem.

"所以 (so)". In contrast, native English speakers usually express the same meaning in a "Consequence–Cause" order, *i.e.,* "He can only write to this level because he did not go to school.". The MT system is not aware of the coherence property, and literally translate the input into "He did no go to school, so can only write to this level".

2.2.3 Consistency

Apart from cohesion and coherence, consistency is another critical issue in document-level translation, where a repeated term should keep the same translation throughout the whole document (Xiao et al. 2011). The underlying assumption is that the same concepts should be consistently referred to with the same words in a translation. However, the consistency in MT output is generally overlooked in most MT systems due to the lack of the use of document contexts.

As shown in Figure 2.15, the Chinese phrase "大都会警察" is a proper noun being equivalent to "metropolitan police". In the document, this term in the first sentence

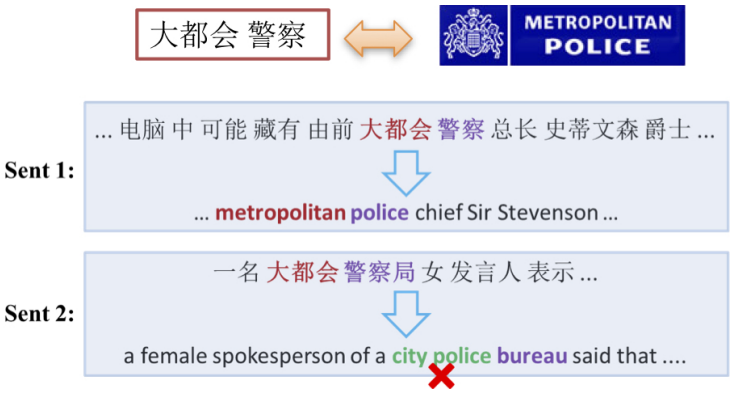


Figure 2.15: An example of consistency and translation problem.

is translated correctly. However, when the term occur again in the second sentence, it is translated into a wrong phrase "city police", leading to inconsistency. To alleviate the inconsistency problems, some researchers investigated different approaches for MT and MT evaluation (Xiao et al. 2011, Guillou 2013, Chen and Zhu 2014).

2.3 Discourse in Machine Translation

There are different strands of research in the literature. One attempts to exploit the macroscopic structure of the input texts to infer better translations. Some work concerns different discourse properties including cohesion, coherence, and consistency. Other work deals with specific linguistic phenomena that are governed by discourse-level processes such as generation of anaphoric pronouns and translation of discourse connectives (Hardmeier 2014). These three strands are not isolated but closely related to each other. For instance, cross-sentence information can not only improve the overall performance of MT but also alleviate inconsistency problems at the same time. In the thesis, we mainly focus on two parts: document-level MT and discourse phenomena for MT. Thus, in this section, we discuss important related work of discourse-aware MT, including document structure as well as discourse phenomena.

As stated at the beginning of this chapter, discourse-aware NMT received relatively little attention from the research community while various discourse-aware approaches have

been investigated for SMT. Thus, we review related work on both conventional SMT and state-of-the-art NMT.

2.3.1 Discourse Structure and Document Structure in Machine Translation

As discussed in Section 2.2.2, coherence is mainly related to discourse or document structure. One attempts to exploit the discourse trees of the input texts to infer better translations. The other strand is to consider the document as a whole to resolve certain ambiguities and inconsistencies.

Discourse Structure Foster et al. (2010) try the first attempt to incorporate structural information into SMT. They tagged each sentence with features such as kind of session, identity of the speaker, time period, and then used domain adaptation methods to balance between an LM trained from similar data and a background LM. Marcu et al. (2000) found that there are significant differences in discourse structure of between Japanese and English. Thus, they propose an "analysis–transfer–translate" pipeline: firstly, Japanese text is parsed into RST tree; and then it is transferred into English style RST tree; finally process translation based on the RST tree. Although all the training data are manually annotated (high cost), the method really improves the translation in term of coherence. Besides, Tu et al. (2013) propose a novel translation framework, which mainly includes three steps: 1) Source RST tree acquisition: a source sentence is parsed into an RST tree; 2) Rule extraction: translation rules are extracted from the source tree and the target string via bilingual word alignment; 3) RST-based translation: the source RST-tree is translated with translation rules. Experiments show that their approach achieves improvements of about +2 BLEU points than the baseline system on Chinese–English.

Because of the superior ability to preserve sequence information over time, LSTM has obtained strong results on a variety of sequence modeling tasks. Sequence models construct sentence representations as an order-sensitive function of the sequence of tokens. In contrast, tree-structured models compose each phrase and sentence representation from

its constituent sub-phrases according to a given syntactic structure over the sentence. Tai et al. (2015) introduce a Tree-LSTM, a generalization of LSTMs to tree-structured network topologies. The difference between Tree-LSTM and LSTM is that the Tree-LSTM composes its state from an input vector and the hidden states of arbitrarily many child units. Thus, the standard LSTM can then be considered a special case of the Tree-LSTM where each internal node has exactly one child. They show its superiority for representing sentence meaning over a sequential LSTM in two tasks: predicting the semantic relatedness of two sentences and sentiment classification.

Although their work show promising improvements, there are several underlying drawbacks: 1) some models are trained on small-scale or manually-created data sets, it is not reliable when adopting these approaches to large-scale MT task; 2) the performance of discourse parser is still not reliable[2], thus incorporating the structure information into NMT will result in error propagation problems.

Document Structure One direction is cache-based methods, which employ cache to retain bilingual phrase pairs from the best hypothesis of previously translated sentences and then use it as an additional feature in log-linear model of SMT. Tiedemann (2010) integrated cache-based language and translation models within a PBSMT decoder and used an exponential decay factor to carry over word preferences from one sentence to the next. When a source phrase is considered for translation, its cache translation score is computed using the phrase probabilities of matching phrases found in the cache and the decay factor. Their examples illustrate better translation especially in repetition and consistency, however, the experimental score show modest improvements. Gong et al. (2011) extended this work by using three caches: dynamic cache, static cache, and topic cache. They show a better improvement when all three caches are used in combination.

Other efforts are in document-level decoding. Focusing on translation consistency, Xiao et al. (2011) employed a forced-decoding method: identify ambiguous words in the output

[2]As shown in the recent share task (CoNLL-2016), the precision of the state-of-the-art discourse parser is only about 40%.

of baseline system, and then obtain a set of consistent translations based on frequencies and finally re-decode input using the filtered set of translation options. Hardmeier et al. (2012) approach translation as an optimization task. He proposed a stochastic local search decoding method for PBSMT, which permits free document-wide dependencies in the models. Their work on decoding try to reduce the searching space but it is difficult to incorporate new knowledge.

Recently, researchers began to explore NN-based document-level approaches for sequence modeling. Conversational models need to predict the next sentence by considering the historical utterances in a conversation. Vinyals and Le (2015) built an end-to-end conversational system using a sequence-to-sequence framework. In order to capture the lager-context information, they simply concatenate previous utterances together as the input. Their preliminary results show that the method is able to converse well and extract knowledge from lager-context. Li et al. (2016) argue that simply incorporating context information into context independent message will increase the workload of a generation system and has the risk of bringing in noise to the generation process. To better preserve the original search intent, Sordoni et al. (2015) proposed a novel HRED to summarize these historical queries. Besides, Serban et al. (2016) adopt the framework to the task of dialogue response generation. They use HRED to summarize a single representation from both the current and previous sentences. Experiments demonstrated that availing of the historical representation helps to maintain the dialogue context.

The continuous vector representation of a symbol encodes multiple dimensions of similarity, equivalent to encoding more than one meaning of a word. Consequently, NMT needs to spend a substantial amount of its capacity in disambiguating source and target words based on the context defined by a source sentence (Choi et al. 2017). Without additional information, standard NMT models are facing inconsistency and ambiguity problems. Calixto and Liu (2017) utilize global image features extracted using a pre-trained convolutional neural network and incorporate them in NMT. Our work is also related to multi-source (Zoph and Knight 2016a) and multi-target NMT (Dong et al. 2015), which incorporate additional

source or target languages. They investigate one-to-many or many-to-one languages translation tasks by integrating additional encoders or decoders into encoder-decoder framework, and their experiments show promising results. More recently, some researchers propose to use an additional set of an encoder and attention to model more information. For example, Jean et al. (2017) use it to encode and select part of the previous source sentence for generating each target word.

Their work encourages us to explore document-level NN models such as HRED for translation task. It can be expected that the powerful structural representations will help to improve the performance of NMT in terms of coherence and consistency.

More recently, there are some new work on document-level NMT. In order to evaluate discourse phenomena in NMT, Bawden et al. (2018) conducted experiments from three aspects: 1) comparing multi-encoder models Zoph and Knight (2016b), Jean et al. (2017) with different strategies; 2) investigating the impacts of source- and target-side history information on NMT; 3) presenting a novel evaluation through the use of two discourse test sets targeted at coreference and lexical coherence/cohesion. Voita et al. (2018) introduced a context-aware model and demonstrated its usefulness for anaphora resolution as well as translation. Besides, Xiong et al. (2018) proposed to use discourse context and reward to refine the translation quality from the perspective of coherence. Some researchers proposed to extend the Transformer model to take advantage of document-level context (Miculicich et al. 2018, Zhang et al. 2018). Following Tu et al. (2017a)'s work, Kuang et al. (2017) and Maruf and Haffari (2017) continue to exploit cache memory for improving the performance of document-level NMT. Through human evaluation, Läubli et al. (2018) found that document-level evaluation for MT can improve to discriminate the errors which are hard or impossible to spot at the sentence level.

2.3.2 Discourse Phenomenon in Machine Translation

As discussed in Section 2.2.1, the main phenomena of cohesion is pronominal anaphora. Targeting cohesion phenomena, some researchers investigated approaches of incorporating

anaphora information to improve the performance of MT. For instance, Le Nagard and Koehn (2010) presented a method to aid English pronoun translation into French for SMT by integrating an anaphora resolution system. In the thesis, we mainly focus on the more complicated phenomenon: DP, which can be regarded as a special case of pronominal anaphora. Thus, in the following contents, we mainly review related work on DP.

Dropped Pronoun Recovery There are two research strands related to DP recovery. One is called Zero Pronoun (ZP) resolution. ZP resolution contains three steps: ZP detection, anaphoricity determination and reference linking. Zhao and Ng (2007), Kong and Zhou (2010), Chen and Ng (2013) proposed rich features using different machine learning models. For example, Chen and Ng (2013) propose a Support Vector Machine (SVM) classifier using 32 features including lexical, syntax and grammatical roles and show significant improvement on this task. Another research direction is related to a wider range of Empty Category (EC) phenomena (Yang and Xue 2010, Cai et al. 2011, Xue and Yang 2013), which aims to recover long-distance dependencies, discontinuous constituents and certain dropped elements (*e.g.,* trace markers, DPs, big PRO[3] etc., while we only focus on DPs.) in phrase structure treebanks (Xue et al. 2005). However, their work mainly focuses on intra-sentential characteristics as opposed to the discourse level. More recently, Yang et al. (2015) explored DP recovery for Chinese text messages based on both ZP and EC.

Most of their work either applies manual annotation (Yang et al. 2015) or uses existing but small-scale resources (*e.g.,* OntoNotes corpus contains 144K coreference instances, but only 15% of them are dropped subjects). There are two drawbacks on current work: 1) performance is not reliable when directly using the results of these systems in translation process; 2) the data is not big enough to drive a large neural model. Therefore, the primary challenge of this work is how to automatically build a large-scale high-quality DP training corpus.

[3] A pronominal determiner phrase without phonological content.

Dropped Pronoun Translation Some work has been done on DP translation for SMT models (Chung and Gildea 2010, Le Nagard and Koehn 2010, Taira et al. 2012, Xiang et al. 2013). Le Nagard and Koehn (2010) presented a method to aid English pronoun translation into French by using the results of a Coreference Resolution (CR) system, Unfortunately, their results are not convincing due to the poor performance of the CR system (Pradhan et al. 2012). Chung and Gildea (2010) systematically examine the effects of EC on MT with three methods: pattern, Conditional Random Field (CRF) (which achieves best results) and parsing. The results show that this work can really improve the end translation, even though the automatic prediction of EC is not highly accurate. Furthermore, Taira et al. (2012) propose both simple rule-based and manual methods to add DPs on the source side for Japanese–English translation. However, the BLEU scores of both methods are nearly identical, which indicates that only considering the single source sentence and forcing the insertion of pronouns may be less principled than tackling the problem head on by integrating them into the SMT model itself.

Their work regards the task of DP/EC recovering as a pre-processing stage for MT. Although these parameters are tuned independently, this direct idea is still worth trying. DP neural translation received relatively little attention from the MT community, thus we are encouraged to explore DP translation for NMT models.

2.4 Summary

In this chapter, we provided detailed background information related to this thesis. We first reviewed the frameworks, models and evaluation metrics of MT, including SMT and NMT. We also gave basic information about discourse and its key properties. We then studied the related work on discourse-aware MT, including discourse/document structure and discourse phenomena.

In the next chapter, we will address our first research question:

RQ 1 *What is the influence of historical contextual information on the per-*

formance of neural machine translation? Can a document-level NMT architecture alleviate inconsistency and ambiguity problems?

We present a novel document-level NMT architecture to capture dependencies across sentences, and improve the translation performance.

Chapter 3

Document-Level Neural Machine Translation

In the previous chapter, we review the background of machine translation and discourse. In this chapter, we introduce a novel DNMT architecture. We describe our first attempt at investigating the potential for implicitly incorporating discourse information into NMT. This chapter directly addresses our first research question as described as **RQ1**, regarding the influence of global context on NMT performance.

> **RQ 1** *What is the influence of historical contextual information on the performance of neural machine translation? Can a document-level NMT architecture alleviate inconsistency and ambiguity problems?*

This chapter is organized as follows. Without loss of generality, we first introduce the motivation of our work on DNMT in Section 3.1. In Section 3.2, we describe our proposed approaches to model cross-sentence context for boosting sentence-level NMT. For further comparison with related work, in Section 3.3, we also review two other DNMT models recently proposed by Jean et al. (2017) and Tu et al. (2018), respectively. The experiments for verifying our proposed approaches are reported in Section 3.4. Quantitative and qualitative analysis is presented in Section 3.5. Finally, we systematically compare our

approaches with these two related models in Section 3.6, which is followed by the summary in Section 3.7.

3.1 Why Global Context?

MT usually models a text by considering isolated sentences based on a strict assumption that the sentences in a text are independent of one another. As we demostrate in this chapter, disregarding dependencies across sentences will negatively affect translation outputs of a text especially in terms of consistency. Although document-aware approaches have been investigated for SMT (Tiedemann 2010, Gong et al. 2011, Xiao et al. 2011, Hardmeier et al. 2012), leveraging global context for NMT has received relatively little attention from the research community. With the advantages of neural networks described in Chapter 2, the performance of NMT has surpassed the performance of conventional SMT on various language pairs (Luong et al. 2015a). Therefore, exploring document-aware approaches for NMT has the potential to further improve translation quality over state-of-the-art MT models.

The continuous vector representation of a symbol (namely h_j, the encoder hidden state of the j-th source word) encodes multiple dimensions of similarity, equivalent to encoding more than one meaning of a word. Consequently, NMT needs to spend a substantial amount of its capacity in disambiguating source and target words based on the context defined by a source sentence (Choi et al. 2017). In Table 3.1, we show an example of the problem of ambiguity in MT. The translation of "机遇" (*i.e.,* "opportunity") suffers from an ambiguity problem, and it is incorrectly translated into "challenge". Figure 3.1 further indicates that the problem is not caused by attending to the wrong source words but the lack of larger context. Consistency is another critical issue in document-level translation, where a repeated term should keep the same translation throughout the whole document (Xiao et al. 2011, Carpuat and Simard 2012). As shown in Table 3.2, the Chinese word "问题" has multiple English translations such as "problem", "question" and "issue". However,

I/O	Sentences
Input	... 开始 都 觉得 ... 大家 觉得 这 也是 一次 **机遇** , 一次 挑战 。
Reference	... initially they all felt that ... everyone felt that this was also an **opportunity** and a *challenge* .
NMT Output	... felt that ... we feel that it is also a **challenge** and a *challenge* .

Table 3.1: An example of the problem of ambiguity in NMT.

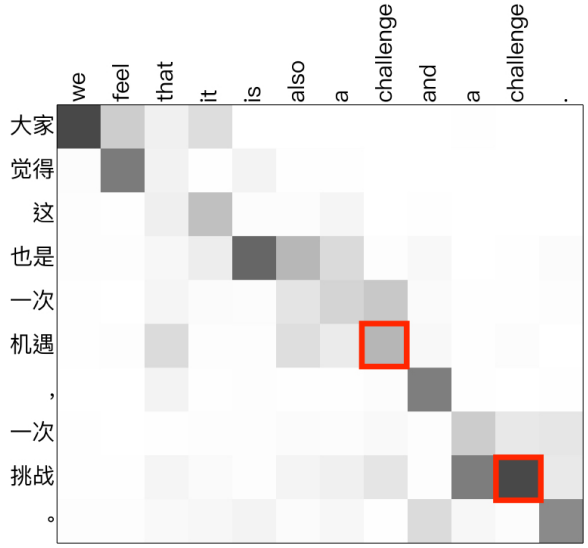


Figure 3.1: Attention matrix of the example in Table 3.1.

the Chinese word should be always translated into "issue" according to the larger context in the document. Nevertheless, current NMT models still process a document by translating each sentence alone, suffering from ambiguity and inconsistency problems arising from a single source sentence as demonstrated. These problems are difficult to alleviate using only limited intra-sentence context.

Cross-sentence context, or global context, has proven helpful to better capture the meaning or intention in sequential tasks such as query suggestion (Sordoni et al. 2015) and dialogue modeling (Vinyals and Le 2015, Serban et al. 2016). Therefore, we propose a cross-sentence context-aware NMT model, which considers the influence of previous sentences in the same document. First, this history is summarized in a hierarchical way. We then integrate the historical representation into NMT in different strategies.

Document	I/O	Sentences
Past	Input	那么 在 这个 问题 上 , 伊朗 的 ...
	Output	well, on this *question* , iran has a relatively ...
	Input	在 任内 解决 伊朗 核 问题 , 不管是 用 和平 ...
	Output	to resolve the iranian nuclear *problem* in his term , ...
Current	Input	那 刚刚 提到 这个 ... 谈判 的 问题 。
	Output	that just mentioned the *issue* of the talks ...

Table 3.2: An example of the problem of consistency in NMT.

In different use-cases, DNMT can consider: 1) either the source or target sentences, or both; 2) either the preceding or following sentences, or both. Actually, in our preliminary experiments, considering target-side history inversely harms translation performance, since it suffers from serious error propagation problems. Furthermore, we set the use-case as pipeline translation. Therefore, our models mainly consider the source-side previous sentences in the same document.

As shown in Section 3.4, experimental results on a large Chinese-English translation task show that our approach significantly improves upon a strong attention-based NMT system by up to +2.1 BLEU points.

3.2 Cross-Sentence Neural Machine Translation Models

In this section, we introduce our proposed approach in detail, which contains two parts: *summarizing global context* and *integrating global summary into NMT*.

3.2.1 Summarizing Global Context

We propose to use a hierarchy of RNN to summarize the cross-sentence context from previous sentences, which deploys an additional document-level RNN on top of the sentence-level RNN encoder (Sordoni et al. 2015). Note that we employ left-to-right RNN, which put more emphasis on the end of the sentence. We hypothesis that the closest a sentence to the current one, the more relevant the first is expected to be for the translation of the latter.

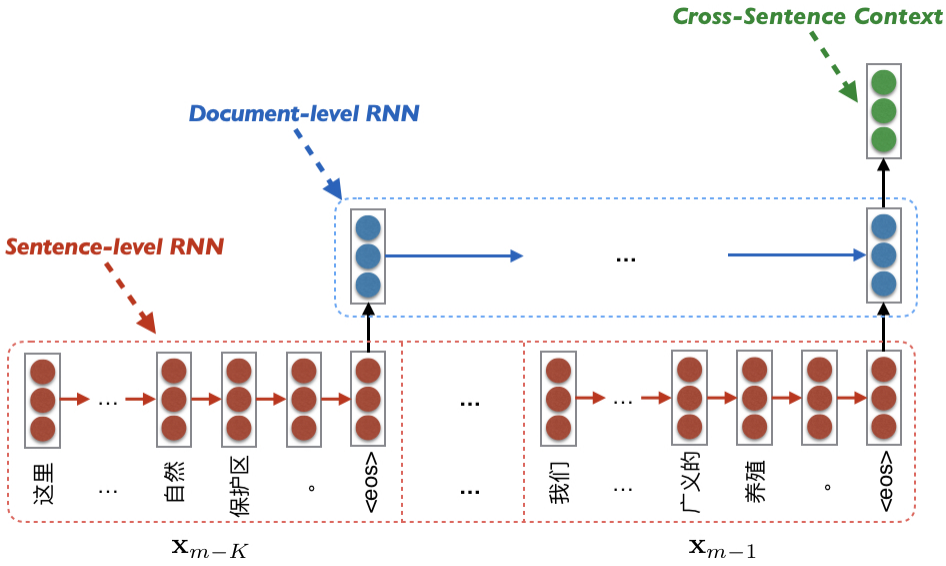


Figure 3.2: Summarizing global context with a hierarchical RNN ($\mathbf{x}_m$ is the m-th source sentence).

Formally, given a source sentence $\mathbf{x}_m$ (*i.e.,* the m-th source sentence in a document) to be translated, we consider its K previous sentences in the same document as cross-sentence context, which can be described as $C = \{\mathbf{x}_{m-K}, ..., \mathbf{x}_{m-1}\}$. As shown in Figure 3.1, we summarize the representation of C in a hierarchical way as described below.

Sentence-Level RNN For a sentence $\mathbf{x}_k$ in cross-sentence context C, the sentence-level RNN reads the corresponding words $\{x_{1,k}, ..., x_{n,k}, \ldots, x_{N,k}\}$ sequentially and updates its hidden state as in Equation (3.1):

$$h_{n,k} = f(h_{n-1,k}, x_{n,k}) \tag{3.1}$$

where $f(\cdot)$ is an activation function, and $h_{n,k}$ is the hidden state at time step n. The length of $\mathbf{x}_k$ is N. The last state $h_{N,k}$ stores order-sensitive information about all the words in $\mathbf{x}_k$, which is used to represent the summary of the whole sentence, *i.e.,* $S_k \equiv h_{N,k}$. After processing each sentence in C, we can obtain all K sentence-level representations, which will be fed into document-level RNN.

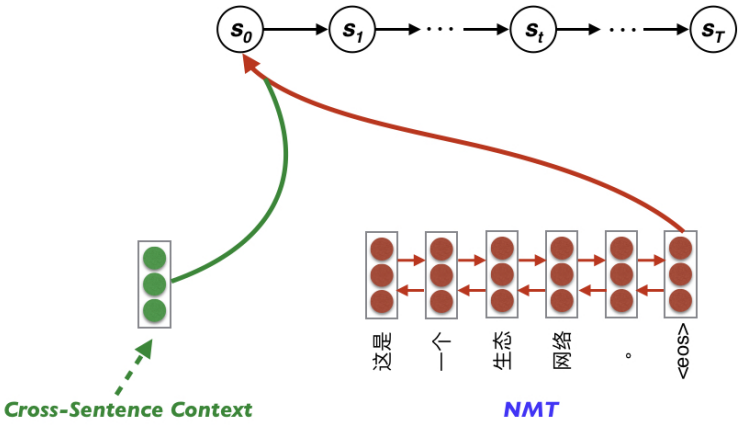


Figure 3.3: The *Initialization* integration strategy.

Document-Level RNN It takes as input the sequence of the above K sentence-level representations $\{S_1, ..., S_k, ..., S_K\}$ and computes the hidden state according to Equation (3.2):

$$z_k = g(z_{k-1}, S_k) \tag{3.2}$$

where $g(\cdot)$ is an activation function, and z_k is the recurrent state at time step k, which summarizes the previous sentences that have been processed to the position k. Similarly, we use the last hidden state to represent the summary of the global context, *i.e.,* $D \equiv z_K$.

After the above two-level encoding, we hypothesize that the global context D has contained rich information from the previous K sentences. It not only captures the dependencies between words, but also considers the discourse relations between sentences. Next, we will integrate D into a standard NMT.

3.2.2 Integrating Global Context into Neural Machine Translation

After obtaining the global context, we design four strategies to integrate history representation D into NMT to translate the current sentence $\mathbf{x}_m$: *Initialization*, *Auxiliary Context*, *Gating Auxiliary Context* and *Combination*.

Initialization Global context can be used as a warm-start to encoder and decoder states during NMT training. As described in Section 2.1.2, the encoder is used to summarize the

source sentence into a vector representation. However, the standard NMT model usually uses all-zero states to initialize its encoder (Bahdanau et al. 2015), without considering any history context. When a human translator translates a sentence, he/she usually retains the knowledge from previous sentences as a background. Therefore, we propose to use the global context D as the initialization state of NMT encoder.

The decoder is employed to generate the target sentence word by word based on the source-side vector representation. For the standard decoder, the initial hidden state is computed as in Equation (3.3):

$$s_0 = \tanh(W_s h_N) \tag{3.3}$$

which uses the last hidden state of the encoder for initialization. This method only uses the information from current source sentence without considering the useful history contexts. Therefore, we rewrite the calculation of the initial hidden state as in Equation (3.4):

$$s_0 = \tanh(W_s h_N + W_D D) \tag{3.4}$$

where h_N is the last hidden state in the encoder and $\{W_s, W_D\}$ are the corresponding weight matrices. As shown in Figure 3.3, we use the history representation D to initialize either the NMT encoder, NMT decoder, or both.

Auxiliary Context As shown in Figure 3.4, the history representation D is used as static cross-sentence context, which works together with the dynamic intra-sentence context produced by an attention model.

In standard NMT, as shown in Figure 3.5 (a), the decoder hidden state at time step i is computed by Equation (3.5):

$$s_i = f(s_{i-1}, y_{i-1}, c_i) \tag{3.5}$$

where y_{i-1} is the most recently generated target word, and c_i is the intra-sentence context summarized by the NMT encoder at time step i. Our *Auxiliary Context* strategy, as shown in Figure 3.5 (b), adds the representation of cross-sentence context D to jointly update the

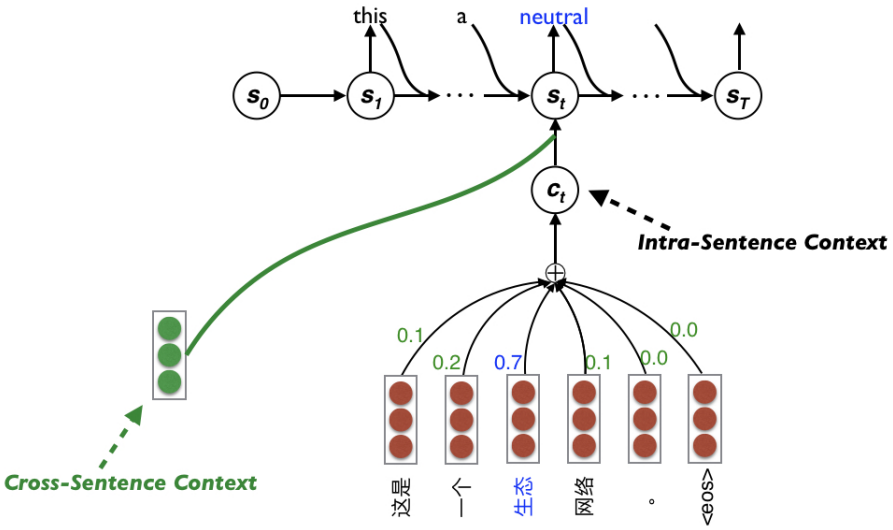


Figure 3.4: The *Auxiliary Context* integration strategy.

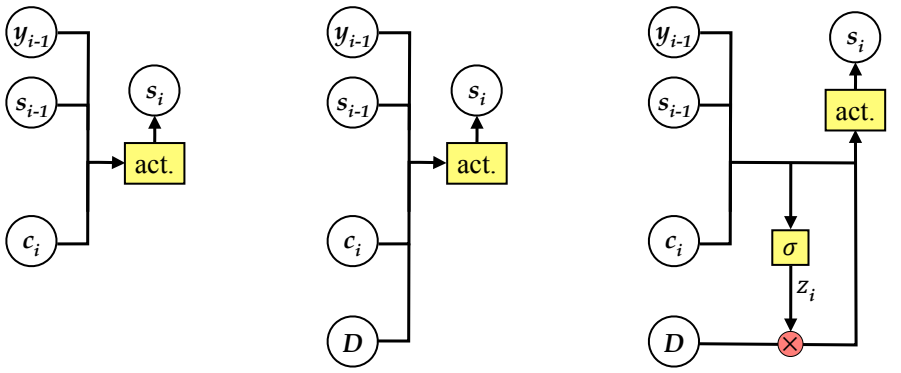
decoding state s_i, as in Equation (3.6):

$$s_i = f(s_{i-1}, y_{i-1}, c_i, D) \tag{3.6}$$

Now the proposed NMT decoder has four inputs rather than the three original ones. The concatenation $[c_i, D]$, which embeds both intra- and cross-sentence contexts, can be fed to the decoder as a single representation. From an implementational point of view, all we need to do is modifying the size of the corresponding parameter matrix. In this strategy, D serves as an auxiliary information source to better capture the meaning of the source sentence.

Gating Auxiliary Context We add a gate to *Auxiliary Context*, which decides the amount of global context to be used in generating the next target word at each step of the decoding process.

The starting point for this strategy is an observation: the need for information from the global context differs from step to step during generation of the target words. For example, global context is more in demand when generating target words for ambiguous source words, and less for more straightforward words. To this end, we extend our auxiliary context strategy by introducing a context gate (Tu et al. 2017a) to dynamically control the



(a) standard decoder (b) decoder with auxiliary context (c) decoder with gating auxiliary context

Figure 3.5: Architectures of NMT with auxiliary context integrations. $act.$ is the decoder activation function, and σ is a sigmoid function.

amount of information flowing from the auxiliary global context at each decoding step, as shown in Figure 3.5 (c).

Intuitively, at each decoding step i, the context gate looks at the decoding environment (*i.e.,* s_i, y_{i-1}, and c_i), and outputs a number between 0 and 1 for each element in D, where 1 denotes "completely transferring this" while 0 denotes "completely ignoring this". The global context vector D is then processed with an element-wise multiplication before being fed to the decoder activation layer.

Formally, the context gate consists of a sigmoid neural network layer and an element-wise multiplication operation. It assigns an element-wise weight to D, computed by Equation (3.7):

$$z_i = \sigma(U_z s_{i-1} + W_z y_{i-1} + C_z c_i) \tag{3.7}$$

where y_{t-1} is the previously generated word, s_t is the t-th decoding hidden state, and c_t is the t-th source representation. Here $\sigma(\cdot)$ is a logistic sigmoid function, and $\{W_z, U_z, C_z\}$ are the weight matrices, which are trained to learn when to exploit global context to maximize the overall translation performance. Note that z_i has the same dimensionality as D, and thus each element in the global context vector has its own weight. Accordingly, the

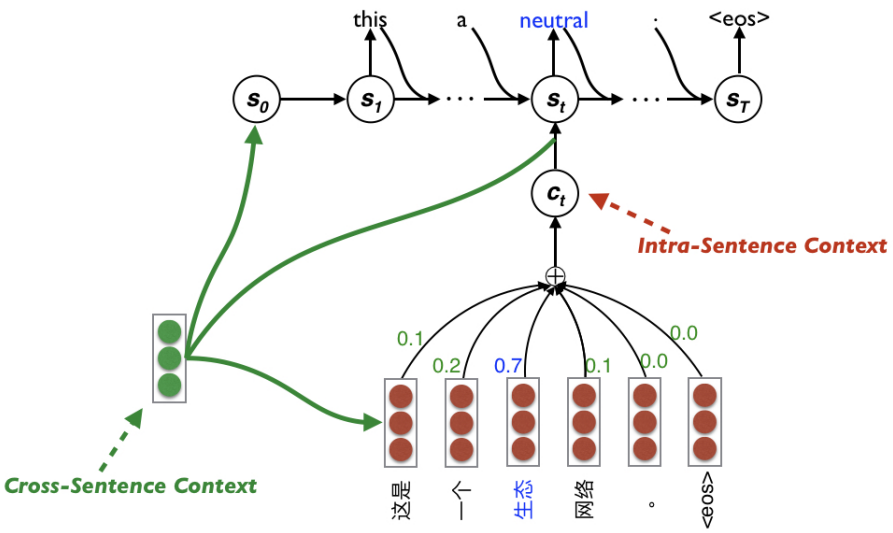


Figure 3.6: Architectures of NMT with *initialization* + *(Gating) Auxiliary Context* integration strategy.

decoder hidden state is updated by Equation (3.8):

$$s_i = f(s_{i-1}, y_{i-1}, c_i, z_i \otimes D) \tag{3.8}$$

Combination The global context D applied with different strategies may transfer different patterns of history information (*e.g.,* word-level dependencies and sentence-level relations) to NMT. Thus, combining multiple strategies together can encourage NMT to better learn history information from larger context.

Finally, we propose to combine *initialization* and *(Gating) Auxiliary Context* integration strategies for NMT, as shown in Figure 3.6.

3.3 Related Document-Level Neural Machine Translation Work

Our work and that of Jean et al. (2017) are two independently early attempts to model cross-sentence context for NMT. To model previous sentences, we employed an hierarchical RNN encoder while Jean et al. (2017) used an additional attention-encoder model. Tu et al. (2018) continue to explore document-level translation using cache-based approaches. In

this section, we mainly focused on these three representative models and introduce their architectures in detail.

More recently, there are some new document-level NMT models were proposed following the above work (as discussed in Section 2.3.1). Bawden et al. (2018) presented a novel evaluation to investigate discourse phenomena in different multi-encoder models (Zoph and Knight 2016b, Jean et al. 2017). They presented a novel evaluation through the use of two discourse test sets targeted at coreference and lexical coherence/cohesion. Furthermore, some researchers extended our cross-sentence models on the top of the state-of-the-art Transformer architecture (Miculicich et al. 2018, Zhang et al. 2018, Voita et al. 2018). Following Tu et al. (2017a)'s work, Kuang et al. (2017) and Maruf and Haffari (2017) continued to exploit cache memory for improving the performance of document-level NMT. In addition to this, some researchers started new strands of document-level NMT. For example, Xiong et al. (2018) proposed to use discourse context and reward to refine the translation quality from the perspective of coherence while our work mainly focus on consistency and disambiguity.

3.3.1 Multi-Encoder

Originally, multi-source (Zoph and Knight 2016a) and multi-target NMT (Dong et al. 2015) were proposed to incorporate additional source or target languages. They investigate one-to-many or many-to-one language translation tasks by integrating additional encoders or decoders into the standard encoder-decoder framework, with promising results.

Recently, some researchers have proposed using an additional encoder-attention set to model more information (*e.g.,* history context or multimodal features). Jean et al. (2017) proposed a multi-encoder approach to encode and select part of the previous source sentence for generating each target word. Calixto and Liu (2017) utilized global image features extracted using a pre-trained convolutional neural network and incorporated them into NMT. Taking Jean et al. (2017)'s model for example, as shown in Figure 3.7, $\mathbf{x}_m$ is the current source sentence to be translated and $\mathbf{x}_{m-1}$ is its previous sentence in the document. There

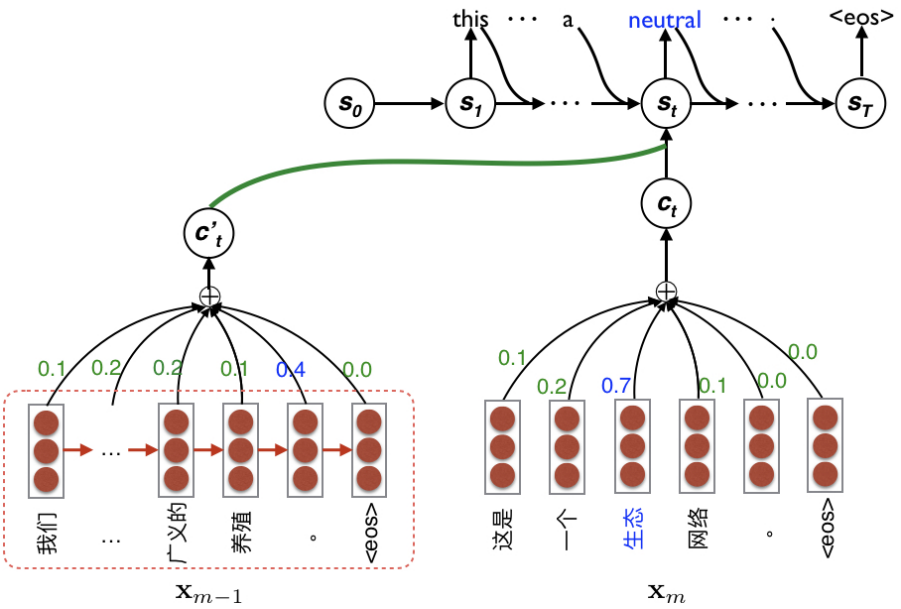


Figure 3.7: Architecture of Multi-Encoder NMT.

are two context vectors from both the current source sentence c_i and its previous sentence c'_i. Accordingly, the NMT decoder hidden state s_i at time i is updated to Equation (3.9):

$$s_i = f(s_{i-1}, y_{i-1}, c_i, c'_i) \tag{3.9}$$

where $f(\cdot)$ is an activation function, and y_{i-1} is the most recently generated target word.

As additional attention leads to more computational cost, Jean et al. (2017) only incorporate limited information such as the single preceding sentence in experiments. However, our architecture is free of this limitation, so we investigated more preceding sentences (e.g. $K = 3$) in our model.

3.3.2 Cache Memory

Neural Turing Machines (Graves et al. 2014) and Memory Networks (Weston et al. 2014, Sukhbaatar et al. 2015) are early models that augment neural networks with a possibly large external memory. Specifically, the Key-Value Memory Network (Miller et al. 2016) is a simplified version of Memory Networks with better interpretability and has yielded

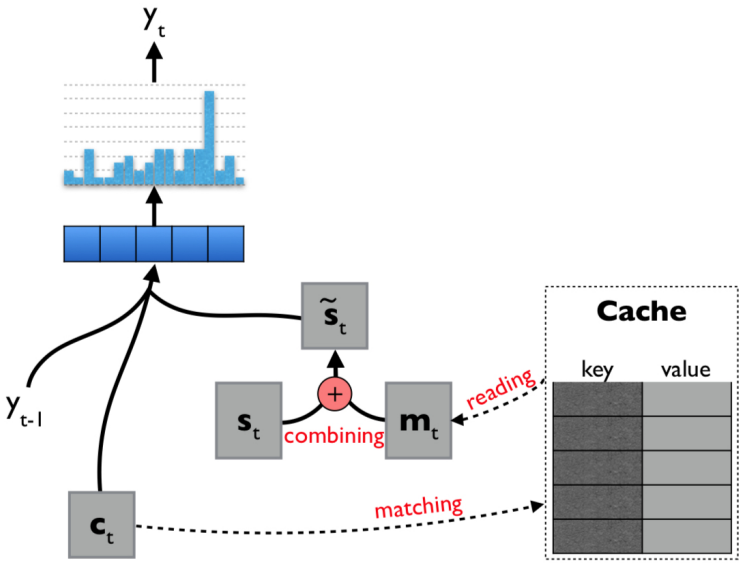


Figure 3.8: Architecture of NMT with a continues cache.

encouraging results in document reading (Miller et al. 2016), question answering (Pritzel et al. 2017) as well as machine translation (Gu et al. 2017, Kaiser et al. 2017). Kaiser et al. (2017) use an external key-value memory to remember rare training events at test time, and Gu et al. (2017) use a memory to store a set of sentence pairs retrieved from the training corpus given the source sentence.

More recently, Tu et al. (2018) adopted memory networks for the task of document-level MT. As shown in Figure 3.8, they proposed to augment NMT models with a cache-like memory network, which stores the translation history in terms of bilingual hidden representations at decoding steps of previous sentences. The cache component is an external key-value memory structure with the keys being attention vectors, and values being decoder states collected from translation history. At each decoding step, the probability distribution over generated words is updated online depending on the history information retrieved from the cache with a query of the current attention vector. Using simply a dot-product for key matching, this history information is quite cheap to store and can be accessed efficiently.

3.3.3 Comparison

Apart from the architecture, there are also two main differences among these three models. First, ours and multi-attention model only exploit source-side history contexts, while cache-based model is able to take advantage of bilingual contexts. The main reason is that multi-attention and our models still store the history context in the form of surface word. This leads to the problem of error propagation when considering the auto-translated history contexts. However, this problem can be alleviated by directly leveraging continuous vectors to represent translation history in the cache-based model. Second, the sizes of history context are different among them. As reported in their papers, multi-attention model considers $N = 1$ history sentence while our model uses $N = 3$. Besides, the cache-based model uses 500 history words (equivalent to $N = 15$ sentences), which is much larger than that of other two models. In comparison experiments (in Section 3.6), we keep their different settings to achieve the best performances.

3.4 Experiments

In this section, we describe the experiment setup and then show the results of our proposed models, against the baselines mentioned.

3.4.1 Data

The Linguistic Data Consortium (LDC)[1] is an open consortium of universities, companies and government research laboratories. It creates, collects and distributes speech and text databases, lexicons, and other resources for linguistics research and development purposes. In recent years, most MT experiments are conducted on LDC corpora. For example, the famous shared task, Open Machine Translation Evaluation 2001-2015,[2] requires to compare different MT systems trained on LDC data.

[1] Available at: `https://www.ldc.upenn.edu`.

[2] `https://www.nist.gov/itl/iad/mig/open-machine-translation-evaluation`.

Data	$\|S\|$	$\|W\|$		$\|V\|$		$\|L\|$	
		Zh	En	Zh	En	Zh	En
LDC	1.25M	27.93M	34.51M	223.25K	114.83K	22.29	27.54
MT05	1,083	29.90K	34.79K	5.64K	1.97K	27.61	32.12
MT06	1,665	38.35K	47.33K	6.72K	2.45K	23.03	32.12
MT08	1,358	32.31K	41.10K	6.74K	2.50K	23.79	30.27

Table 3.3: Number of sentences ($|S|$), words ($|W|$), vocabulary ($|V|$), and averaged sentence length ($|L|$) comprising the training, tuning and test corpora.

We carried out experiments on Chinese–English translation tasks. The data were extracted from the LDC. As the document information is necessary when selecting the previous sentences, we collect all LDC corpora that contain document boundaries and combine them together as our training data.[3] For validation and testing, we use the data sets from NIST Open Machine Translation Evaluation[4] (OpenMT), which also contain document boundaries. We chose the NIST05 (MT05) as our tuning set, and NIST06 (MT06) and NIST08 (MT08) as test sets.[5]

We apply standard MT corpus preparation methods[6] to pre-process all the data. In particular, we employ the Jieba toolkit[7] for Chinese word segmentation, and Moses (Koehn et al. 2007) toolkit[8] for English word tokenization. We also clean the training data by filtering sentences with more than 80 tokens. The statistics of the corpora used for the experiment are listed in Table 3.3. As seen, the training corpus contains more than 1.25 million sentence pairs and the tuning/test sets contain around 1,500 sentence pairs. The average lengths of sentences in these corpora are similar. We used case-insensitive BLEU score (Papineni et al. 2002) as our evaluation metric, and sign-test (Collins et al. 2005) for calculating statistical significance.

[3]The indexes of selected LDC corpora are: 2003E07, 2003E14, 2004T07, 2005E83, 2005T06, 2006E24, 2006E34, 2006E85, 2006E92, 2007E87, 2007E101, 2007T09, 2008E40, 2008E56, 2009E16, 2009E95.

[4]`https://www.nist.gov/itl/iad/mig/open-machine-translation-evaluation`.

[5]The LDC indexes of these corpora are LDC2010T14, LDC2010T17 and LDC2010T21, respectively.

[6]Available at `http://www.statmt.org/moses/?n=Moses.Baseline`.

[7]Available at `https://github.com/fxsjy/jieba`.

[8]Available at `http://www.statmt.org/moses`.

3.4.2 Models Setup

The Workshop on Machine Translation (WMT) is one of popular annual share task on MT. The best model in WMT is usually regarded as the recent state-of-the-art benchmark. Therefore, we reuses the best settings Wang et al. (2014), Sennrich et al. (2016, 2017) to setup SMT and NMT models in our experiments.

For training the SMT models, we employ the phrase-based model in Moses[9]. Furthermore, we train a 5-gram language model using the SRI Language Modelling Toolkit[10] (Stolcke 2002). To obtain word alignment, we run GIZA++[11] (Och and Ney 2003) on the training data. We use minimum error rate training (Och 2003) to optimize the feature weights.

For training the NMT models, we extended the open source attention-based NMT model, Nematus[12] (Sennrich et al. 2017), with our cross-sentence modelling code. We limited the source and target vocabularies to the most frequent 35K words in Chinese and English, covering approximately 97.1% and 99.4% of the data in the two languages, respectively. Each model was trained on sentences of length up to 80 words in the training data with early stopping. The word-embedding dimension was 600, the hidden layer size was 1000, and the batch size was 80. All our models considered the previous three sentences (i.e., $K = 3$) as cross-sentence context.[13] We trained for 20 epochs using Adadelta (Zeiler 2012), and selected the model that yielded the best performance on the tuning set. It should be emphasized that we did not use the pre-train strategy, since we found training from scratch achieved a better performance.

3.4.3 Results

Table 3.4 shows the translation performance in terms of BLEU score. We investigate two baselines and six proposed variation models including:

[9] Available at: `http://www.statmt.org/moses`.

[10] Available at: `http://www.speech.sri.com/projects/srilm/`.

[11] Available at: `http://web.archive.org/web/20100221051856/http://code.google.com:80/p/giza-pp`.

[12] Available at `https://github.com/EdinburghNLP/nematus`.

[13] In our preliminary experiments, we have tested different history length (*i.e.,* $K = 1$, $K = 3$ and $K = 5$), and found that models with $K = 3$ can achieve the best translation performance.

#	System	MT05	MT06		MT08		Average	
		BLEU	BLEU	△	BLEU	△	BLEU	△
1	SMT BASE	33.08	32.69	–	23.78	–	28.24	–
2	NMT BASE	34.35	35.75	–	25.39	–	30.57	–
3	+Init$_{enc}$	36.05	36.44†	+0.69	26.65†	+1.26	31.55	+0.98
4	+Init$_{dec}$	36.27	36.69†	+0.94	27.11†	+1.72	31.90	+1.33
5	+Init$_{enc+dec}$	36.34	36.82†	+1.07	27.18†	+1.79	32.00	+1.43
6	+Auxi	35.26	36.47†	+0.72	26.12†	+0.73	31.30	+0.73
7	+Gating Auxi	36.64	37.63†	+1.88	26.85†	+1.46	32.24	+1.67
8	COMBINA.	**36.89**	**37.76**†	+2.01	**27.57**†	+2.18	**32.67**	+2.10

Table 3.4: Evaluation of translation quality. "Init" denotes Initialization of encoder ("enc"), decoder ("dec"), or both ("enc+dec"), and "Auxi" denotes Auxiliary Context. "†" indicates statistically significant difference ($P < 0.01$) from the baseline NMT. Average is calculated on test sets (*i.e.,* MT06 and MT08).

SMT BASE: SMT baseline trained using Moses;

NMT BASE: NMT baseline trained using Nematus;

+Initi$_{enc}$: NMT encoder is initialized by global context;

+Initi$_{dec}$: NMT decoder is initialized by global context;

+Initi$_{enc+dec}$: both NMT encoder and decoder are initialized by global context;

+Auxi: global context is used as auxiliary to jointly update each decoding state;

+Gating Auxi: a gate is added to *+Auxi* model;

COMBINATION: combining *+Gating Auxi* model with *+Initi$_{enc+dec}$* model.

Baselines (Rows 1-2)

As can be seen from the table, SMT BASE – the state-of-the-art SMT system, achieves 28.24 BLEU points on average, and NMT BASE – a traditional phrase-based NMT system, achieves 30.57 BLEU points on average. NMT BASE significantly outperforms SMT BASE by 2.3 BLEU points on average, indicating that it is a strong NMT baseline system. The

difference between NMT and SMT is consistent with the results in Tu et al. (2017b) (i.e., 26.93 vs. 29.41) on training corpora of a similar scale.

We use NMT BASE (Row 2) as our strong baseline for further comparison. Clearly the proposed models (Rows 3-8) significantly outperform the baseline in all cases, although there are considerable differences among different variations.

Initialization (Rows 3-5)

Init_{enc} and Init_{dec} improve translation performance by around +1.0 and +1.3 BLEU points (on average) individually, proving the effectiveness of warm-start with cross-sentence context. Furthermore, it also demonstrated that both NMT encoder and decoder need history information for source-side summarization and target-side generation. Combining these two individual initialization approaches achieves a weak further improvement (+0.1 BLEU point on average), which indicates that the NMT encoder and decoder may share global context knowledge to a large extent.

Auxiliary Context (Rows 6-7)

The auxiliary context strategy can achieve +0.73 BLEU points on average, indicating that global context is helpful in generating target words. Furthermore, the gating auxiliary context strategy achieves a significant improvement of around +1.0 BLEU point over its non-gating counterpart. This shows that the introduced context gate learns to distinguish the different needs of the global context for generating target words.

Combination (Row 8)

Finally, we combine the best variants from the *initialization* and *auxiliary context* strategies, *i.e.,* +$\text{Init}_{\text{enc+dec}}$+Gating Auxi. The combination model achieves the best performance overall, improving upon NMT by +2.1 BLEU points on average. This verifies our hypothesis in Section 3.2 that different strategies may capture different patterns of history information, and the two types of strategy are complementary to one another.

As described in Section 3.2.2, the "Initialization" methods can provide NMT models

more useful information in larger contexts. In particularly, it makes NMT more sensitive to repeated words in a text, and learn to generate consistent translations. Furthermore, the "Auxiliary Context" method acts on target word generation, which enhances the capability of disambuiting for NMT models. The gating mechanism can further improve the performance by filtering useless or noisy history contexts. Finally, combining these approaches together can accumulatively improve translation performance.

3.5 Analysis

In this section, we conducted extensive analysis to better understand our model in terms of alleviating ambiguity and inconsistency problems. As the combination model achieves the best performance, we analyze sentences from outputs generated by COMBINATION and NMT BASE models, respectively.

3.5.1 Effect of Global Context

We investigate to what extent the mistranslated errors are fixed by the proposed system. We randomly select 15 documents (*i.e.,* about 60 sentences) from the test sets (*i.e.,* outputs generated by COMBINATION and NMT BASE models). Actually, the size of sampled data is small, and we plan to label more data for human and automatic evaluation in future work.

As shown in Table 3.5, we count how many related errors: 1) are made by NMT BASE (*Total*), and 2) fixed by our COMBINATION model (*Fixed*); as well as 3) newly generated (*New*). Regarding the *Ambiguity* problem, while we found that 38 checkpoints (*i.e.,* words or phrases) were translated into incorrect equivalents, 76% of them are corrected by our model. Similarly, there are 32 *Inconsistency* errors made by the baseline system. Our model solved 75% of them including lexical, tense and definiteness (definite or indefinite articles) cases. However, we also observe that our system brings relative 21% new errors. According to the analysis, we confirm that the improvements of our models come from alleviating ambiguity and inconsistency problems.

Errors	Ambiguity		Inconsistency		All	
	Count	△	Count	△	Count	△
Total errors	38	–	32	–	70	–
Fixed errors	29	-76.32%	24	-75.00%	53	-75.71%
New errors	7	+18.42%	8	+25.00%	15	+21.43%

Table 3.5: Statistics of translation error analyzed on COMBINATION and NMT BASE outputs.

Length	MT05	MT06	MT08	Average
$K = 0$	34.35	35.75	25.38	30.57
$K = 1$	36.11	36.14	26.87	31.51 (+0.94)
$K = 3$	36.27	36.69	27.11	31.90 (+1.33)
$K = 5$	35.23	36.01	25.94	30.98 (+0.41)

Table 3.6: Evaluation of the "+Init$_{\text{dec}}$" model with different history lengths.

3.5.2 Effect of History Length

By analyzing the training corpus, we found that a document contains around 5 sentences in average. Thus, we mainly tested our models with three settings on history length (*i.e.,* $K = 1$, $K = 3$ and $K = 5$). In our preliminary experiments, we evaluated the "+Init$_{\text{dec}}$" model due to the fast training speed. As shown in Table 3.6, $K = 0$ represents the "NMT BASE" model, which can achieve 30.57 BLEU points. Our model improves the baseline by +0.94 BLEU point when $K = 1$ while +1.33 BLEU points when $K = 3$. However, the translation performance declines (only 30.98 BLEU points) when considering more history sentences ($K = 5$). In general, our model with $K = 3$ can achieve the best translation performance, and we finally choose $K = 3$ in our main experiments (in Section 3.4.3).

3.5.3 Case Study

As shown in Table 3.7, we also list two examples (selected from test sets) to explain how our approach alleviates translation problems.

The translation of the word "腐官" (*corrupt officials*) suffers from an ambiguity prob-

I/O	Sentences
History	这 不 等于 明着 提前 告诉 **贪官** 们 赶紧 转移 罪证 吗 ?
Input	能否 遏制 和 震慑 **腐官** ?
Reference	Can it inhibit and deter **corrupt officials**?
NMT BASE	Can we contain and deter the *enemy*?
OUR	Can it contain and deter the **corrupt officials**?
History	**中国队** 经常 是 在 形势 大好 的 情况 下 不会 踢球 , ...
Input	这 确实 是 **中国队** 不能 " 善终 " 的 一个 原因 。
Reference	This is indeed a reason why **the Chinese team** could not have a "good ending."
NMT BASE	This is indeed the reason why *China* can not be "hospice."
OUR	This is indeed one of the reasons why **the Chinese team** can not have a "good ending."

Table 3.7: Example translations. We italicize some *mistranslated* errors and highlight the **correct** ones in bold.

lem. Its word embedding vector encodes more than one notion of similarity. The intra-sentence context is insufficient to predict the correct translation. Thus, the word "腐官" is mistranslated as "*enemy*" by the baseline system. With the help of the similar word "贪官" in the previous sentence, our approach successfully correct this mistake. This demonstrates that cross-sentence context indeed helps resolve certain ambiguities.

The phrase "中国队" (*the Chinese team*) has already arisen in previous sentence, and it is translated into the correct English phrase. However, when the phrase appears again in the current source sentence, it is mistranslated as "*China*" by the baseline system with a local intra-sentence context. By considering global context, our model successfully fixes this inconsistency error.

3.6 Comparison with Related Work

In this section, we conduct experiments to compare our approach with other DNMT models (*i.e.,Multi-Encoder* and *Cache Memory*) as discussed in Section 3.3.

3.6.1 Data

We carried out experiments on Chinese–English translation tasks on multiple domains, each of which differs from the others in terms of topic, genre and style.

LDC The training corpus is the same as that used in the main experiment described in Section 3.4.1. Most sentences in this corpus come from the news domain. They are formal articles with syntactic structures such as complicated conjoined phrases, which make textual translation very difficult. We choose the NIST 2002 (MT02) dataset as our tuning set, and the NIST 2003-2008 (MT03-MT08) datasets as test sets.

Subtitle The subtitles are extracted from TV episodes, which are usually simple and short.[14] Most of the translations of subtitles do not preserve the syntactic structures of their original sentences at all. We randomly select two episodes as the tuning set, and as other two episodes as the test set.

TED The corpora are from the MT track on TED Talks of IWSLT2015 (Cettolo et al. 2012).[15] Koehn and Knowles (2017) point out that NMT systems have a steeper learning curve with respect to the amount of training data, resulting in worse quality in low-resource settings. The TED talks are difficult to translate given the variety of topics in quite small-scale training data. We choose the "dev2010" dataset as the tuning set, and the combination of "tst2010-2013" datasets as the test set.

We pre-process the data using the same methods as in Section 3.4.1. The statistics of the corpora are listed in Table 3.8. As can be seen, the average lengths of the source sentences in LDC, TVsub, and TED corpora are 22.3, 5.6, and 19.5 words, respectively. We again used case-insensitive BLEU score (Papineni et al. 2002) as our evaluation metric, and sign-test (Collins et al. 2005) for calculating statistical significance.

[14] Available at: `https://github.com/longyuewangdcu/tvsub`.
[15] Available at: `https://wit3.fbk.eu/mt.php?release=2015-01`.

Corpus	Set	$\|S\|$	$\|W\|$		$\|V\|$		$\|L\|$	
			Zh	En	Zh	En	Zh	En
LDC	Train	1.25M	27.93M	34.51M	223.25K	114.83K	22.29	27.54
	Tune	1.08K	29.90K	34.79K	5.64K	1.97K	27.61	32.12
	Test	3.02K	70.66K	88.43K	12.40K	8.61K	23.40	29.28
TVsub	Train	2.15M	12.10M	16.60M	151.00K	90.80K	5.63	7.71
	Tune	1.09K	6.67K	9.25K	1.74K	1.35K	6.14	8.52
	Test	1.15K	6.71K	9.49K	1.79K	1.39K	5.82	8.23
TED	Train	0.21M	4.1M	4.4M	85.66K	54.24K	19.52	20.95
	Tune	0.89K	21.3K	17.5K	3.87K	4.33K	23.93	17.86
	Test	5.5K	104.1K	92.2K	10.83K	13.36K	18.93	16.76

Table 3.8: Number of sentences ($|S|$), words ($|W|$), vocabulary ($|V|$), and average sentence length ($|L|$) comprising the training, tuning and test corpora.

3.6.2 Building the Models

For fair comparison, we re-implemented three models (*i.e.,Our best model*, *Multi-Encoder* and *Cache Memory*) based on our own attention-based NMT system,[16] which incorporates dropout (Hinton et al. 2012) on the output layer and improves the attention model by feeding the most recently generated word.

For training the baseline model, we limited the source and target vocabularies to the most frequent 35K words in Chinese and English, and employ an unknown replacement post-processing technique (Jean et al. 2014, Luong et al. 2015b). We trained each model with sentences of length up to 80 words in the training data. We shuffled mini-batches as we proceed and the mini-batch size is 80. The word-embedding dimension is 620 and the hidden layer dimension is 1000. We trained for 20 epochs using (Zeiler 2012), and selected the model that yields the best performance on the validation set.

For our DNMT models, we used the same setting as baseline if applicable. The parameters of our model that are related to the standard encoder and decoder were initialized by the baseline model and were fixed in the following step. We further trained the new parameters

[16]Code repository: `https://github.com/tuzhaopeng/nmt`.

related to the cache for another 5 epochs. Again, the model that performs best on the tuning set was selected as the final model.

3.6.3 Results

Table 3.9 shows the translation performance on multiple domains with different textual styles. As seen, all DNMT models outperform the baseline system (i.e., BASE) in all cases, demonstrating the effectiveness of incorporating global context into NMT in different ways.

First of all, our proposed best model (*i.e.,* OURS) consistently outperforms the baseline system in all domains, which confirms the robustness of our approach. Especially in the news domain (*i.e.,* LDC), OURS achieves 36.52 BLEU points, and it performs best compared with other two comparable DNMT models (*i.e.,* CACHE and MULTI). CACHE and MULTI can also achieve +1.09 and +0.72 BLEU point improvements than the baseline, respectively.

Surprisingly, the CACHE approach performs the best in dialogue and speech domains (*i.e.,* TVsub and TED). We attribute the superior translation quality of this approach in the dialogue domain to the exploitation of target-side information, since most of the translations of dialogues in this domain do not preserve the syntactic structure of their original sentences at all. They are completely paraphrased in the target language and seem very hard to be improved with only source-side cross-sentence contexts. On the other hand, MULTI achieves marginal or no improvement in the dialogue domain.

Table 3.10 shows the model complexity. The CACHE model only introduces 4M additional parameters, which is small compared to both the numbers of parameters in the existing model (*i.e.,* 84.2M) and OURS (*i.e.,* 18.8M) and MULTI (i.e., 20M). CACHE is more efficient in training, which benefits from training cache-related parameters only. To minimize a waste in computation, the other models sort 20 mini-batches by their lengths before parameter updating (Bahdanau et al. 2015), while the CACHE model cannot enjoy this benefit since it depends on the hidden states of preceding sentences. Concerning decoding with additional attention models, OURS and CACHE approach do not slow down the

System	LDC		TVsub		TED		Average	
	BLEU	△	BLEU	△	BLEU	△	BLEU	△
BASE	35.39	–	32.92	–	11.69	–	26.70	–
MULTI	36.11*	+0.72	33.00	+0.08	12.46*	+0.77	27.19	+0.49
CACHE	36.48*	+1.09	**34.30***	+1.38	**12.68***	+0.99	27.82	+1.12
OURS	**36.52***	+1.13	33.34*	+0.42	12.43*	+0.74	27.43	+0.73

Table 3.9: Translation qualities on multiple domains. "*" indicates statistically significant difference ($P < 0.01$) from "BASE" , and "△" denotes relative improvement over "BASE".

System	Parameter	Speed	
		Train	Test
BASE	84.2M	1469.1	21.1
MULTI	104.2M	933.8	19.4
CACHE	88.2M	1163.9	21.1
OURS	103.0M	300.2	20.8

Table 3.10: Model complexity. "Speed" is measured in words/second for both training and testing. We employ a beam search with beam being 10 for testing.

decoding speed, while MULTI decreases decoding speed by 8.1%.

3.7 Summary

In this section, we proposed a novel approach to DNMT with complementary strategies to integrating cross-sentence context: 1) a warm-start of the encoder and decoder with global context representation, and 2) cross-sentence context serves as an auxiliary information source for updating decoder states, in which an introduced context gate plays an important role. We quantitatively and qualitatively demonstrated that the presented model significantly outperforms a strong attention-based NMT baseline system. We release the code for these experiments at `https://www.github.com/tuzhaopeng/LC-NMT`. We also systematically compare our model with two other DNMT models.

Our models benefit from larger contexts, and would be possibly further enhanced by other document-level information, such as discourse relations. We propose to study such models for full-length documents with more linguistic features in future work. We release the code for these experiments at: `https://github.com/longyuewangdcu/Cross-Sentence-NMT`.

In our future work, we expect several developments that will shed more light on utilizing long-range contexts, *i.e.,* designing novel architectures, such as employing discourse relations instead of directly using decoder states as cache values.

In the next chapter, we will improve cohesion in MT by exploring a specific discourse phenomena, dropped pronoun.

Chapter 4

Neural Dropped Pronoun Recovery and Its Application to Statistical Machine Translation

Pro-Drop is a discourse phenomenon, where certain classes of pronouns can be omitted when they are pragmatically or grammatically inferable from the context. However, it is challenging for MT models to explicitly realize DPs in the source language to the target language. In this chapter, we investigate the impact of DP recovery on translation quality especially in terms of cohesion. Aiming at SMT, we propose a NN-based DP recovery approach to alleviate the problems caused by missing pronouns. This chapter directly addresses our second research question as described below:

> **RQ 2** *How do dropped pronouns affect the performance of machine translation? Is it possible to build a robust drop pronoun recovery model for statistical machine translation?*

This chapter is organized as follows. Without loss of generality, we first introduce the motivation and background on DP translation in Section 4.1 and Section 4.2, respectively. In Section 4.3, we describe our proposed approach of recovering DPs to boost SMT. To

verifying the proposed approach, we conduct experiments on Chinese–English data in Section 4.4. Quantitative and qualitative analyses are presented in Section 4.5. To demonstrate the generality of our model, we adapt our approach to Japanese–English translation in Section 4.6, which is followed by a summary of the chapter in Section 4.7.

4.1 Introduction to Dropped Pronoun Translation

In pro-drop languages such as Chinese, pronouns can be omitted to make the sentence compact yet comprehensible when the identity of the pronouns can be inferred from the context. These omissions are not problem for humans since we can easily recall the missing pronouns from the context, but this poses difficulties for MT when translating from pro-drop languages to non-pro-drop languages (*e.g.,* English), since translation of such missing pronouns cannot normally be reproduced.

As pronouns are crucial for the syntactic structures of sentences and discourse information such as anaphora, pro-drop may not only result in missing translations of corresponding elements, but also harm the syntactic structure and even the semantic meaning of the output. As shown in Table 4.1, the SMT model fails to be aware of implicit pronouns (*i.e.,* "你们 (*you*)" and "它 (*it*)") in inputs, which resulting in poor translation outputs. Therefore, recovering DP is very significant to MT.[1]

In response to this problem, some researchers have investigated approaches for DP translation (Chung and Gildea 2010, Le Nagard and Koehn 2010, Taira et al. 2012, Xiang et al. 2013). Taira et al. (2012) explore simple rule-based and manual methods to add DP on the source side for Japanese–English translation. However, the BLEU scores of both methods are nearly identical, which indicates that only considering the single source-side inputs and forced insertion of pronouns may be less principled than tackling the problem head on by integrating them into the SMT system itself. Le Nagard and Koehn (2010) present a method to aid English pronoun translation into French by integrating an additional coref-

[1]Note that, zero/null anaphora resolution (Welo 2013) is to determine the antecedent of an implicit anaphor, which contains three steps: zero pronoun detection, anaphoricity determination and co-reference link. Whereas, DP recovery is to detect the zero pronoun position and then generate corresponding pronoun surface.

I/O	Sentences
Input	(你们) 要 不要 去 看 电影 好 啊
Reference	Do you want to go to the cinema ? okay !
Output	*you want* to see a movie . okay . yeah .
Input	(它) 根本 没 那么 严重
Reference	it is not that bad .
Output	*wasn 't* that bad .

Table 4.1: Examples of translating DPs where words in brackets are invisible in SMT decoding.

erence system. Unfortunately, the results are not convincing due to the poor performance of the coreference system in open domain (Pradhan et al. 2012). Chung and Gildea (2010) systematically examine the effects of recovering empty categories[2] in different recovery models: pattern based, conditional random fields based and parsing based models. Results show that this pipeline method can really improve the translation quality even though the automatic prediction of empty categories is not highly accurate.

We propose a novel and robust approach to recall missing pronouns and integrate them with SMT. The first challenge is that the data for training DP recovery models are very scarce. Previous work either applies manual annotation (Yang et al. 2015) or uses existing but small-scale resources such as the Penn Treebank (Chung and Gildea 2010, Xiang et al. 2013). However, it is difficult to train a robust DP recovery model using such small data for open-domain translation task. In contrast, we explore an unsupervised approach to automatically build a large-scale DP training corpus. Inspired by an initial idea that two languages are more informative than one (Dagan et al. 1991, Burkett et al. 2010), we found that parallel corpus can be used to map explicit pronouns in the target side (*i.e.,* non-pro-drop language) to the implicit pronouns in the source side (*i.e.,* pro-drop language) with the help of alignment information. To this end, we propose a simple but effective method: bidirectional search algorithm with LM scoring.

[2]This task aims to recover long-distance dependencies, discontinuous constituents and certain dropped elements (Yang and Xue 2010, Cai et al. 2011, Xue and Yang 2013). It includes trace markers, dropped pronoun, big PRO etc, while we mainly focus on dropped pronoun in our study.

After building the DP training data, we can apply various supervised approaches to build DP recovery models. We divide the task into two phases: *DP Detection* (from which position a pronoun is dropped), and *DP Prediction* (which pronoun surface/word is dropped). Due to the powerful capacity of NN, we model DP detection as sequential labelling task using RNN, and DP prediction as classification task using MLP.

Finally, we improve the translation quality by integrating the recalled DPs into SMT system in different strategies. More specifically, we extract an additional phrase table from the DP-inserted parallel corpus to produce a "pronoun-complete" translation model. In addition, we pre-process the input sentences by recalling missing pronouns via the DP generator. This makes the input sentences more consistent with the additional pronoun-complete phrase table. To alleviate the error propagation of DP generator, we feed the translation system N-best DP candidates via confusion network decoding (Rosti et al. 2007).

To validate the effect of the proposed approach, we carried out experiments on Chinese–English translation task. Experimental results on a large-scale subtitle corpus show that our approach significantly improves the baseline system by up to +1.58 BLEU points. To verify the robustness, we also adapt our approach to Japanese–English translation task.

4.2 Dropped Pronoun

Among major languages, for example, Chinese and Japanese are pro-drop languages (Huang 1984, Nakamura 1987), while English is not (Haspelmath 2001). In this section, we first review the characteristics of pronouns in English, Chinese and Japanese, respectively. We then discuss the DP phenomena in Chinese–English and Japanese–English language pairs from a bilingual point of view.

4.2.1 Pronouns in Different Languages

In English, Quirk et al. (1985) classifies the principal pronouns into three groups: personal pronouns, possessive pronouns and reflexive pronouns, defining them as central pronouns.

Category	Subject	Object	Possessive Adjective	Possessive	Reflexive
1st SG	I	me	my	mine	myself
2nd SG	you	you	your	yours	yourself
3rd SGM	he	him	his	his	himself
3rd SGF	she	her	her	hers	herself
3rd SGN	it	it	its	its	itself
1st PL	we	us	our	ours	ourselves
2nd PL	you	you	your	yours	yourselves
3rd PL	they	them	their	theirs	themselves

Table 4.2: Central pronouns in English. Abbreviations of categories: Person Type = {1st, 2nd, 3rd}, Number = {SG (singular), PL (plural)}, Gender = {M (male), F (female), N (neutral)}.

As shown in Table 4.2, all of the central pronouns have diverse forms to demonstrate or indicate different person, number, gender and function. For example, the pronoun "*we*" represents the first person in plural form and functions as subject in a sentence, while the pronoun "*him*" indicates the masculine third person in singular form and functions as a object of a verb.

Generally, Chinese pronouns correspond to the personal pronouns in English, and the Chinese pronominal system is relatively simple as there is no inflection, conjugation, or case makers (Li and Thompson 1989). Thus, there is no surface difference between subjective and objective pronouns, which are called basic pronouns. Besides, possessive and reflexive pronouns can be generated by adding some particle or modifier (*e.g.,* "的" and "们") based on the basic pronouns.

As shown in Table 4.3, the Chinese pronouns are not strictly consistent to the English pronouns. In other words, one Chinese pronoun can be mapped to several English pronouns (*i.e.,* "one-to-many" mapping). For instance, the Chinese pronoun "我" can be translated into either the subjective personal pronoun "*I*" or the objective personal pronoun "*me*", according to different contexts. Furthermore, there are also some "many-to-one" cases. For example, the pronouns "他们", "她们", "它们" can be translated into the same English

Category	Subject/Object	Possessive (+ particle "的")	Reflexive （+ word "自己"）
1st SG	我 (*I/me*)	我 的 (*my/mine*)	我 自己 (*myself*)
2nd SG	你 (*you*)	你 的 (*your/yours*)	你 自己 (*yourself*)
3rd SGM	他 (*he/him*)	他 的 (*his*)	他 自己 (*himself*)
3rd SGF	她 (*she/her*)	她 的 (*her/hers)*	她 自己 (*herself*)
3rd SGN	它 (*it*)	它 的 (*its*)	它 自己 (*itself*)
1st PL	我们 (*we/us*)	我们 的 (*our/ours*)	我们 自己 (*ourselves*)
2nd PL	你们 (*you*)	你们 的 (*your/yours*)	你们 自己 (*yourselves*)
3rd PLM	他们 (*they/them*)	他们 的 (*their/theirs*)	他们 自己 (*themselves*)
3rd PLF	她们 (*they/them*)	她们 的 (*their/theirs*)	她们 自己 (*themselves*)
3rd PLN	它们 (*they/them*)	它们 的 (*their/theirs*)	它们 自己 (*themselves*)

Table 4.3: Chinese pronouns and correspondences in English. Abbreviations of categories: Person Type = {1st, 2nd, 3rd}, Number = {SG (singular), PL (plural)}, Gender = {M (male), F (female), N (neutral)}.

pronoun "*they*", because the Chinese pronominal system considers gender for third person plural pronouns while English does not. "你们/你 - *you*" is another many-to-one example, because the English pronominal system does not differentiate between the singular and plural forms for second person pronoun while the Chinese system does.

Similar to Chinese, the Japanese pronouns can be altered to possessive and reflexive through adding the particle (*e.g.,* "の") or modifier (*e.g.,* "自 分") to the basic pronouns, respectively. Besides, the same form of pronouns in Japanese can be used to function as subject or object with different particles. For example, the particle "は" comes after the subjective pronouns, while the particle "を" occurs after the objective pronouns.

In Table 4.4, we only list the most commonly used forms of subjective and objective pronouns, because possessive and reflexive pronouns can be generated by adding corresponding particles. Different from English and Chinese, Japanese has a large number of pronoun variations, which are borrowed in archaism. The Japanese pronominal system considers more factors such as gender, age, and relative social status of the speaker and audience. For instance, the first person singular pronoun "私" is used in formal situations,

Category	Subject/Object
1st SG	私, 我, 俺, 僕, 儂, 家, etc. (*I/me*)
2nd SG	お前, おまえ, なん, 君, 貴方, あなた, あんた, 貴様, etc. (*you*)
3rd SGM	そいつ, あいつ,あの人, あの方, 彼, etc. (*he/him*)
3rd SGF	そいつ, あいつ, あの人, あの方, 彼女, etc. (*she/her*)
3rd SGN	そいつ. (*it*)
1st PL	我々, 我等, etc. (*we/us*)
2nd PL	お前, おまえ, なん, 君, 貴方, あなた, あんた, 貴様, etc. (*you*)
3rd PL	彼等 (*they/them*)

Table 4.4: Commonly-used Japanese pronouns and correspondences in English. Abbreviations of categories: Person Type = {1st, 2nd, 3rd}, Number = {SG (singular), PL (plural)}, Gender = {M (male), F (female), N (neutral)}.

while "僕" and "俺" refer to male pronouns and are normally used in informal contexts. Besides, "儂" is mostly used in old Japanese society or to indicate old male characters, while "家" is frequently used by young girls.

4.2.2 Dropped Pronoun in Translation

When translating from pro-drop to non-pro-drop languages, pronouns are frequently dropped on the source side but should be retained on the target side. Figure 4.1 illustrates the DP phenomenon in between pro-drop and non-pro-drop languages. As shown in Chinese–English sentence pairs (*i.e.,Sentence 1–2*), the subject pronouns "你 (*you*)", "我 (*I*)" and the object pronouns "它 (*it*)", "你 (*you*)" are all omitted on the Chinese side. In Japanese–English examples (*i.e.,Sentences 3–4*), the subject pronouns "あなた (*you*)", "私 (*I*)" and the object pronouns "それ (*it*)" with their corresponding particles (*e.g.,* "を", "は") are also omitted on the Japanese side.

We validate this finding by analyzing large Chinese–English parallel corpus, which consist of sentence pairs extracted from movie and TV episode subtitles. As shown in Figure 4.2, in around one million Chinese–English sentence pairs, there are 6.5 million Chinese pronouns while there are 9.4 million English pronouns, which shows that more

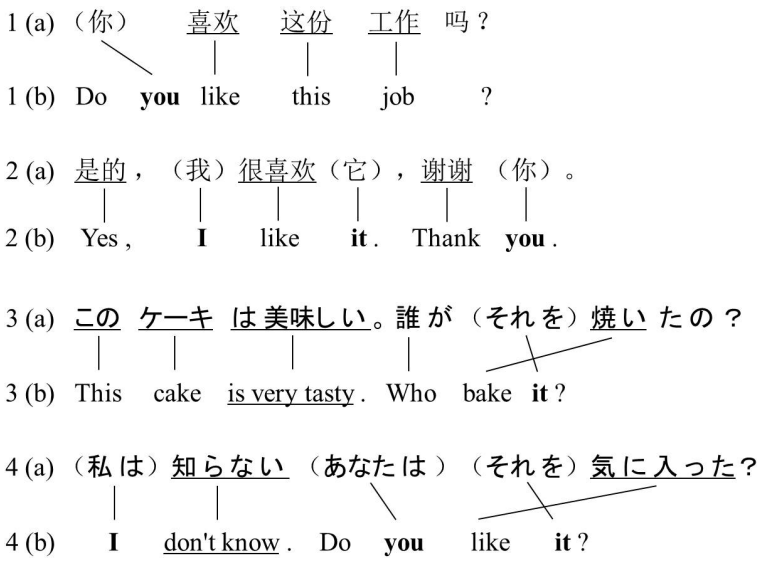


Figure 4.1: Examples of dropped pronouns in Chinese–English (*i.e.,Sentence 1–2*) and Japanese–English (*i.e.,Sentence 3–4*) parallel corpora. The pronouns in the brackets are omitted.

than 2.9 million Chinese pronouns are relatively omitted.

Besides, the extent of pro-drop in different domains or genres are different (Yang et al. 2015). We analyzed two large Chinese–English corpora in newswire and dialogue domains, respectively.[3] As shown in Table 4.5, around 26.55% of English pronouns are dropped in the dialogue domain, while only 7.35% of pronouns are dropped in the newswire domain. It shows that the pro-drop phenomenon is more prevalent in informal genres such as dialogues than formal genres. And the most frequently DPs in newswire are the dummy pronoun "它 (it)" (Baran et al. 2012), which can be recovered by baseline MT model and may not be crucial to translation performance in terms of BLEU score. This high proportion within informal genres shows the importance of addressing the challenge of translation of dropped pronouns, thus we verify our approaches with respect to the dialogue domain.

[3]The *Dialogue* corpus consists of subtitles extracted from movie subtitle websites; The *Newswire* corpus is available at China Workshop on Machine Translation (CWMT).

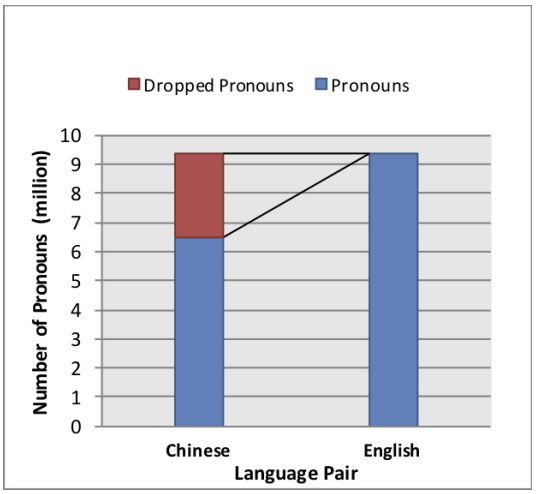


Figure 4.2: Statistics of dropped pronouns in Chinese–English (left) and Japanese–English (right) parallel corpora in movie subtitle domain.

Genres	# Sentences	# Chinese Pronouns	# English Pronouns	# DPs
Dialogue	2.15M	1.66M	2.26M	26.55%
Newswire	3.29M	2.27M	2.45M	7.35%

Table 4.5: Statistics of pronouns in different genres.

4.3 Dropped Pronoun Generation and Translation

The framework of our proposed approach is shown in Figure 4.3, which contains three main components: *DP training data annotation*, *DP generation*, and *SMT integration*. Given a parallel corpus, we first automatically annotate with DPs the source side by mapping aligned pronouns from the target side. With the auto-annotated DP training corpus, we then build a model to recover DPs for source side sentences. Finally, we integrate the DP generator into SMT with different strategies. In the following sub-sections, we introduce each component in detail.

4.3.1 Dropped Pronoun Training Corpus Construction

Given a parallel corpus, we employ an unsupervised word alignment method (Och and Ney 2003) to produce word alignment matrix for each sentence pair. From observing the alignment matrix, as shown in Figure 4.4, we found that there exists a diagonal line based on

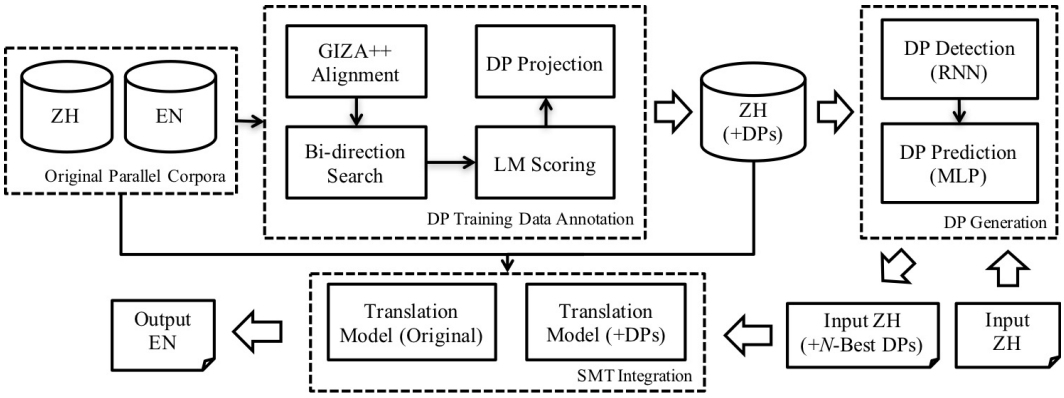


Figure 4.3: Architecture of our proposed approach (taking Chinese-to-English translation for example).

aligned blocks and it is possible to predict Dropped Pronoun Position (DPP) on the source side according to the heuristic rule.

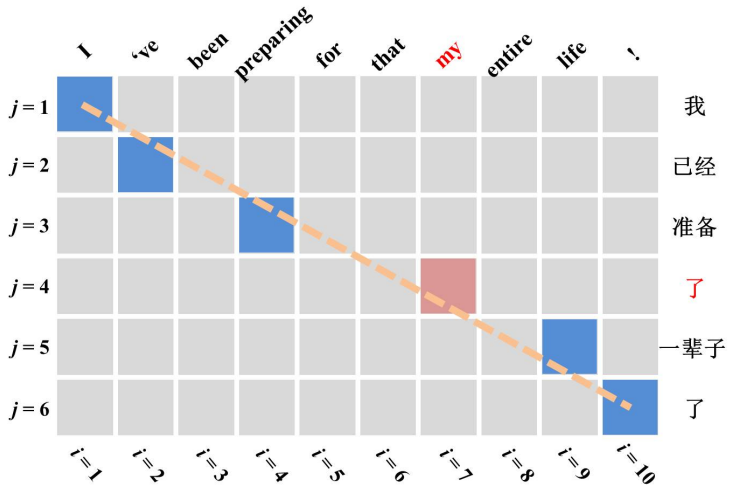


Figure 4.4: Example of DP annotation using word alignment matrix. Blue blocks represent already aligned words between source side and target side, while Red block represents predicted alignment.

Accordingly, we propose a bidirectional search algorithm as shown in Algorithm 1. Given the alignment matrix *Matrix* and the misaligned pronoun position *Misalign*, the algorithm searches from *Misalign* to the beginning and the end of the target sentence, respectively. If one word in the target language is aligned with one word in the source language, we call them aligned words (the value is set as 1), otherwise they are considered to be misaligned words (the value is set as 0). The algorithm tries to find the nearest preceding and

following aligned words around *Misalign*, and then to project them to the DPPs (*start* or *end*) on the source side.

Algorithm 1: Bidirectional search algorithm in MATLAB™

```
function [DP_start, DP_end] = Bi-Search(Matrix, Misalign)
   row = sum(Matrix, 1);
   row_true = find(row == 1);
   left_side = row_true(row_true < Misalign);
   DP_start = find(Matrix(:, left_side(end)) == 1);
   right_side = row_true(row_true > Misalign);
   DP_end = find(Matrix(:, right_side(1)) == 1);
end
```

We use the Chinese–English example (in Figure 4.4) to further illustrate how to annotate the DP "我的" in Chinese sentence. We consider the alignments as a binary $I \times J$ matrix with the cell block at position (i, j), to decide whether an alignment exists between a Chinese word at position i and an English word at position j. For each pronoun on the English side (*i.e.,* "*I*", "*my*"), we check whether it has an aligned pronoun on the Chinese side. Once there is a pronoun such as "*my*" (*i.e.,* $i = 7$) has no alignment, we hypothesize this English pronoun possibly corresponds to a DP (marked as DP_{MY}). We then determine the possible positions of DP_{MY} on the Chinese side (an approximate area, *i.e.,* red block) by considering the preceding and following alignment blocks (*i.e.,* "*preparing*-准备" ($i = 4, j = 3$) and "*life*-一辈子" ($i = 9, j = 5$)) along the diagonal line. After that, there are still two possible positions to insert DP_{MY} (*i.e.,* the two gaps before or after the Chinese word "了"). To further determine the exact DPP, we generate possible candidate sentences by inserting the Chinese translation of DP_{MY} into all possible positions (*i.e.,* "我 已经 准备 **我的** 了 一辈子 了" and "我 已经 准备 了 **我的** 一辈子 了"). We employ an n-gram LM to score these candidates and select the one with the lowest perplexity as the final result. Finally, a large DP training corpus is automatically built by utilizing parallel data.

Note that, the Chinese equivalent of DP_{MY} can be directly translated according to

Table 4.3. Some English pronouns may correspond to more than one Chinese pronouns, such as "*they* - 他们 / 她们 / 它们". In this case, we consider all the possible Chinese pronouns as the candidates. As the amount and type of DPs vary in different genres (as shown in Table 4.5), we train the LM on a large monolingual data in newswire domain (detailed in Section 4.4). In order to reduce the problem of incorrect DP insertion caused by incorrect alignments, we use a large amount of additional parallel corpus to improve the quality of word alignment.

4.3.2 Dropped Pronoun Generation

After building the DP training data, we can apply various supervised approaches to build DP recovery models. In light of the recent success of applying deep neural network technologies in natural language modelling (Raymond and Riccardi 2007, Mesnil et al. 2013), we propose a NN based DP generation approach in two phases: 1) we first employ an RNN model to predict the DPP; and 2) then train a classifier with MLP to predict the Dropped Pronoun Surface (DPS).

Dropped Pronoun Position Detection This task is to label each word if there is a pronoun dropped before this word, which can intuitively be regarded as a sequence labelling problem. We expect the output to be a sequence of labels $y^{(1:n)} = (y^{(1)}, y^{(2)}, \cdots, y^{(t)}, \cdots, y^{(n)})$ given a sentence consisting of words $w^{(1:n)} = (w^{(1)}, w^{(2)}, \cdots, w^{(t)}, \cdots, w^{(n)})$, where $y^{(t)}$ is the label of word $w^{(t)}$. In our task, there are binary labels L = {*NA*, *DP*} (corresponding to non-pro-drop or pro-drop pronouns), thus $y^{(t)} \in L$.

Given an input word $w^{(t)}$, we produce an embedding representation (Mikolov et al. 2013a) $\mathbf{v}^{(t)} \in \mathbb{R}^d$ where d is the dimension of the representation vectors. In order to capture short-term temporal dependencies, we employ a context window to ordered concatenation of word embedding vectors (Mesnil et al. 2013), as in Equation 4.1:

$$\mathbf{x_d}^{(t)} = \mathbf{v}^{(t-k)} \oplus \cdots \oplus \mathbf{v}^{(t)} \oplus \cdots \oplus \mathbf{v}^{(t+k)} \tag{4.1}$$

where k is the context window size.

We feed RNN unit with the concatenated word embeddings vector $\mathbf{x_d}^{(t)}$ to learn the dependency of sentences, which can be formulated as Equation 4.2:

$$\mathbf{h}^{(t)} = f(\mathbf{U}\mathbf{x_d}^{(t)} + \mathbf{V}\mathbf{h}^{(t-1)}) \tag{4.2}$$

where $f(x)$ is a sigmoid function at the hidden layer. $\mathbf{U}$ is the weight matrix between the input and the hidden nodes, and $\mathbf{V}$ is the weight matrix between the context nodes and the hidden nodes. At the output layer, a softmax function is adopted for labelling, as in Equation 4.3:

$$y^{(t)} = g(\mathbf{W_d}\mathbf{h}^{(t)}) \tag{4.3}$$

where $g(z_m) = \frac{e^{z_m}}{\sum_k e^{z_k}}$, and $\mathbf{W_d}$ is the output weight matrix.

Dropped Pronoun Surface Prediction Once the DP position is detected, the next step is to determine the exact pronoun surface. Accordingly, we train an m-class classifier (*i.e.,* $m = 20$ in our experiments), where each class refers to a distinct pronoun as described in Section 4.2.1.

We employ a number of features based on previous work (Xiang et al. 2013, Yang et al. 2015). As shown in Table 4.6, we extract lexical, contextual and syntax feature sets. For lexical features (*i.e.,* Row 1–4), we extract words, Part-Of-Speech (POS) tags and pronouns around the DPP. About the larger-context feature set (*i.e.,* Row 5–8), we consider the pronouns and nouns in *N* preceding or following sentences. In order to model the syntax features (*i.e.,* Row 9–10), we retrive the syntax tree, and combined tags of the sub-tree nodes from *DPP* / *DPP*-1 to the root. Note that we only extract the non-pro-drop pronouns in Row 3–6. Each unique feature is treated as a word, and assigned a "word embedding". The embeddings of the features are then fed into an MLP. We fix the number of features for the variable-length features, where missing ones are tagged as *None*. Accordingly, all training instances share the same feature length. To pre-process the training data, we select

all instances that contain DP from the original DP corpus. During testing time, *DPP* is given by our DPP detection model.

#	Feature Set	Description
1	Lexical	M surrounding words around *DPP*
2		M surrounding POS tags around *DPP*
3		pronouns before *DPP* in the current sentence
4		pronouns after *DPP* in the current sentence
5	Larger Context	pronouns in preceding N sentences
6		pronouns in following N sentences
7		nouns in preceding N sentences
8		nouns in following N sentences
9	Syntax	path from *DPP* to the root in the syntax tree
10		path from *DPP*-1 to the root in the syntax tree

Table 4.6: List of features. *DPP* is the DP position, *M* is the word-level window size surrounding *DPP*, and *N* as the sentence-level window size surrounding current sentence (*i.e.,* the one contains *DPP*).

We employ a feed-forward neural network with four layers. The input $\mathbf{x_p}$ comprises the embeddings of the set of all possible feature indicator names. The middle two layers $\mathbf{a}^{(1)}$, $\mathbf{a}^{(2)}$ use Rectified Linear function R as the activation function, as in Equation 4.4 and 4.5:

$$\mathbf{a}^{(1)} = R(\mathbf{W_p}^{(1)}\mathbf{x_p} + \mathbf{b}^{(1)}) \tag{4.4}$$

$$\mathbf{a}^{(2)} = R(\mathbf{W_p}^{(2)}\mathbf{a}^{(1)} + \mathbf{b}^{(2)}) \tag{4.5}$$

where $\mathbf{W_p}^{(1)}$ and $\mathbf{b}^{(1)}$ are the weights and bias connecting the first hidden layer to second hidden layer; and so on. The last layer $\mathbf{y_p}$ adopts the softmax function $g(\cdot)$, as in Equation 4.6:

$$\mathbf{y_p} = g(\mathbf{W_p}^{(3)}\mathbf{a}^{(2)}) \tag{4.6}$$

4.3.3 Integration into Machine Translation

The integration into SMT is three folds: 1) using DP-inserted parallel corpus to train an additional translation model; 2) generating DP for input sentences at decoding time; and 3) generating N-best DP lattice for input at decoding time.

DP-Enhanced Translation Model We train an additional translation model (*i.e.,TM+DP*) on the new parallel corpus, whose source side is inserted with DPs derived from the target side via the alignment matrix (as described in Section 4.3.1). We hypothesize that inserting DPs in training data can help to obtain a better alignment, which can benefit translation. The whole translation process is based on the boosted translation model, *i.e.,* with DPs inserted. As far as translation model combination is concerned, we directly feed SMT the multiple phrase tables. The gain from the additional translation model is mainly from complementary information about the recalled DPs from the annotated data.

DP-Generated Input Another integration strategy is to pre-process the input sentence by inserting possible DPs with the generator (detailed in Section 4.3.2) so that the DP-inserted input (*i.e.,Input+DP*) is translated. The recovered DPs would be explicitly translated into the target language, so that the possibly missing pronouns in the translation might be recalled. This makes the input sentences and DP-enhanced translation model more consistent in terms of recalling DPs.

N-best DP-Generated Input The *DP-Generated Input* method suffers from a major drawback: it transfers the 1-best DP generation result to decoding, which potentially introduces translation mistakes due to the propagation of generation errors. To alleviate this problem, an obvious solution is to offer more DP alternatives. Related studies have shown that SMT systems can benefit from widening the annotation pipeline (Liu et al. 2009, Tu et al. 2010, 2011, Liu et al. 2013). In the same direction, we propose to feed the decoder N-best DP candidates, which allow the SMT to arbitrate between multiple ambiguous hypotheses from upstream processing so that the best translation can be produced. The general method is to

make the input with N-best DPs into a confusion network. In our experiment, we use the Moses confusion network decoding (Rosti et al. 2007) and each prediction result in the N-best list is assigned a weight of $1/N$.

4.4 Experiments

In this section, we describe the data, model setup and results on experiments of our proposed models.

4.4.1 Data

Experiments evaluate the method for translation of Chinese–English subtitles. About training data, more than one million sentence pairs were extracted from movie and TV episode subtitles.[4] We randomly select two complete television episodes as the tuning set, and another two episodes as the test set. Note that all sentences maintain their contextual information at the discourse level, which can be used for feature extraction in Section 4.3.2.

We pre-processed the extracted subtitles using our in-house scripts (Wang et al. 2016), including sentence boundary detection and bilingual sentence alignment etc. In particular, we employ Jieba toolkit[5] for Chinese word segmentation, and Moses toolkit[6] for English word tokenization. The statistics of our data are listed in Table 4.7. As seen, sentences in subtitle domain are generally short and the Chinese side, as expected, contains many examples of DP.

To obtain high-quality DP annotations (detailed in Section 4.3.1) from parallel corpus, we first enlarge the original parallel data with 9 million OpenSubtitles2016 (Lison and Tiedemann 2016)[7] for building word alignments. Secondly, we also use a large monolingual corpus[8] in formal genre (as discussed in 4.2.2) for LM scoring (detailed in Section 4.3.3).

[4] Subtitle websites: `http://www.opensubtitles.org` and `http://weisheshou.com`.
[5] Available at `https://github.com/fxsjy/jieba`.
[6] Available at `http://www.statmt.org/moses`.
[7] Available at `http://opus.nlpl.eu/OpenSubtitles2016.php`.
[8] Sogou Chinese News Collection Corpus: `http://www.sogou.com/labs/dl/ca.html`.

Data	$\|S\|$	$\|W\|$		$\|P\|$		$\|V\|$		$\|L\|$	
		Zh	En	Zh	En	Zh	En	Zh	En
Train	1.04M	6.15M	8.18M	0.60M	0.82M	0.10M	76.64K	5.91	7.87
Tune	1,086	6.66K	9.19K	0.76K	1.03K	1.74K	1.41K	6.13	8.46
Test	1,154	6.71K	9.43K	0.76K	0.96K	1.79K	1.42K	5.81	8.17

Table 4.7: Number of sentences ($|S|$), words ($|W|$), pronouns ($|P|$), vocabulary ($|V|$), and averaged sentence length ($|L|$) comprising the training, tuning and test corpora. K stands for thousands and M for millions.

Besides, in translation task, we only use the target side of original parallel subtitle corpus for LM.

4.4.2 Model Setup

We carry out our experiments using the PBSMT model in Moses (Koehn et al. 2007). Furthermore, we train 5-gram language models using the SRI Language Toolkit (Stolcke 2002). We run GIZA++ (Och and Ney 2003) on parallel to obtain word alignment. As the DP annotation method relies on the quality of alignment, we employ "intersection" alignment strategy, which has higher precision, but lower recall. We use Minimum Error Rate Training (MERT) (Och 2003) to optimize the feature weights.

The NN models are implemented using the neural network library, Theano (Bergstra et al. 2010). We build DP position detector using RNN with the following settings: context window = 5, the size of hidden layer = 200, embedding size = 200, iterations = 10. We train the model in 10 epochs. The DP classifier is built on MLP with the following settings: hidden layer size = 200, embedding size = 100, iterations = 200. Both models are trained with randomly initialized embeddings.

4.4.3 Results

We report the results of DP annotation, DP generation and DP translation. For MT evaluation, we used case-insensitive 4-gram BLEU (Papineni et al. 2002) and *sign-test* (Collins

Data	Detection	Prediction
Tune	0.94	0.92
Test	0.95	0.92

Table 4.8: Evaluation of DP annotation method on tuning and test sets.

et al. 2005) to test for statistical significance. We also used micro-averaged F-score (Powers 2011) to measure DP generation quality.

Dropped Pronoun Annotation We follow the annotation method (detailed in Section 4.3.1) to automatically label DPs in training/tuning/test set. In order to check whether the annotation method is reasonable, we also manually label DP on the source side of tuning/test set according to the pronouns on the target side. To this end, the results are shown in Table 4.8. The agreement between automatic labels and manual labels on DP detection are 94% and 95% on tuning and test sets and 92% and 92% on DP prediction, respectively. Since sentence structures in Chinese and English are mainly consistent (*i.e.,* Subject-Verb-Object (SVO)), our method can easily achieve above 90% accuracy indicate that it is trustworthy for further steps.

Dropped Pronoun Generation We then built the DP generator according to Section 4.3.2 and measure the accuracy (in terms of words) of the proposed models in two phases: 1) *DPP Detection* shows the performance of our RNN based DP position detection. We consider the tags for each word (*i.e.,* pro-drop or non-pro-drop before the each word), without considering the exact pronoun word; 2) *DPS Prediction* shows the performance of the MLP based classifier in determining the exact DP surface based on detection. Thus, we measure accuracies of both detected positions and predicted pronouns.

Table 4.9 lists the results of the DP generator. The F-score of *DP Detection* achieves 88% and 86% on the Tuning and Test sets, respectively. However, it has lower F-scores of 66% and 65% for the *DP Prediction* on the Tuning and Test sets, respectively. This indicates that generating the exact DP words is really a difficult task. Considering that the

Task	Data	Precision	Recall	F-score
DPP Detection	Tune	0.88	0.84	0.86
	Test	0.88	0.87	0.88
DPS Prediction	Tune	0.67	0.63	0.65
	Test	0.67	0.65	0.66

Table 4.9: Evaluation of DP generation approach on tuning and test sets.

DP generation is not highly accurate, we propose to recall N-best DP candidates to alleviate error propagation problem.

Dropped Pronoun Translation According to Section 4.3.3, we integrate DP generation into SMT and evaluate translation quality. Table 4.10 summaries the results of translation performance with different integration strategies. Clearly all the proposed models (Rows 2-8) significantly outperform the baseline in all cases, although there are considerable differences among different variations. "Baseline" (Row 1) uses the original input to feed the SMT system. "+DP-ins. TM" (Row 2) denotes using an additional translation model trained on the DP-inserted training corpus, while "+DP-gen. Input N" (Rows 4-8) denotes further completing the input sentences with the N-best pronouns generated from the DP generator. "Oracle" (Rows 9-10) uses the input with manual ("Manual") or automatic ("Auto") insertion of DPs by considering the target set. Taking "Auto Oracle" for example, we annotate the DPs via alignment information (supposing the reference is available) using the technique described in Section 4.3.1.

The baseline system uses the parallel corpus and input sentences without inserting/generating DPs. It achieves 20.06 and 18.76 in BLEU score on the development and test data, respectively. The BLEU scores are relatively low because 1) we have only one reference, and 2) dialogue machine translation is still a challenge for the current SMT approaches. By using DP-enhanced translation model, we improve the performance consistently on both development (*i.e.,* +0.26) and test data (*i.e.,* +0.61). This indicates that the inserted DPs are really helpful for SMT. Thus, the gain in the "+DP-ins TM" is mainly from the improved

#	Systems	Dev Set	Test set
1	Baseline	20.06	18.76
2	+DP-ins. TM	20.32 (+0.26)	19.37 (+0.61)
3	+DP-gen. Input		
4	1-best	20.49 (+0.43)	19.50 (+0.74)
5	2-best	20.15 (+0.09)	18.89 (+0.13)
6	4-best	20.64 (+0.58)	19.68 (+0.92)
7	6-best	21.61 (+1.55)	20.34 (+1.58)
8	8-best	20.94 (+0.88)	19.83 (+1.07)
9	Manual Oracle	24.27 (+4.21)	22.98 (+4.22)
10	Auto Oracle	23.10 (+3.04)	21.93 (+3.17)

Table 4.10: Evaluation of DP translation quality.

alignment quality. We can further improve translation performance by completing the input sentences with our DP generation model. We test N-best DP insertion to examine the performance, where $N = \{1, 2, 4, 6, 8\}$. Working together with "DP-ins. TM", 1-best generated input already achieves +0.43 and +0.74 BLEU score improvements on development and test set, respectively. The consistency between the input sentences and the DP-inserted parallel corpus contributes most to these further improvements. As N increases, the BLEU score grows, peaking at 21.61 and 20.34 BLEU points when N=6. Thus, we achieve a final improvement of +1.55 and +1.58 BLEU points on the development and test data, respectively. However, when adding more DP candidates, the BLEU score decreases by 0.97 and 0.51. The reason for this may be that more DP candidates add more noise, which harms the translation quality. The oracle system uses the input sentences with manually annotated DPs rather than "DP-gen. Input". The performance gap between "Oracle" and "+DP-gen. Input" shows that there is still a large space for further improvement for the DP generation model.

4.5 Analysis

In this section, we first select sample sentences to further investigate the effect of DP generation on translation.

In the following sentences, we show a positive case (Case A), a negative case (Case B) and a neutral case (Case C) of translation by using DP insertion (i.e. "+DP-gen. Input 1-best") as well as N-best case (Case D) (i.e. "+DP-gen. Input N-best"). In Cases A-C, we give (a) the original Chinese sentence and its translation generated by the baseline system, (b) the DP-inserted Chinese sentence and its translation generated by "+DP-gen. Input 1-best" system, and (c) the reference English sentence. In Case D, (a) is the original Chinese sentence and its translation, and (b)-(d) are N-best DP-generated Chinese sentences and their MT outputs, and (e) is the reference.

In Case A (in Figure 4.5), the output of (a) (generated by the original Chinese sentence) is incomplete because it is missing a subject on the English side. However, by adding a DP "你 (you)" via our DP generator, "*Do you*" is produced in the output of (b). It not only gives a better translation than (a), but also makes the output a formal general question sentence. We found that inserting DPs into interrogative sentences helps both reordering and grammar. Generally, Case A shows that 1-best DP generation can really help translation.

Figure 4.5: Positive effect of DP generation on translation.

In Case B in Figure 4.6, however, our DP generator mistakenly regards the simple sentence as a compound sentence and inserts the wrong pronoun "我 (I)" in (b), which causes an incorrect translation output (worse than (a)). This indicates that we need a highly

accurate source-sentence parse tree for more correct detection of the antecedent of DPs. Besides, some errors are caused by pre-processing such as Chinese segmentation and part-of-speech (POS) tagging. For instance, a well-tagged sentence should be "他/PN 好/VA 有/VE 魅力/NN (He has a good charm)". However, in our experiments, the sentence is incorrectly tagged as "他/PN 好/VA 有魅力/VE" and the DP generator inserts a DP "我 (I)" between "好" and "有魅力". Therefore, our features should be extracted based on a natural language processing toolkit with good performance.

Case B

(a) 你 肯定 看过 那 电视剧 。
You must have seen that show .

(b) 你 肯定 ⟨**我**⟩ 看过 那 电视剧 。
You are sure ⟨**I**⟩ 've seen that show .

(c) You must have seen that show .

Figure 4.6: Negative effect of DP generation on translation.

In Case C (in Figure 4.7), the translation results are the same in (a) and (b). Such unchanged cases often occur in "fixed" linguistic chunks such as preposition phrases ("on *my* way"), greetings ("see *you* later" , "thank *you*") and interjections ("*my* God"). However, the alignment of (b) is better than that of (a) in this case. It also shows that even though the DP is inserted in a wrong place, it can still be reordered into the correct translation due to the powerful target LM. This explains why end-to-end performance can be improved even with a sub-optimal DP generator.

In Case D (in Figure 4.7), (a) is the original Chinese sentence and its translation; (b) is the 1-best DP-generated Chinese sentence and its MT output; (c) stands for 2-best, 4-best and 6-best DP-generated Chinese sentences and their MT outputs (which are all the same); (d) is the 8-best DP-generated Chinese sentence and its MT output; (e) is the reference. The N-best DP candidate list is "我 (I)", "你 (You)", "他 (He)", "我们 (We)", "他们 (They)", "你们 (You)", "它 (It)" and "她 (She)". In (b), when integrating an incorrect 1-best DP

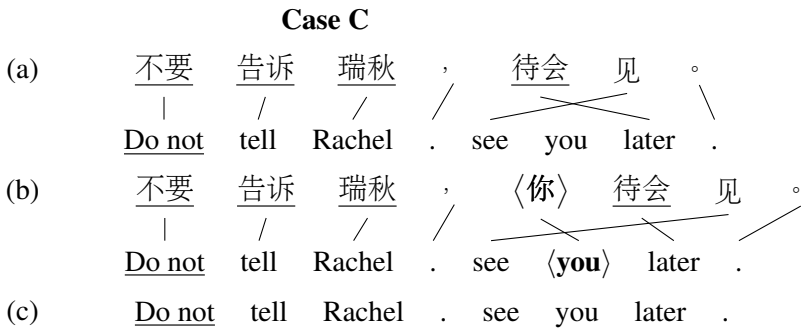


Figure 4.7: Neutral effect of DP generation on translation.

into MT, we obtain the wrong translation. When considering more DPs (2-/4-/6-best) in (c), the SMT system generates a correct translation by weighting the DP candidates during decoding. When further increasing N (8-best), (d) shows a wrong translation again due to increased noise.

Figure 4.8: Effects of N-best DP generation on translation.

4.6 Adaption to Japanese–English Translation

In this section, we adapt our approach for Japanese–English translation task.

#	Systems	Dev Set	Test set
1	Baseline	18.24	16.54
2	+DP-ins. TM	18.58 (+0.34)	16.86 (+0.32)
3	+DP-gen. Input		
4	1-best	18.54 (+0.30)	16.79 (+0.25)
5	2-best	18.79 (+0.55)	17.08 (+0.54)
6	4-best	19.32 (+1.08)	17.50 (+0.96)
7	6-best	19.11 (+0.87)	17.41 (+0.87)
8	8-best	18.84 (+0.60)	17.11 (+0.57)
9	Manual Oracle	20.78 (+2.54)	18.84 (+2.30)
10	Auto Oracle	20.06 (+1.82)	18.31 (+1.77)

Table 4.11: Evaluation of Japanese–English DP translation quality.

4.6.1 Experiment Setup

For Japanese–English training data, we extract 0.5 million sentence pairs from OpenSubtitles2016[9]. The LM for DP annotation is trained on combined data[10]. All models are same as ones used for Chinese–English translation task.

4.6.2 Results

The agreements between automatic labels and manual labels on DP prediction are around 80%, which relatively lower than Chinese–English corpus. The main reason is that Japanese is a Subject-Object-Verb (SOV) language while English is in SVO order. It is difficult for bidirectional search algorithm on distinct language pairs. About Japanese DP generation, "DPP Detection" achieves 81% and 80% F1 scores on the Tuning and Test sets, respectively, while "DPS Predition" just obtains 59% and 58%, respectively.

Table 4.11 shows the DP translation performance. As the training data are smaller, the "Baseline" system achieves 18.24 and 16.54 in BLEU score on the tuning and test sets,

[9]We use part of OpenSubtitles2016 corpus, which is available at `http://opus.lingfil.uu.se/OpenSubtitles2016.php`.

[10]We collect a number of monolingual corpora: KFTT (`http://www.phontron.com/kftt`), NTCIR (`http://warehouse.ntcir.nii.ac.jp/openaccess/rite/10RITE-Japanese-wiki.html`) and Wikipedia XML Corpus (`http://www-connex.lip6.fr/~denoyer/wikipediaXML`).

respectively. The best BLEU scores are 19.32 (+1.08) and 17.50 (+0.96) on tuning and test set when N=4. The improvement is relatively lower because it is more difficult to recover DPs in Japnese.

4.7 Summary

In this section, we have presented a novel approach to recall missing pronouns for machine translation from a pro-drop language to a non-pro-drop language. We first propose an automatic approach to DP annotation, which utilizes alignment matrix from parallel data and shows high consistency compared with the manual annotation method. We then applied neural networks to DP detection and prediction tasks with rich features. About integration into translation, we employ confusion networks decoding with N-best DP prediction results instead of ponderously inserting only 1-best DP into input sentences. Finally we implemented above models into a well designed DP translation architecture.

Experiments on both Chinese–English and Japanese-English translation tasks show that it is crucial to identify DPs to improve the overall translation performance. Our analysis shows that insertion of DPs affects the translation to a large extent.

Our main findings in this section are fourfold:

- Bilingual information can help to build monolingual models without any manually annotated training data for DP recovery task;
- Benefiting from representation learning, neural network-based models work well without complex feature engineering work for DP recovery task;
- N-best DP integration works better than 1-best DP insertion;
- Our approach is robust and can be applied on pro-drop languages especially for Chinese.

Chapter 5

Dropped Pronoun Reconstruction for Neural Machine Translation

As discussed in the last chapter, pro-drop leads to significant problems in conventional MT. Previous research has investigated DP translation for SMT and obtained promising results (Chung and Gildea 2010, Taira et al. 2012). Inspired by these previous successes, in this chapter, we investigate DP translation for the state-of-the-art NMT. It is an early attempt to learn to tackle DP translation for NMT models. This chapter directly answers our third research question as described below:

> **RQ 3** *Does neural machine translation still suffer from dropped pronoun problems? If so, how should we embed DP information into neural network models?*

This chapter is organized as follows. We first introduce the motivation of DP translation on NMT in Section 5.1. In Section 5.2, we describe our novel reconstruction-based approach to alleviating DP translation problems for NMT models. We conduct experiments on Chinese–English dialogue translation and show the results in Section 5.3. We quantitatively and qualitatively demonstrated that the presented model significantly outperforms a strong NMT baseline system in Section 5.4. We demostrate the the reliability and ro-

I/O	Sentences
Input	(它) 根本 没 那么 严重
Ref	**It** is not that bad
SMT	Wasn 't that bad
NMT	**It** 's not that bad
Input	这块 面包 很 美味 ! 你 烤 的 (它) 吗 ?
Ref	The bread is very tasty ! Did you bake **it** ?
SMT	This bread , delicious ! Did you bake ?
NMT	The bread is delicious ! Are you baked ?

Table 5.1: Examples of when our strong baseline NMT system fails to accurately translate DPs. Words in brackets are DPs that are invisible in decoding.

bustness of our model compared to others, and adapt the approach to Japanese–English translation task in Section 5.5, which is followed by the chapter summary in Section 5.6.

5.1 Why Dropped Pronoun Neural Translation?

As discussed in Section 4.1, pronouns are frequently omitted in pro-drop languages, generally leading to significant challenges with respect to the production of complete translations. Furthermore, this problem is especially severe in informal genres such as dialogues and conversation, where pronouns are more frequently omitted to make utterances more compact (Yang et al. 2015).

Researchers have investigated methods of alleviating the DP problem for conventional SMT models showing promising results (Le Nagard and Koehn 2010, Xiang et al. 2013). In addition to their papers, we proposed NN-based DP recovery model to boost SMT models in Chapter 4. Modeling DP translation for the more advanced NMT models, however, has received no attention, resulting in low performance in this respect even for state-of-the-art approaches.

Due to the ability to capture semantic information with distributed representations, ideally, the hidden states (either encoder-side or decoder-side) of NMT should embed the

System	Baseline	Oracle	△
SMT	30.16	35.26	+5.10
NMT	31.80	36.73	+4.93

Table 5.2: Translation performance improvement ("△") with manually annotated DPs ("Oracle"). "Oracle" uses the input with manual annotation of DPs by considering the reference.

missing DP information by learning the alignments between bilingual pronouns from the training corpus. In practice, however, NMT models only manage to successfully translate some simple DPs, but still fail when translating anything more complex. As shown in Table 5.1, the NMT model succeeds in translating the simple dummy pronoun (upper panel), while it fails on a more complicated one (bottom panel); SMT fails on both cases. We also conducted a preliminary experiment to exploit the upper bound and lower bound of performance on DP translation. About "Oracle" (upper bound) setting, we manually annotated DPs in input source sentences by considering the reference. For "Baseline" (lower bound) setting, there is no pre-processing for input. We show empirical results in Table 5.2 with the following two observations: 1) NMT indeed outperforms SMT when translating pro-drop languages; and 2) the performance of the NMT model can increase further by improving the translation of DPs. Finally, we narrow the gap between correct DP translation for NMT models to improve translation quality for pro-drop languages with advanced models.

More specifically, we propose a novel reconstruction-based approach to alleviate DP problems for NMT. Firstly, we explicitly and automatically annotate DPs for each source sentence in the training corpus using alignment information from the parallel corpus. Accordingly, each training instance is represented as a triple ($\mathbf{x}$, $\mathbf{y}$, $\hat{\mathbf{x}}$), where $\mathbf{x}$ and $\mathbf{y}$ are source and target sentences, and $\hat{\mathbf{x}}$ is the annotated source sentence. Next, we apply a standard encoder-decoder NMT model to translate $\mathbf{x}$, and obtain two sequences of hidden states from both the encoder and decoder. This is followed by introducing an additional *reconstructor* (Tu et al. 2017b) to reconstruct the annotated source sentence $\hat{\mathbf{x}}$ with hidden states from either the encoder or decoder, or both components. With auxiliary training objec-

tives, in terms of reconstruction scores, the parameters associated with the NMT model are guided to produce enhanced hidden representations that are encouraged as much as possible to embed annotated DP information.

Reconstruction is a standard concept in auto-encoder models, that guide them towards learning representations that capture the underlying explanatory factors for the observed input (Bourlard and Kamp 1988, Vincent et al. 2010). An auto-encoder model consists of an encoding function to compute a representation from an input, and a decoding function to reconstruct the input from the representation. The parameters involved in the two functions are trained to maximize the *reconstruction score*, which measures the similarity between the original input and reconstructed input. Inspired by the concept of *reconstruction*, Tu et al. (2017b) proposed guiding decoder hidden states to embed complete source information by reconstructing the hidden states back to the original source sentence. Our approach differs as follows: 1) we introduce not only a decoder-side reconstructor but also an encoder-side reconstructor to learn enhanced hidden states of both the encoder and decoder; and 2) we guide the hidden states to embed complete source information as well as the labelled DP information.

Experiments on a large-scale Chinese–English corpus show that the proposed approach significantly improves performance by addressing the DP translation problem. Furthermore, when reconstruction is applied only in training, it improves parameter training by producing better hidden representations that embed the DP information. Results show improvement over a strong NMT baseline system of +1.35 BLEU points without any increase in decoding speed. When additionally applying reconstruction during testing, we obtain a further +1.06 BLEU point improvement with only a slight decrease in decoding speed of approximately 18%.

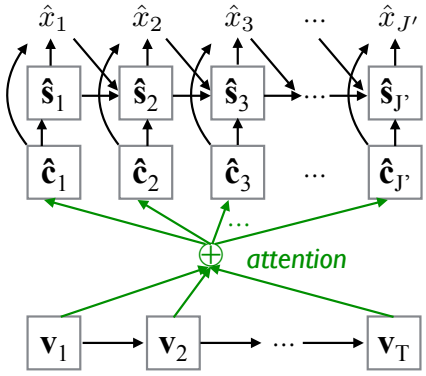


Figure 5.1: Architecture of the reconstructor.

5.2 Reconstruction-based Neural Machine Translation

In this section, we discuss methods of extending NMT models with a *reconstructor* to improve DP translation, which is inspired by "reconstruction", a standard concept in auto-encoder models (Bourlard and Kamp 1988, Vincent et al. 2010, Socher et al. 2011), and while has successfully been applied to NMT models (Tu et al. 2017b) recently.

5.2.1 Reconstructor

The basic idea of our approach is to reconstruct the annotated source sentence from the latent representations of the NMT model and use the reconstruction score to measure how well the DPs can be recalled from the latent representations. With the reconstruction score as an auxiliary training objective, we aim to encourage the latent representations to embed DP information, and thus recall the DP translation with enhanced representations.

The reconstructor reads a sequence of hidden states and the annotated source sentence, and outputs a reconstruction score. It employs an attention model (Bahdanau et al. 2015, Luong et al. 2015a) to reconstruct the annotated source sentence $\hat{\mathbf{x}} = \{\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_{J'}\}$ word by word, which is conditioned on the input latent representations $\mathbf{v} = \{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_T\}$, as shown in Figure (5.1). The reconstruction score is computed by Equation (5.1):

$$R(\hat{\mathbf{x}}|\mathbf{v}) = \prod_{j=1}^{J'} R(\hat{x}_j|\hat{x}_{<j}, \mathbf{v}) = \prod_{j=1}^{J'} g_r(\hat{x}_{j-1}, \hat{\mathbf{s}}_j, \hat{\mathbf{c}}_j) \tag{5.1}$$

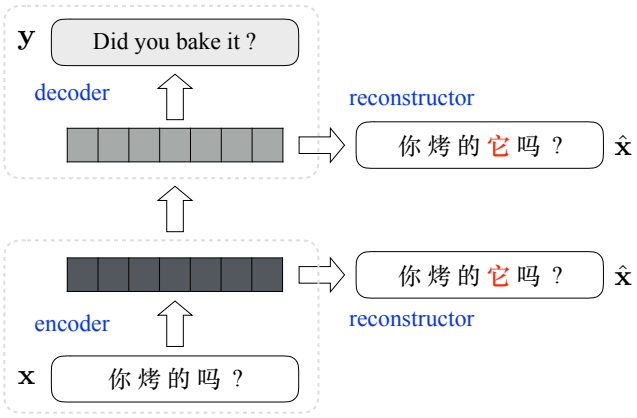


Figure 5.2: Architecture of reconstructor-augmented NMT. The two independent reconstructors reconstruct the annotated source sentence from hidden states in the encoder and decoder, respectively.

where $\hat{\mathbf{s}}_j$ is the hidden state in the reconstructor, and computed by Equation 5.2:

$$\hat{\mathbf{s}}_j = f_r(\hat{x}_{j-1}, \hat{\mathbf{s}}_{j-1}, \hat{\mathbf{c}}_j) \tag{5.2}$$

Here $g_r(\cdot)$ and $f_r(\cdot)$ are respectively softmax and activation functions for the reconstructor. The context vector $\hat{\mathbf{c}}_j$ is computed as a weighted sum of hidden states $\mathbf{v}$, as in Equation (5.3):

$$\hat{\mathbf{c}}_j = \sum_{t=1}^{T} \hat{\alpha}_{j,t} \cdot \mathbf{v}_t \tag{5.3}$$

where the weight $\hat{\alpha}_{j,t}$ is calculated by an additional attention model. The parameters related to the attention model, $g_r(\cdot)$, and $f_r(\cdot)$, are independent of the standard NMT model. The labeled source words $\hat{\mathbf{x}}$ share the same word embeddings with the NMT encoder.

5.2.2 Reconstructor Augmentation

We augment the standard encoder-decoder-based NMT model with the introduced reconstructor, as shown in Figure 5.2. The standard encoder-decoder reads the source sentence $\mathbf{x}$ and outputs its translation $\mathbf{y}$ along with the likelihood score. We introduce two independent reconstructors with their own parameters, each of which reconstructs the annotated source sentence $\hat{\mathbf{x}}$ from the encoder and decoder hidden states, respectively.

Encoder-Reconstructor-Decoder When adding a reconstructor to the encoder side only, we replace the standard encoder with an enhanced *auto-encoder*. In the case of auto-encoding, the encoder hidden states are not only used to summarize the original source sentence, but also to embed the recalled DP information from the annotated source sentence.

Encoder-Decoder-Reconstructor This is analogous to the framework proposed by Tu et al. (2017b), except that we reconstruct the annotated source sentence rather than the original sentence itself. It encourages the decoder hidden states to embed complete information from the source side, including the recalled DPs.

Combination As seen, reconstructors applied on different sides of the corpus may capture different patterns of DP information, and using them together can encourage both the encoder and decoder to learn recalled DP information. Our approach is very much inspired by recent success within question-answering, where a single information source is fed to multiple memory layers so that new evidence is captured in each layer and combined into subsequent layers (Sukhbaatar et al. 2015, Miller et al. 2016).

5.2.3 Learning and Inference

Learning We train both the encoder-decoder and the introduced reconstructors together in a single end-to-end process. The two-reconstructor model (as shown in Figure 5.2) is described below (the other two individual models correspond to each part). The training objective can be revised as in Equation (5.4):

$$\begin{aligned} J(\theta,\gamma,\psi) \quad &= \quad \arg\max_{\theta,\gamma,\psi} \sum_{n=1}^{N} \Big\{ \underbrace{\log P(\mathbf{y}^n|\mathbf{x}^n;\theta)}_{\textit{likelihood}} \\ &+ \quad \lambda \underbrace{\log R_{enc}(\hat{\mathbf{x}}^n|\mathbf{h}^n;\theta,\gamma)}_{\textit{enc-rec}} \\ &+ \quad \eta \underbrace{\log R_{dec}(\hat{\mathbf{x}}^n|\mathbf{s}^n;\theta,\psi)}_{\textit{dec-rec}} \Big\} \end{aligned} \tag{5.4}$$

where θ is the parameter matrix in the encoder-decoder, and γ and ψ are model parameters related to the *encoder-side reconstructor* ("enc-dec") and *decoder-side reconstructor* ("dec-rec"), respectively. λ and η are hyper-parameters that balance the preference between likelihood and reconstruction scores; $\mathbf{h}$ and $\mathbf{s}$ are encoder and decoder hidden states. The original training objective $P(\cdot)$ guides the standard NMT counterpart to provide better translations. Furthermore, the auxiliary reconstruction objectives ($R_{enc}(\cdot)$ and $R_{dec}(\cdot)$) guide the related part of the parameter matrix θ to learn better latent representations, which are used to reconstruct the annotated source sentence. The parameters of the model are trained to maximize the likelihood and reconstruction scores of a set of training examples $\{[\mathbf{x}^n, \mathbf{y}^n]\}_{n=1}^N$.

Inference Once a model is trained, we can use a beam search to find a translation that approximately maximizes the corresponding scores (*e.g.,* likelihood and reconstruction scores) in two strategies: 1) decoding with reconstruction, and 2) decoding without reconstruction. Note that, as the hidden states of the encoder are static, we only use the decoder-side reconstructor at decoding time.

In testing, reconstruction can serve as a reranking technique to select a better translation from the k-best candidates generated by the decoder. Each translation candidate is assigned a likelihood score from the standard encoder-decoder, as well as reconstruction score(s) from the newly added reconstructor(s). As shown in Figure 5.3, given an input sentence, a two-phase scheme is used:

1. The standard encoder-decoder produces a set of translation candidates, each of which is a triple consisting of a translation candidate, its corresponding decoder-side hidden layers s, and its likelihood score P.

2. For each translation candidate, the reconstructor reads its corresponding hidden layer on the target side and outputs an auxiliary reconstruction score R. Linear interpolation of the likelihood P and reconstruction score R produces an overall score, which

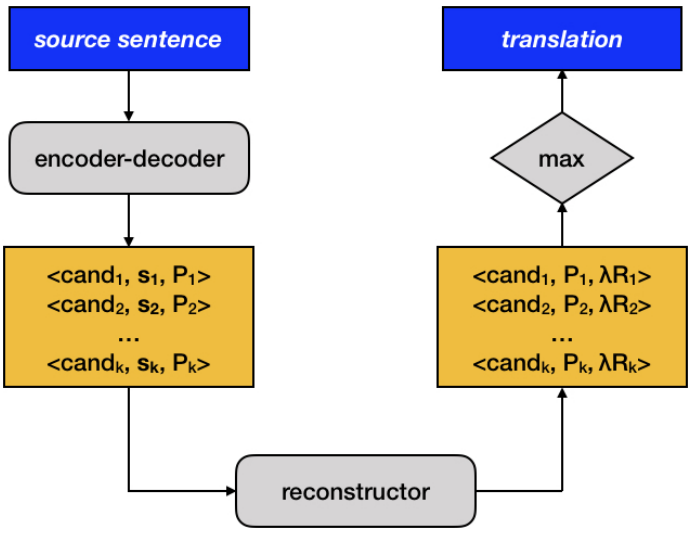


Figure 5.3: Illustration of decoding with reconstruction.

is used to select the final translation.[1]

When using reconstruction in testing, it requires external resource (*i.e.,* monolingual DP label tool) and more computations (*i.e.,* calculation of reconstruction scores). To reduce the dependency and cost, we can also perform decoding without the reconstructor. We only employ a standard encoder-decoder model with better trained parameters so that the parameters can produce enhanced latent representations that embed DP information. Such information is invisible in the original input sentence but can be learned from the training data with similar context.

DP Annotation and Generation Accordingly, there are two different methods to recover DPs at training and testing phases, respectively. In the training phase when the target sentence is available, we automatically annotate DPs for the source sentence using alignment information. During the testing phase, since the target sentence is invisible, we employ an external prediction model, which is trained on annotated source sentences in the training corpus. The details are described in Section 5.3.2.

[1]The interpolation weight λ in testing is the same as in training.

Data	$\|S\|$	$\|W\|$		$\|P\|$		$\|V\|$		$\|L\|$	
		Zh	En	Zh	En	Zh	En	Zh	En
Train	2.15M	12.1M	16.6M	1.66M	2.26M	151K	90.8K	5.63	7.71
Tune	1.09K	6.67K	9.25K	0.76K	1.03K	1.74K	1.35K	6.14	8.52
Test	1.15K	6.71K	9.49K	0.77K	0.96K	1.79K	1.39K	5.82	8.23

Table 5.3: Number of sentences ($|S|$), words ($|W|$), pronouns ($|P|$), vocabulary ($|V|$), and averaged sentence length ($|L|$) comprising the training, tuning and test corpora.

5.3 Experiments

In this section, we describe the data, model setup and results on the performance of our proposed models.

5.3.1 Data

Experiments evaluate the method for translation of Chinese–English subtitles. We extract more than two million sentence pairs from the subtitles of television episodes.[2] We pre-processed the extracted data using our in-house scripts (Wang et al. 2016), including sentence-boundary detection and bilingual sentence alignment. Finally, we obtained a high-quality corpus which includes the discourse information.

Table 5.3 shows the statistics of the corpus. Within the subtitle corpus, sentences are generally short and the Chinese side, as expected, contains many examples of DPs. We randomly select two complete television episodes as the tuning set, and another two episodes as the test set.[3]

5.3.2 DP Annotation and Generation

Similar to Chapter 4, we automatically annotate DPs for training and test data. In the *training phase*, where the target sentence is available, we annotate DPs for the source sentence using alignment information. These annotated source sentences can be used to build a

[2]The data were crawled from the subtitle website `http://www.zimuzu.tv`.

[3]Our released corpus is available at `https://github.com/longyuewangdcu/tvsub`.

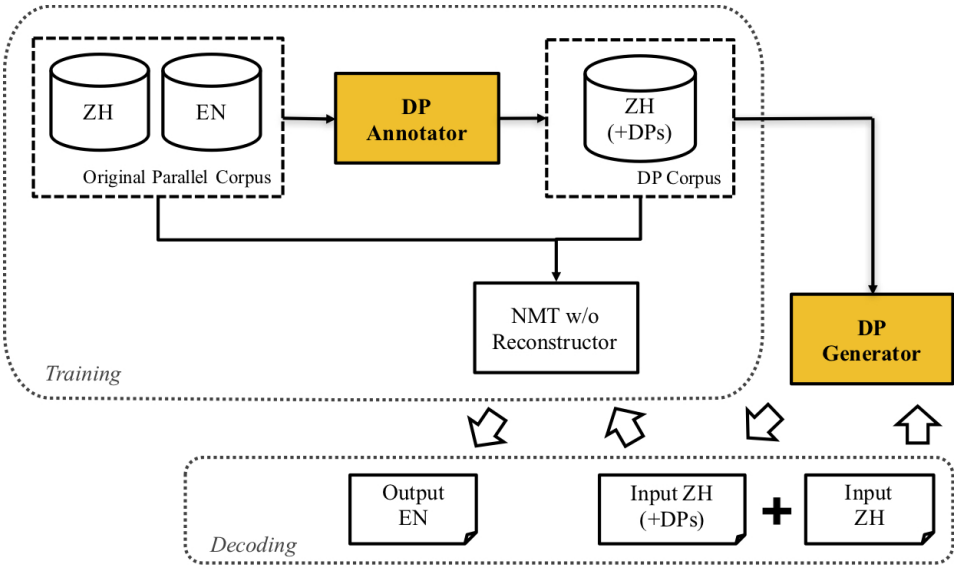


Figure 5.4: Illustration of DP Annotation and Generation.

monolingual DP generator using NN, which is used to annotate test sentences since the target sentence is not available during the *testing phase*. The F-scores of the two approaches on our data are 92.99% and 65.21%, respectively. After automatic annotation and generation, the number of pronouns on the Chinese side in training, tuning and test data are 2.09M, 0.98K, 0.96K, respectively, which is roughly consistent with pronoun frequency on the English side.

As shown in Figure 5.4, the usage of the annotated source sentences is two-fold:

1. *Baseline (+DPs)*: a stronger baseline system trained on the new parallel corpus (*i.e.,* annotated source sentence, target sentence), which is evaluated on the new test sentences annotated by the monolingual DP generator.

2. *Our models*: the proposed models use the hidden states to reconstruct the annotated source sentences.

Note that for the source sentences that have no DPs, we use the original ones as annotated source sentences; otherwise we use the DP-annotated sentences.

5.3.3 Model Setup

The baseline is our re-implemented attention-based NMT system, which incorporates dropout (Hinton et al. 2012) on the output layer and improves the attention model by feeding in the most recently generated word. For training the baseline models, we limited the source and target vocabularies to the most frequent 30K words in Chinese and English, covering approximately 97.2% and 99.3% of the words in the two languages, respectively. Each model was trained on sentences of length up to a maximum of 20 words with early stopping. Mini-batches were shuffled during processing with a mini-batch size of 80. The word-embedding dimension was 620 and the hidden layer size was 1,000. We trained for 20 epochs using Adadelta (Zeiler 2012), and selected the model that yielded the best performance on the tuning set.

The proposed model was implemented on top of the baseline model with the same settings where applicable. The hidden layer size in the reconstructor was 1,000. Following Tu et al. (2017b), we initialized the parameters of our models (*i.e.,* encoder and decoder, except those related to reconstructors) with the baseline model. We further trained all the parameters of our model for another 15 epochs.

5.3.4 Results

We investigate two baselines and three proposed variation models including:

Baseline: standard NMT model trained on original parallel corpus;

Baseline (+DP): standard NMT model trained on new parallel corpus whose source-side sentences are annotated with DPs;

+ enc-rec: NMT augmented with encoder-side reconstructor trained on triple corpus;

+ dec-rec: NMT augmented with decoder-side reconstructor trained on triple corpus;

+ enc-rec + dec-rec: NMT augmented with two-side reconstructors trained on triple corpus.

Model	#Params	Speed		BLEU	
		Training	Decoding	Test	△
Baseline	86.7M	1.60K	2.61	31.80	– / –
Baseline (+DPs)	86.7M	1.59K	2.63	32.67†	+0.87 / –
+ enc-rec	+39.7M	0.71K	2.63	33.67†‡	+1.87 / +1.00
+ dec-rec	+34.1M	0.84K	2.18	33.48†‡	+1.68 / +0.81
+ enc-rec + dec-rec	+73.8M	0.57K	2.16	**35.08**†‡	**+3.28 / +2.41**

Table 5.4: Evaluation of translation performance for Chinese–English. Training speed is measured in words/second and decoding speed is measured in sentences/second with beam size being 10. The two numbers in the "△" column denote performance improvements over "Baseline" and "Baseline (+DPs)", respectively. "†" and "‡" indicate statistically significant difference ($p < 0.01$) from "Baseline" and "Baseline (+DPs)", respectively. All listed models except "Baseline" exploit the annotated source sentences.

We used the case-insensitive 4-gram NIST BLEU metric (Papineni et al. 2002) for evaluation, and *sign-test* (Collins et al. 2005) to test for statistical significance. Table 5.4 shows translation performance for Chinese–English in terms of BLEU score.

Baselines There are two baseline NMT models: one trained and evaluated on the original parallel data without any explicitly annotated DPs (*i.e.,* "Baseline"), and the other trained and evaluated on the annotated data (*i.e.,* "Baseline (+DPs)"). As can be seen from the BLEU scores, the latter significantly outperforms the former, indicating that explicitly recalling translation of DPs helps produce better translations. Benefiting from the explicitly annotated DPs, the stronger baseline system is able to improve performance over the standard baseline system built on the original data where the pronouns are missing. Note that, the performance of our baseline is close to that of the state-of-the-art system, Nematus, using the same training corpus.

Parameters In terms of additional parameters introduced by the reconstruction models, both reconstructors introduce a large number of parameters. Beginning with the baseline model's 86.7M parameters, the encoder-side reconstructor adds 39.7M new parameters, while the decoder-side reconstructor adds a further 34.1M new parameters. Furthermore,

adding reconstructors to both sides leads to additional 73.8M parameters. More parameters may capture more information, at the cost of added complexity in training.

Speed Although gains are made in terms of translation quality by introducing reconstruction, we need to consider the potential trade-off with respect to a possible increase in training and decoding time, due to the large number of newly introduced parameters resulting from the incorporation of reconstructors into the NMT model. When running on a single GPU device Tesla K80, the training speed of the baseline model is 1600 target words per second, and this reduces to 570 words per second when reconstructors are added to both sides. In terms of decoding time trade-off, our most complex model only increases decoding speed by 18%. We attribute this to the fact that no beam search is required for calculating reconstruction scores, which avoids the very costly data swap between GPU and CPU memories.

Translation Quality Clearly the proposed approach significantly improves translation quality in all cases, although there are still considerable differences among the proposed variants. Introducing encoder-side and decoder-side reconstructors individually improves translation performance over "Baseline (+DPs)" by +1.0 and +0.8 BLEU points, respectively. Combining them together achieves the best performance overall, which is +2.4 BLEU points better than the strong baseline model. This confirms our assumption that reconstructors applied to the source and target sides indeed capture different patterns for translating DPs.

5.4 Analysis

We conducted extensive analysis on Chinese–English translation to better understand our model in terms of the contributions of reconstruction from training and testing, the effect of reconstructed input, the effect of DP labelling accuracy, and the ability to handle long sentences.

Model	Test	△
Baseline	31.80	– / –
Baseline (+DPs)	32.67	+0.87 / –
+ enc-rec	33.67	+1.87 / +1.00
+ dec-rec	33.15	+1.35 / +0.48
+ enc-rec + dec-rec	**34.02**	**+2.22 / +1.35**

Table 5.5: Translation results when *reconstruction is used in training only while not used in testing*.

5.4.1 Contribution Analysis

As mentioned in Section 5.3.2, the effect of reconstruction is two-fold: 1) it improves the training of baseline parameters, which leads to better hidden representations that embed labelled DP information learned from the training data; and 2) it serves as a reranking metric in testing to measure the quality of DP translation.[4] Table 5.5 lists translation results when the reconstruction model is used in training only. Results show that all variants outperform the baseline models, and applying reconstructors to both sides achieves the best performance overall. This is encouraging, since no extra resources nor computation are introduced to online decoding, making the approach highly practical, *e.g.,* for translation in industrial applications.

5.4.2 Effect of Reconstruction

Some researchers may argue that the proposed method acts much like dual learning (He et al. 2016a) and reconstruction (Tu et al. 2017b), especially when sentences have no DPs, which can benefit the overall translation, not just with respect to DPs only. To investigate to what extent the improvements are indeed made by explicitly modeling DP translation, we examine the performance of variants which reconstruct hidden states to the original input sentence instead of the source sentence annotated with DPs, as shown in Table 5.6. Note that the variant "+ dec-rec" in this setting is exactly the model proposed by Tu et al. (2017b).

[4]In testing, the encoder-side reconstructor reconstructs the same labelled source sentence with the same encoder hidden states, so all translation candidates share the same encoder-side reconstruction score. Accordingly, in such cases, reconstruction cannot be used as a reranking metric.

Model	Test	△
Baseline	31.80	– / –
Baseline (+DPs)	32.67	+0.87 / –
+ enc-rec	33.21	+1.41 / +0.54
+ dec-rec	33.08	+1.28 / +0.41
+ enc-rec + dec-rec	**33.25**	**+1.45 / +0.58**

Table 5.6: Translation results when hidden states are *reconstructed into the original source sentence* instead of the source sentence labelled with DPs.

As seen, although the variants significantly outperform the "Baseline" model without using any DP information, the absolute improvements are still worse than our proposed model that explicitly exploits DP information (*i.e.,* 1.45 BLEU vs. 3.28 BLEU). This validates our hypothesis that explicitly modeling DP translation contributes most to the improvement.

5.4.3 Effect of DP Generation Accuracy

For each sentence in testing, the DPs are labelled automatically by a DP generator model, the accuracy of which is 65.21% measured in F-score. The annotation errors may propagate to the NMT models, and have the potential to negatively affect translation performance. We investigate this using manual annotation and automatic annotation, as shown in Table 5.7. The analysis firstly shows that there still exists a significant gap in performance, and this could be improved by improving the accuracy of DP generator. Secondly, our models show a relatively smaller distance in performance from the oracle performance ("Manual"), indicating that the proposed approach is more robust to annotation errors.

Model	Automatic	Manual	△
Baseline (+DPs)	32.67	36.73	+4.06
+ enc-rec	33.67	37.58	+3.91
+ dec-rec	33.48	37.23	+3.75
+ enc-rec + dec-rec	35.08	38.38	+3.30

Table 5.7: Translation performance gap ("△") between manually ("Manual") and automatically ("Automatic") annotated DPs for input sentences in testing.

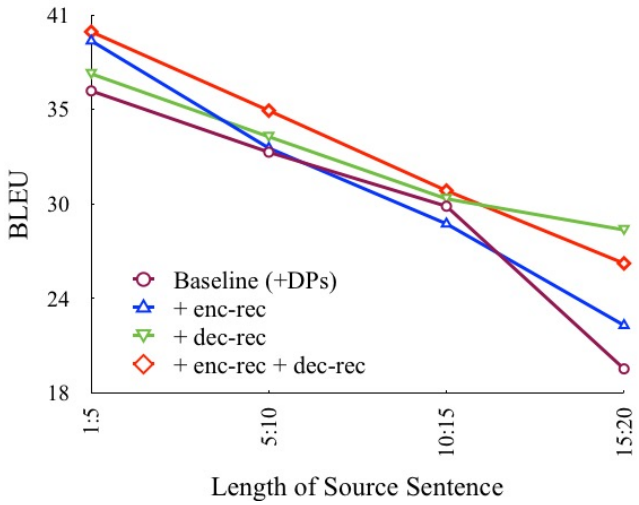


Figure 5.5: Performance of the generated translations with respect to the lengths of the source sentences.

5.4.4 Length Analysis

Following Bahdanau et al. (2015), Tu et al. (2016) and Tu et al. (2017a), we group sentences of similar lengths together and compute the BLEU score for each group, as shown in Figure 5.5. The proposed models outperform the baseline for most span lengths, although there are still some notable differences. The improvement achieved by the source-side reconstructor is mainly for translation of short (< 5) sentences, while that of the target-side reconstructor is mainly for translation of long (> 15) sentences. The reasons for this are 1) reconstruction can make encoder-side hidden states contain complete source information including DP information and subsequently achieve good performance on short sentences, while at the same time, they cannot guarantee that all the information will be transferred to the decoder side (*i.e.,* relatively bad performance on long sentences); 2) similar to the findings of Tu et al. (2017b), the decoder-side reconstructor can make translation more adequate, which significantly alleviates inadequate translation problems for longer sentences. Combining them together can take advantage of both models, and thus the improvements are more substantial for all span lengths.

Model	Error	Sub.	Obj.	Dum.	All
Baseline (+DP)	Total	112	41	45	198
+ enc-rec	Fixed	51	22	28	101
	New	25	8	4	37
+ dec-rec	Fixed	57	21	17	95
	New	19	10	6	36
+ enc-rec + dec-rec	Fixed	50	34	33	117
	New	11	14	7	32

Table 5.8: Translation error statistics on different types of pronouns: subject ("Sub."), object ("Obj.") and dummy ("Dum.") pronouns.

5.4.5 Error Analysis

We investigate to what extent DP-related errors are fixed by the proposed models. We randomly select 500 sentences from the test set and count errors produced by the strong baseline model ("Total"), and what proportion of these are fixed ("Fixed") or newly introduced ("New") by our approach, as shown in Table 5.8. All the proposed models can fix different kinds of DP problems, and the "+ enc + dec" variant achieves the best performance, which is consistent with the translation results reported above. The "+ enc + dec" model fixed 59.1% of the DP-related errors, while only introducing 16.2% new errors. This confirms that our improvement in terms of automatic metric scores indeed comes from alleviating DP translation errors.

Among all types of pronouns, translation errors object and dummy pronouns,[5] which can be usually inferred with intra-sentence context, are easy to alleviate. In contrast, errors related to the subject of a given sentence are more difficult, since annotating DPs in such cases generally requires cross-sentence context. Table 5.9 shows three typical examples of successfully fixed, failed to fix, and newly introduced subject-case pronouns.

[5]A dummy pronoun (*i.e.,* "it") is a pronoun used in syntax without explicit meaning. It is used in Germanic languages such as English but not in Pro-drop languages such as Chinese.

Fixed Error	
Input	等 我 搬进 来 **(我)** 可以 买 一台 泡泡机 吗 ?
Ref.	When I move in, can I get a bubble machine?
NMT	When I move in *to* buy a bubble machine.
Our	When I move in, can **I** buy a bubble machine?
Non-Fixed Error	
Input	**(他)** 是 个 训练营 ?
Ref.	It is a camp?
NMT	*He* was a camp?
Our	*He*'s a camp?
Newly Introduced Error	
Input	**(我)** 要 把 这 戒指 还给 你
Ref.	I need to give this ring back to you.
NMT	**I**'m gonna give you the ring back.
Our	*To* give it back to you.

Table 5.9: Example translations where subject-case pronouns in brackets are dropped in the original input but labeled by the DP generator. We italicize some *mis-translated* errors and highlight the **correct** ones in bold.

5.5 Comparison and Adaptation

In this section, we first conduct experiments to compare our approach with other models to see whether they can also help DP translation or not. Secondly, we adapt our approach to Japanese–English to show the robustness of our models.

5.5.1 Comparison to Other Work

Recently, it was shown that NMT can be improved by feeding auxiliary information sources beyond the original input sentence. The additional sources can be in various forms, such as parallel sentences in other languages (Dong et al. 2015, Zoph and Knight 2016a), cross-sentence contexts (Jean et al. 2017, Tu et al. 2018), generation recommendations from other translation models (He et al. 2016b, Wang et al. 2017, Gu et al. 2017, Wang et al. 2017), or syntax information (Li et al. 2017, Zhou et al. 2017). In the same direction, we provide

complementary information in terms of source sentences labelled with DPs.

For the purpose of comparison, we reimplemented the multi-source model of Zoph and Knight (2016a), which introduces an alternate encoder (shared parameters) and attention model (independent parameters) that take annotated sentences as an additional input source.

Furthermore, some may argue that the improvements in BLEU are mainly due to the increase in model parameters (*e.g.,* +73.8M) or deeper layers (*e.g.,* two reconstruction layers). To answer these concerns, we compared the following two models:

- Multi-Layer (Wu et al. 2016): a system with a three-layer encoder and three-layer decoder. The additional layers introduce 75.1M parameters, which is of a similar scale to the proposed model (*i.e.,* 73.8M).

- Baseline (+DPs) + Enlarged Hidden Layer: a system with the same setting as "Baseline (+DPs)" except that layer size is 2100 instead of 1000. This variant introduces 86.6M parameters, which is even more than the most complicated variant of our proposed models.

Table 5.10 shows the comparison results. This multi-source model significantly outperforms our "Baseline" model without annotated DP information, but only marginally outperforms the "Baseline (+DPs)" that uses annotated DPs. One possible reason is that the two sources (*i.e.,* original input and labelled input sentences) are too similar to one another, making it difficult to distinguish them from annotated DPs. We found that the multi-layer model significantly outperforms its single-layer counterpart "Baseline (+DPs)", while significantly underperforms our best model (*i.e.,* 33.46 BLEU vs. 35.08 BLEU). The "Baseline (+DPs)" system with the enlarged hidden layer, however, does not achieve any improvement. This indicates that explicitly modeling DP translation is the key factor to the performance improvements we have seen.

Model	#Params	Speed		BLEU	
		Training	Decoding	Test	△
Baseline	86.7M	1.60K	2.61	31.80	– / –
Baseline (+DPs)	86.7M	1.59K	2.63	32.67†	+0.87 / –
Multi-Source (Zoph and Knight 2016a)	+20.7M	1.17K	1.27	32.81†	+1.01 / +0.14
Multi-Layer (Wu et al. 2016)	+75.1M	0.61K	2.42	33.36†	+1.56 / +0.69
Baseline (+DPs) + Enlarged Hidden Layer	+86.6M	0.68K	2.51	32.00†	+0.20 / -0.67

Table 5.10: Evaluation of translation performance for Chinese–English. Training speed is measured in words/second and decoding speed is measured in sentences/second with beam size being 10. The two numbers in the "△" column denote performance improvements over "Baseline" and "Baseline (+DPs)", respectively. "†" and "‡" indicate statistically significant difference ($p < 0.01$) from "Baseline" and "Baseline (+DPs)", respectively. All listed models except "Baseline" exploit the annotated source sentences.

5.5.2 Japanese–English Translation

To validate the robustness of our approach on other pro-drop languages, we conducted experiments on Opensubtitle2016[6] data for Japanese–English translation. We also randomly select around 1000 tuning and testing sets, respectively.

We used the same settings as in our Chinese–English experiments, except that the vocabulary size is 20,001. As shown in Table 5.11, our model also significantly improves translation performance on the Japanese–English task, demonstrating the efficiency and potential universality of the proposed approach.

Model	Test	△
Baseline (+DPs)	20.55	–
+ enc-rec + dec-rec	21.84	+ 1.29

Table 5.11: Evaluation of translation performance for Japanese–English.

[6]Available at: `http://opus.nlpl.eu/OpenSubtitles2016.php`

5.6 Summary

We have proposed an early attempt to model DP translation for NMT systems. Hidden states are guided in both the encoder and decoder to embed the DP information by reconstructing them back to the source sentence labelled with DPs. The effect of the reconstruction model is two-fold: 1) it improves parameter training for producing better latent representations; and 2) it measures the quality of DP translation, which is combined with likelihood to better measure the overall quality of translations. We quantitatively and qualitatively show that the proposed approach significantly improves translation performance across different language pairs, and can be further improved by developing better DP labelling models. Our main contributions can be summarized as follows:

1. We show that although NMT models advance SMT models on translating pro-drop languages, there is still large room for improvement;

2. We introduce a reconstruction-based approach to improve dropped pronoun translation;

3. We release a large-scale bilingual dialogue corpus, which consists of 2.2M Chinese–English sentence pairs.[7]

In future work we plan to validate the effectiveness of our approach on other text genres with different prevalence of DPs. For example, in formal text genres (*e.g.,* newswire), DPs are not as common as in the informal text genres, and the most frequently dropped pronouns in Chinese newswire is the third person singular "它" ("*it*") (Baran et al. 2012), which may not be crucial to translation performance.

In the next chapter, we will explore how to improve DP translation using cross-sentence information in an end-to-end manner.

[7]Our released corpus is available at `https://github.com/longyuewangdcu/tvsub`.

Chapter 6

An End-to-End Dropped Pronoun Translation Model by Exploiting Cross-Sentence Context

In Chapter 3, we demonstrated that the NMT model can improve translation quality by considering document-level information. As discussed in Chapters 4 and 5, pro-drop is a particular discourse phenomenon which needs cross-sentence context for DP recovery and translation. In this chapter, we propose a novel approach to jointly learn to translate and predict DPs with cross-sentence context. This chapter directly addresses our fourth research question as described below:

> **RQ 4** *Can we build a fully end-to-end neural model for dropped pronoun translation? Is cross-sentence context useful for dropped pronoun prediction?*

This chapter is organized as follows. We first introduce the motivation of end-to-end modelling and cross-sentence context for DP translation in Section 6.1. In Section 6.2, we describe our advanced approaches based on the original reconstruction-based NMT model 5. We conduct experiments on the Chinese–English translation and provide the promising results in Section 6.3. We also quantitatively and qualitatively demonstrate that

the presented model significantly outperforms the best reconstruction-based NMT model in Section 6.4, which is followed by the chapter summary in Section 6.5.

6.1 Why End-to-End Modelling and Cross-Sentence Context?

As discussed in Chapter 5, we introduced two independent reconstructors with their own parameters, which reconstruct the DP-annotated source sentence from the encoder and decoder hidden states, respectively. The central idea underpinning the approach is to guide the corresponding hidden states to embed the recalled source-side DP information and subsequently to help the NMT model generate the missing pronouns with these enhanced hidden representations. The DPs can be automatically annotated for training and test data using two different strategies. In the *training phase*, where the target sentence is available, we annotate DPs for the source sentence using alignment information. These annotated source sentences can be used to build a neural DP predictor, which can be used to annotate test sentences since the target sentence is not available during the *testing phase*. Although this previous model achieved significant improvements, there nonetheless exist three drawbacks:

1. Recent work shows that NMT models can benefit from sharing a component across different tasks and languages (Dong et al. 2015, Firat et al. 2016, Zoph and Knight 2016a, Anastasopoulos and Chiang 2018). However, there is no interaction between our two separate reconstructors, which misses the opportunity to exploit potentially useful relations between encoder and decoder representations.

2. The *testing phase* is still a pipeline method, where the DP annotation is automatically performed by an external DP prediction model. However, the DP predictor only has an accuracy of 66% F1-score (as shown in Section 4.4.3), which propagates numerous errors to the translation model.

3. The DNMT model has shown promising results by modelling document-level information. Although the DP prediction model considers the document-level features,

we just simply integrate the prediction results into sentence-level NMT models.

In response to these problems, we propose to improve our original model from three perspectives. First, in order to better exploit representations, we use a *shared* reconstructor to read hidden states from both encoder and decoder. Second, to avoid the error propagation problem, we integrate a DP predictor into NMT to *jointly* learn to translate and predict DPs. Incorporating these as two auxiliary loss terms can guide both the encoder and decoder states to learn critical information relevant to DPs. Third, we further improve DP translation using cross-sentence representations, which are summarized by hierarchical RNN. Experimental results on a Chinese–English subtitle corpus show that the three modifications can accumulatively improve translation performance, and the best result is +1.5 BLEU points better than that reported in Chapter 5. In addition, the jointly learned DP prediction model significantly outperforms its external counterpart by 9% in F1-score.

6.2 An End-to-End Dropped Pronoun Translation Model with Cross-Sentence Context

In this section, we discuss approaches of 1) shared reconstruction mechanism, 2) learning to jointly translate and predict DPs, and 3) incorporating cross-sentence context into NMT.

6.2.1 Shared Reconstructor

Recent work shows that NMT models can benefit from sharing a component across different tasks and languages. Taking multi-language translation as an example, Firat et al. (2016) share an attention model across languages while Dong et al. (2015) share an encoder. Our work is most similar to the work of Zoph and Knight (2016a) and Anastasopoulos and Chiang (2018), which share a decoder and two separate attention models to read from two different sources. In contrast, we share information at the level of reconstruction frame.

The architectures of our proposed shared reconstruction model are shown in Figure 6.1. Formally, the shared reconstructor reads from both the encoder and decoder hidden states,

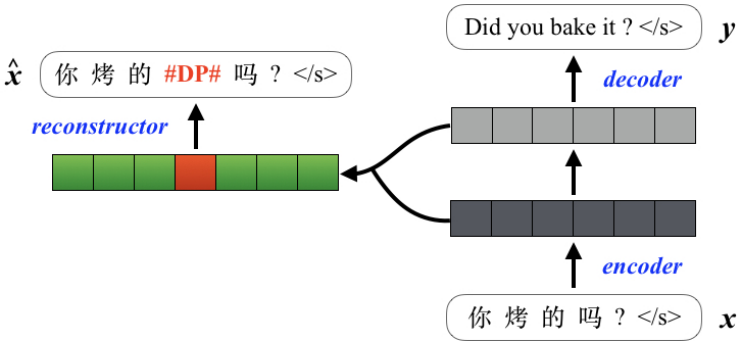


Figure 6.1: Architecture of the shared reconstructor, in which the words in red are automatically annotated DPs.

as well as the DP-annotated source sentence, and outputs a reconstruction score. It uses two separate attention models to reconstruct the annotated source sentence $\hat{\mathbf{x}} = \{\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_T\}$ word by word, and the reconstruction score is computed by Equation (6.1):

$$R(\hat{\mathbf{x}}|\mathbf{h}^{enc}, \mathbf{h}^{dec}) = \prod_{t=1}^{T} g_r(\hat{x}_{t-1}, \mathbf{h}_t^{rec}, \hat{\mathbf{c}}_t^{enc}, \hat{\mathbf{c}}_t^{dec}) \tag{6.1}$$

where $\mathbf{h}_t^{rec}$ is the hidden state in the reconstructor, and computed by Equation (6.2):

$$\mathbf{h}_t^{rec} = f_r(\hat{x}_{t-1}, \mathbf{h}_{t-1}^{rec}, \hat{\mathbf{c}}_t^{enc}, \hat{\mathbf{c}}_t^{dec}) \tag{6.2}$$

Here $g_r(\cdot)$ and $f_r(\cdot)$ are, respectively, softmax and activation functions for the reconstructor. The context vectors $\hat{\mathbf{c}}_t^{enc}$ and $\hat{\mathbf{c}}_t^{dec}$ are the weighted sum of $\mathbf{h}^{enc}$ and $\mathbf{h}^{dec}$, respectively, as in Equations (6.3) and (6.4):

$$\hat{\mathbf{c}}_t^{enc} = \textstyle\sum_{j=1}^{J} \hat{\alpha}_{t,j}^{enc} \cdot \mathbf{h}_j^{enc} \tag{6.3}$$

$$\hat{\mathbf{c}}_t^{dec} = \textstyle\sum_{i=1}^{I} \hat{\alpha}_{t,i}^{dec} \cdot \mathbf{h}_i^{dec} \tag{6.4}$$

Note that the weights $\hat{\alpha}^{enc}$ and $\hat{\alpha}^{dec}$ are calculated by two separate attention models. We propose two attention strategies which differ as to whether the two attention models have interactions or not.

Independent Attention It calculates the two weight matrices independently, as in Equations (6.5) and (6.6):

$$\hat{\alpha}^{enc} = \text{ATT}_{enc}(\hat{x}_{t-1}, \mathbf{h}^{rec}_{t-1}, \mathbf{h}^{enc}) \quad (6.5)$$

$$\hat{\alpha}^{dec} = \text{ATT}_{dec}(\hat{x}_{t-1}, \mathbf{h}^{rec}_{t-1}, \mathbf{h}^{dec}) \quad (6.6)$$

where $\text{ATT}_{enc}(\cdot)$ and $\text{ATT}_{dec}(\cdot)$ are two separate attention models with their own parameters.

Interactive Attention It feeds the context vector produced by one attention model to another attention model. The intuition behind this is that the interaction between two attention models can lead to a better exploitation of the encoder and decoder representations. As the interactive attention is directional, we have two options (Equation (6.7) and (6.8)) which modify either $\text{ATT}_{enc}(\cdot)$ or $\text{ATT}_{dec}(\cdot)$ while leaving the other one unchanged:

- *enc→dec*:

$$\hat{\alpha}^{dec} = \text{ATT}_{dec}(\hat{x}_{t-1}, \mathbf{h}^{rec}_{t-1}, \mathbf{h}^{dec}, \hat{\mathbf{c}}^{enc}_{t}) \quad (6.7)$$

- *dec→enc*:

$$\hat{\alpha}^{enc} = \text{ATT}_{enc}(\hat{x}_{t-1}, \mathbf{h}^{rec}_{t-1}, \mathbf{h}^{enc}, \hat{\mathbf{c}}^{dec}_{t}) \quad (6.8)$$

6.2.2 Joint Prediction of Dropped Pronouns

Inspired by recent successes of multi-task learning (Dong et al. 2015, Luong et al. 2016), we propose to jointly learn to translate and predict DPs. As shown in Table 6.1, we explored to predict the exact DP words,[1] the accuracy of which is only 66% in F1-score. By analyzing the translation outputs, we found that 16.2% of errors are newly introduced and caused

[1]Unless otherwise indicated, in this chapter, the terms "DP" and "DP word" are identical.

Prediction	F1-score	Example
DP Words	66%	你 烤 的 它 吗 ?
DP Position	88%	你 烤 的 #DP# 吗 ?

Table 6.1: Evaluation of external models on predicting the positions of DPs ("DP Position") and the exact words of DPs ("DP Words").

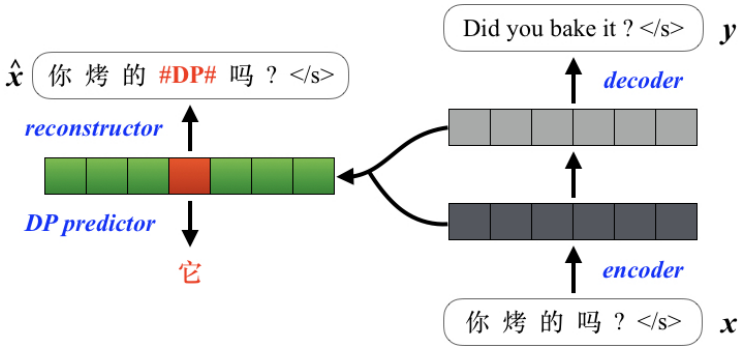


Figure 6.2: Architecture of the DPP-augmented NMT model, in which the words in red are automatically annotated DPs and DPPs.

by errors from the DP predictor. Fortunately, the accuracy of predicting the DPP is much higher, which provides the chance to alleviate the error propagation problem. Thus, our first method is similar to our model introduced in Section 6.5, but the difference is that we learn to generate DPs at the predicted positions using a jointly trained DP predictor, which is fed with informative representations in the reconstructor. Although the strategy improves performance by alleviating the error propagation problem, it still relies on an external toolkit to detect DPPs. Thus, in our second method, we move one step further by proposing an end-to-end DP translation model, which does not rely on any external toolkit.

DPP-Augmented NMT Model We integrate the DPP Predictor into the NMT model, as shown in Figure 6.2. We leverage the information of DPPs predicted by an external model, which can achieve an accuracy of 88% in F1-score. Accordingly, we transform the original DP prediction problem to DP word generation given the pre-predicted DP positions. Since the DPP-annotated source sentence serves as the reconstructed input, we introduce an additional *DP-generation loss* function, which measures how well the DP is generated

from the corresponding hidden state in the reconstructor.

Let $\mathbf{dp} = \{dp_1, dp_2, \ldots, dp_D\}$ be the list of DPs in the annotated source sentence, and $\mathbf{h}^{rec} = \{\mathbf{h}_1^{rec}, \mathbf{h}_2^{rec}, \ldots, \mathbf{h}_D^{rec}\}$ be the corresponding hidden states in the reconstructor. The generation probability is computed by Equation (6.9):

$$\begin{aligned} P(\mathbf{dp}|\mathbf{h}^{rec}) &= \prod_{d=1}^{D} P(dp_d|\mathbf{h}_d^{rec}) \\ &= \prod_{d=1}^{D} g_p(dp_d|\mathbf{h}_d^{rec}) \end{aligned} \tag{6.9}$$

where $g_p(\cdot)$ is softmax for the DP predictor.

We train both the encoder-decoder and the shared reconstructors together in a single end-to-end process, and the training objective is Equation (6.10):

$$\begin{aligned} J(\theta, \gamma, \psi) = \arg\max_{\theta,\gamma,\psi} \Big\{ & \underbrace{\log L(\mathbf{y}|\mathbf{x};\theta)}_{\textit{likelihood}} \\ & + \underbrace{\log R(\hat{\mathbf{x}}|\mathbf{h}^{enc}, \mathbf{h}^{dec};\theta,\gamma)}_{\textit{reconstruction}} \\ & + \underbrace{\log P(\mathbf{dp}|\hat{\mathbf{h}}^{rec};\theta,\gamma,\psi)}_{\textit{prediction}} \Big\} \end{aligned} \tag{6.10}$$

where $\{\theta, \gamma, \psi\}$ are, respectively, the parameters associated with the encoder-decoder, shared reconstructor and the DP prediction model. The auxiliary reconstruction objective $R(\cdot)$ guides the related part of the parameter matrix θ to learn better latent representations, which are used to reconstruct the DPP-annotated source sentence. The auxiliary prediction loss $P(\cdot)$ guides the related part of both the encoder-decoder and the reconstructor to learn better latent representations, which are used to predict the DPs in the source sentence.

End-to-End DP Translation Model We cast DP prediction as a sequence labelling task, where each word is labelled if there is a pronoun missing before it. Given the reconstructed input $\mathbf{x} = \{x_1, x_2, \ldots, x_T\}$ with the last word x_T being the end-of-sentence tag "

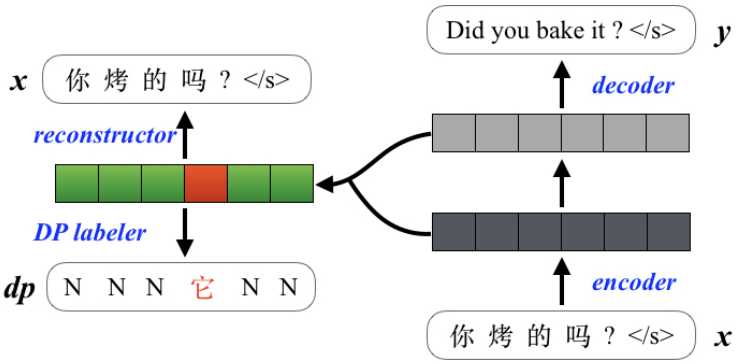


Figure 6.3: Architecture of the end-to-end DP translation model, in which the words in red are automatically annotated DPPs.

$\langle$eos$\rangle$",[2] the output to be labelled is a sequence of labels $\mathbf{dp} = \{dp_1, dp_2, \ldots, dp_T\}$ with $dp_t \in \{N\} \cup \mathbb{V}_{dp}$. Among the label set, "$N$" denotes no DP, and $\mathbb{V}_{dp}$ is the vocabulary of pronouns[3]. Taking Figure 6.3 as an example, the label sequence "N N N 它 N N" indicates that the pronoun "它" is missing before the fourth word "吗". More specifically, we model the probability of generating the label sequence $\mathbf{dp}$ as in Equation (6.11):

$$\begin{aligned} P(\mathbf{dp}|\mathbf{h}^{rec}) &= \prod_{t=1}^{T} P(dp_t|\mathbf{h}_t^{rec}) \\ &= \prod_{t=1}^{T} g_l(dp_t, \mathbf{h}_t^{rec}) \end{aligned} \tag{6.11}$$

where $g_l(\cdot)$ is softmax for the DP labeler. As can be seen, there is no reliance on external DP/DPP prediction models.

The newly introduced components are trained together with the standard encoder-decoder

[2]We introduce " $\langle$eos$\rangle$" to cover the case where a pronoun is missing at the end of a sentence.
[3]We employ the pronoun vocabulary used in Table 4.3, which contains 25 distinct Chinese pronouns.

in an end-to-end manner:

$$
\begin{aligned}
J(\theta, \gamma, \psi) = \arg\max_{\theta,\gamma,\psi} \Big\{ & \underbrace{\log L(\mathbf{y}|\mathbf{x};\theta)}_{\textit{likelihood}} \\
& + \underbrace{\log R(\mathbf{x}|\mathbf{h}^{enc}, \mathbf{h}^{dec};\theta,\gamma)}_{\textit{reconstruction}} \\
& + \underbrace{\log P(\mathbf{dp}|\mathbf{h}^{rec};\theta,\gamma,\psi)}_{\textit{labeling}} \Big\}
\end{aligned} \tag{6.12}
$$

where $\{\theta, \gamma, \psi\}$ are, respectively, the parameters associated with the encoder-decoder, shared reconstructor and the DP labeling model. The usage of the auxiliary reconstruction objective $R(\cdot)$ is two-fold: 1) it guides the reconstructor states to embed necessary source-side information, which is then used to predict the DP labels; and 2) it serves as a reranking technique to select a better translation from the k-best candidates in testing (Tu et al. 2017b). The auxiliary labeling loss $P(\cdot)$ guides the hidden states of both the encoder-decoder and the reconstructor to embed the DPs in the source sentence. Although the calculation of labeling loss relies on explicitly annotated labels, it is only used in training to guide the parameters to learn DP-enhanced representations. Benefiting from the implicit integration of DP information, we remove the reliance on external DP prediction model in testing.

6.2.3 Cross-Sentence Context Augmentation

The DP labelling model in Section 6.2.2 only considers each single sentence, which misses potentially useful discourse information from surrounding sentences. Thus, as shown in Figure 6.4, we exploit information from previous sentences, which have proven useful for pronoun prediction (Voita et al. 2018).

Cross-Sentence Context Summarization As described in Chapter 4, we consider the previous K source sentences $\mathbf{X} = \{\mathbf{x}^{-K}, \ldots, \mathbf{x}^{-1}\}$, which is summarized in a hierarchical way as shown in the left panel of Figure 6.4. For each sentence $\mathbf{x}^{-k}$, we employ a *word-*

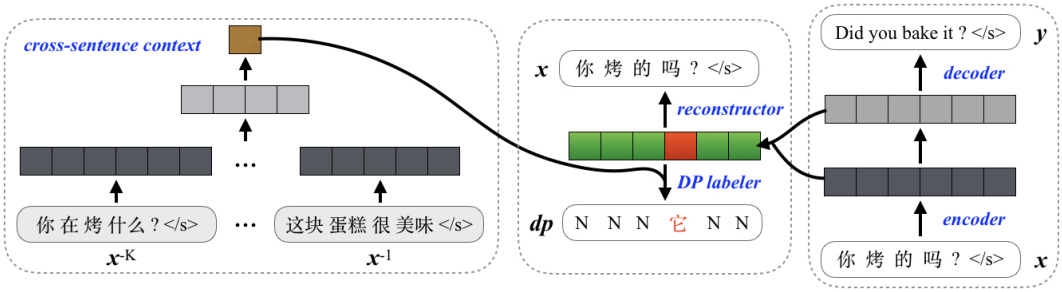


Figure 6.4: Architecture of the end-to-end DP translation model with cross-sentence context, in which the words in red are automatically annotated DPPs.

level encoder to summarize the representation of the whole sentence as in Equation (6.13):

$$\mathbf{h}^{-k} = \text{ENCODER}_{word}(\mathbf{x}^{-k}) \tag{6.13}$$

After we can obtain all sentence-level representations $\mathbf{H}^X = \{\mathbf{h}^{-K}, \ldots, \mathbf{h}^{-1}\}$, we feed them into a *sentence-level encoder* to produce a vector that represents the summary of the cross-sentence context, as in Equation (6.14):

$$\mathbf{C} = \text{ENCODER}_{sentence}(\mathbf{H}^X) \tag{6.14}$$

Following Voita et al. (2018), we share the parameters of word-level context encoder with the source encoder.

Integration into DP Prediction Intuitively, we can follow the method in Chapter 4 to transform the contextual representation to decoder states as an auxiliary input, which are subsequently propagated to reconstructor states as in Equation (6.15):

$$\mathbf{h}^{dec} = \text{DECODER}(\mathbf{h}^{enc}, \mathbf{C}) \tag{6.15}$$

In this way, the cross-sentence context can benefit both the generation of the translation and DP prediction. However, one potential problem with this strategy is that the propagation path is long: $\mathbf{C} \rightarrow \mathbf{h}^{dec} \rightarrow \mathbf{h}^{rec} \rightarrow \mathbf{dp}$, which may suffer from the vanishing effect.

To shorten the propagation path, we directly feed the cross-sentence context to the calculation of labeling loss, as in Equation (6.16):

$$P(\mathbf{dp}|\mathbf{h}^{rec}, \mathbf{C}) = \prod_{t=1}^{T} g_l(dp_t, \mathbf{h}_t^{rec}, \mathbf{C}) \tag{6.16}$$

where the path becomes $\mathbf{C} \rightarrow \mathbf{dp}$.

6.3 Experiments

In this section, we describe the experimental setup and results on the performance of our proposed models.

6.3.1 Setup

To compare our work with the results reported in previous work (in Chapter 5), we conducted experiments on the same Chinese–English TV Subtitle corpus. As described in Section 5.3.1, the training, validation, and test sets contain 2.15M, 1.09K, and 1.15K sentence pairs, respectively. We used case-insensitive 4-gram NIST BLEU metrics (Papineni et al. 2002) for evaluation, and *sign-test* (Collins et al. 2005) to test for statistical significance.

We implemented our models on the same code repository[4] and used the same configurations (*e.g.,* vocabulary size = 30K, hidden size = 1000). It should be emphasized that we did not use the pre-train strategy as in the previous chapter, since we found training from scratch achieved a better performance in the shared reconstructor setting.

6.3.2 Results

We totally investigate four baselines and four proposed models including:

Baseline: standard NMT model trained on original parallel corpus;

[4] `https://github.com/tuzhaopeng/nmt`

Baseline (+DPs): standard NMT model trained on new parallel corpus whose source-side sentences are annotated with DPs. At decoding time, we employ an external model to recover DPs for input sentences;

Baseline (+DPPs): standard NMT model trained on new parallel corpus whose source-side sentences are annotated with DPPs. At decoding time, we employ an external model to recover DPPs for input sentences;

Separate-Recs⇒(+DPs): the best reconstruction model proposed in Chapter 5, which use two independent reconstructors reconstruct the DP-annotated source sentence from the encoder and decoder hidden states;

Shared-Rec⇒(+DPPs): the proposed shared-recontruction model trained on new parallel corpus whose source-side sentences are annotated with DPPs;

+ Joint (+DP Predictor): we integrate DP prediction model into **Shared-Rec⇒(+DPPs)** model to jointly learn to translate and predict DPs;

+ Joint (+DP Labeler): we integrate DP labelling model into **Shared-Rec⇒(+DPPs)** model to jointly learn to translate DP and label tags for each token;

+ Cross-Sentence Context: we integrate cross-sentence model into **Shared-Rec⇒(+DPPs) + Joint (+DP Labeler)** model.

Table 6.2 shows the translation results. It is clear that the proposed models significantly outperform the baselines in all cases, although there are considerable differences among different variations.

Baselines (Rows 1-4): The three baselines (Rows 1, 2, and 4) are all trained on the standard NMT model, but differ with respect to the training data used: 1) **Baseline**: original parallel corpus; 2) **Baseline (+DPs)**: new parallel corpus whose source-side sentences are annotated with DPs; 3) **Baseline (+DPPs)**: new parallel corpus whose source-side sentences are annotated with DPPs. The baseline trained on the DPP-annotated data ("Baseline (+DPPs)", Row 4) outperforms the other two counterparts (*i.e.,* +1.38 and +0.51 BLEU

#	Model	#Params	Speed		BLEU
			Train	Decode	
Previous Work in Chapter 5					
1	Baseline	86.7M	1.60K	15.23	31.80
2	Baseline (+DPs)	86.7M	1.59K	15.20	32.67
3	Separate-Recs⇒(+DPs)	+73.8M	0.57K	12.00	35.08
Our Models					
4	Baseline (+DPPs)	86.7M	1.54K	15.19	33.18
5	Shared-Rec⇒(+DPPs)	+86.6M	0.52K	11.87	35.27†‡
6	+ Joint (+DP Predictor)	+91.9M	0.48K	11.84	36.53†‡
7	+ Joint (+DP Labeler)	+86.7M	0.54K	11.96	36.04†‡
8	+ Cross-Sentence Context	+121.2M	0.40K	11.71	**36.77**†‡

Table 6.2: Evaluation of translation performance. "Baseline" is trained and evaluated on the original data, while "Baseline (+DPs)" and "Baseline (+DPPs)" are trained on the data annotated with DPs and DPPs, respectively. Training and decoding (beam size is 10) speeds are measured in words/second. "†" and "‡" indicate statistically significant difference ($p < 0.01$) from "Baseline (+DDPs)" and "Separate-Recs⇒(+DPs)", respectively.

point), indicating that the error propagation problem does affect the performance of translating DPs. It suggests the necessity of jointly learning to translate and predict DPs. Furthermore, "Separate-Recs⇒(+DPs)" (Row 3) is the best model reported in Chapter 5, which we employed as another strong baseline (*i.e.,* 35.08 BLEU points).

Our Models (Rows 5-8): As described in Section 6.2.1, using our shared reconstructor (Row 5) on DPPs can achieve 35.27 BLEU points. This method not only outperforms the corresponding baseline (Row 4), but also surpasses its separate reconstructor counterpart (Row 3). It indicates that the DP reconstructor can really benefit from sharing the knowledge between the encoder and decoder. Here we only show the performance of the best attention strategy (*i.e.,* interactive *enc*→*dec*). We will further analyze different strategies in Section 6.4.

Based on the shared reconstruction model, we explore two joint learning methods: "+ DP Predictor" and "+ DP Labeler" (as discussed in Section 6.2.2). First, introducing a joint

DP prediction objective (Row 6) can achieve a further improvement of +1.26 BLEU points. These results verify that the shared reconstructor and jointly predicting DPs can accumulatively improve translation performance. Second, our end-to-end DP translation model (Row 7) achieves a relatively small improvement of +0.77 BLEU point. The main reason is that the "+ DP Labeler" model just label DPs based on the current sentence without considering any cross-sentence information. In contrast, the "+ DP Predictor" approach employs an external DPP prediction model which models a number of larger context features (as shown in Table 4.6).

As introduced in Section 6.2.3, we add hierarchical RNN to the "+ DP Labeler" model to model cross-sentence context. The "+ Cross-Sentence Context" approach (Row 8) achieves the best performance of 36.77 BLEU points, which is +1.69 better than the strong baseline (Row 3) and further +0.24 BLEU point than the best joint model (Row 6). We attribute the superior performance to the fact that the cross-sentence context over encoder representations embeds useful DP information, which can help to better label and translate DPs.

Speed Similar to the results of previous work in Section 5.3.4, the proposed approach improves BLEU scores at the cost of decreased training speed, which is due to the large number of newly introduced parameters resulting from the incorporation of shared reconstructor (or joint components or hierarchical RNN) into the NMT model. In terms of decoding time trade-off, our most complex model (Row 8) only increases decoding speed by 2.5% comparing with the original reconstruction model (Row 3). We attribute this to the fact that no beam search is required for calculating reconstruction scores, which avoids the very costly data swap between GPU and CPU memories.

6.4 Analysis

We conducted extensive analysis to better understand our model in terms of the effect of shared reconstruction from training and testing, the effect of different attention strategies, and the effect of DP prediction accuracy.

Model	Test	△
Baseline (+DPPs)	33.18	–
Separate-Recs (+DPs)	34.02	+0.84
Shared-Rec (+DPPs)	**34.80**	**+1.62**

Table 6.3: Translation results when *reconstruction is used in training only while not used in testing*.

Model	Test	△
Baseline (+DPs)	32.67	–
Baseline (+DPPs)	33.18	–
Separate-Recs⇒(+DPs)	34.02	+0.84
Separate-Recs⇒(+DPPs)	32.87	-0.31
Shared-Rec⇒(+DPs)	33.05	-0.13
Shared-Rec⇒(+DPPs)	**34.80**	**+1.62**

Table 6.4: Translation results using different types of DP.

Effect of Shared Reconstruction As mentioned previously, the effect of reconstruction is two-fold: 1) it improves the training of baseline parameters, which leads to better hidden representations that embed labelled DP information learned from the training data; and 2) it serves as a reranking metric in testing to measure the quality of DP translation. Table 6.3 lists translation results when the reconstruction model is used in training only. We can see that the proposed "Shared-Rec" model still outperforms both the strong baseline and the best "Separate-Rec" model reported in Chapter 5. This is encouraging since no extra resources and computation are introduced to online decoding, which makes the approach highly practical, *e.g.,* for translation in industrial applications.

Effect of DP and DPP As shown in Table 6.1, DP and DPP contain two different types of DP-related information: DP word and DP position. Here we compare translation performance using DP and DPP on various models and the results are shown in Table 6.4. For baseline models, inserting DP placeholders (*i.e.,* "<DP>") into training/testing data performs better than adding exact DP words (*e.g.,* "我"). The interesting finding is that the separate reconstruction models prefer the DP word while the proposed shared recon-

Model	Test	△
Baseline (+DPPs)	33.18	–
Separate-Recs⇒(+DPs)	35.08	+1.90
Shared-Rec$_{independent}$⇒(+DPPs)	35.88	+2.70
Shared-Rec$_{enc\rightarrow dec}$⇒(+DPPs)	**36.53**	+3.35
Shared-Rec$_{dec\rightarrow enc}$⇒(+DPPs)	35.99	+2.81

Table 6.5: Translation results using different attention strategies in the shared reconstructor (+Joint DP Predictor).

Models	Precision	Recall	F1-score
External	0.67	0.65	0.66
Joint	0.74	0.76	0.75

Table 6.6: Evaluation of DP prediction accuracy. "External" model is *separately* trained on DP-annotated data with external neural methods (Chapter 4), while "Joint" model is *jointly* trained with the NMT model (Section 6.2.2).

structors perform better when incorporating DP position information. This indicates that 1) different types of DPs represent soft or hard information, and 2) as the encoder and decoder share the same reconstruction component, position information is more generalized than word (surface) for shared representations.

Effect of Interactive Attention Among the variations of shared reconstructors in Table 6.5, we found that an interaction attention from encoder to decoder achieves the best performance, which is +3.35 BLEU points better than our baseline and +1.45 BLEU points better than the best separate reconstruction model. We attribute the superior performance of "Shared-Rec$_{enc\rightarrow dec}$" to the fact that the attention context over encoder representations embeds useful DP information, which can help to better attend to the representations of the corresponding pronouns in the decoder side.

DP Prediction Accuracy As shown in Table 6.6, the jointly learned model significantly outperforms the external one by 9% in F1-score. We attribute this to the useful contextual information embedded in the reconstructor representations, which are used to generate the

Model	Type	Errors		
		Com.	*Cor.*	*All*
BASE	Total	72	105	177
+ Joint (+DP Labeler)	Fixed	56	68	124
	New	21	17	38
	Total	37	54	91
+ Cross-Sentence Context	Fixed	62	84	146
	New	25	10	35
	Total	35	31	66

Table 6.7: Translation error statistics. "Com." denotes completeness errors, and "Cor." for correctness errors.

exact DP words.

Error Analysis We finally investigate how the proposed approaches improve the translation by human evaluation. We randomly select 550 sentences from the test set. As shown in Table 6.7, we count how many completeness errors (*e.g.,* under-translation) and correctness errors (*e.g.,* mistaken-translation) are fixed (*Fixed*) and newly generated (*New*) by our models.

About *completeness*, we found that 72 sentences were incompletely translated due to DPs, while 78% and 86% of these errors are fixed by the our end-to-end ("+ Joint (+DP Labeler)") and discourse-aware ("+ Cross-Sentence Context") models, respectively. We observe that most corrected translations become longer and well-structured by generating DPs. About *correctness*, we found that 105 words/phrases were translated into incorrect equivalents, resulting in quite different meanings in translations. Among them, 65% and 80% errors are solved by giving correct DPs provided by our models. However, we also observe that our systems brings relative 20% new errors. According to the anaylysis, we confirm that the improvement of our models come from alleviating completeness and correctness problems.

Model	Sub.	Obj.	Dum.	Total
Source	430	151	189	770
Human	541	199	232	972
Baseline	425	159	209	793
+ Joint (+DP Predictor)	483	177	214	874
+ Joint (+DP Labeler)	506	177	210	893
+ Cross-Sentence Context	515	181	207	903

Table 6.8: Number of pronouns in source sentence and generated translations.

Statistics of DP Generation In this experiment, we investigate how many DPs are recovered in the translation output. Table 6.8 lists the statistics on the test set that consists of 1,500 sentences. The source sentence contains 770 pronouns and human translation contains 972 pronouns, which indicate that 21% (*i.e.,* $(972 - 770)/972$) of pronouns in the source sentences are dropped. The translation generated by the baseline model contains 793 pronouns, which is nearly the same with the source sentence. This confirms the claim that translation of implicit pronouns cannot normally be reproduced. Explicitly modeling DP translation consistently improves the generation of pronouns, which indicates that the improved DP translation indeed contributes most to the performance improvement. Besides, it is relatively easier for most proposed models to recover dropped dummy pronouns. Because dummy pronoun usually depends on intra-sentential discourse information. Subject pronouns are more challenging due to their dependencies to inter-sentential discourse knowledge.

6.5 Summary

In this chapter, we proposed three effective approaches of translating DPs with NMT models: *shared* reconstructor, *jointly* learning and *cross-sentence* context to translate and predict DPs. Through experiments we verified that 1) shared reconstruction is helpful to share knowledge between the encoder and decoder; 2) joint learning of the DP prediction model indeed alleviates the error propagation problem by improving prediction accuracy; and 3) cross-sentence context is helpful to capture discourse information for DP prediction model.

Experiments show that the three approaches accumulatively improve translation performance. This chapter directly answers our third research question as described below:

> **RQ 4** *Can we build a fully end-to-end neural model for dropped pronoun translation? Is cross-sentence context useful for dropped pronoun prediction?*

Furthermore, the method is not restricted to the DP translation task and could potentially be applied to other sequence generation problems where additional source-side information could be incorporated. In the next chapter, we will conclude and present avenues for future research.

Chapter 7

Conclusion

In this chapter, we provide conclusions for the previous chapters and revisit the research questions with the answers we have provided to them. We then summarise the contributions of our work in this thesis. Later in this chapter, we explore various possibilities for further research.

7.1 Conclusion and Research Questions

In Chapter 1, we provided the motivations underpinning our study of discourse-aware NMT. By analyzing a discourse-level example, we discussed the errors in translation outputs caused by overlooking discourse information in MT models. We then presented our specific research questions, which can be divided into two parts: document-level NMT architecture and dealing with discourse phenomena for MT, as follows:

Part I: Building a Document-Level Architecture

> **RQ 1** *What is the influence of historical contextual information on the performance of neural machine translation? Can a document-level NMT architecture alleviate inconsistency and ambiguity problems?*

Part II: Targeting a Specific Discourse Phenomenon

> **RQ 2** *How do dropped pronouns affect the performance of machine translation? Is it possible to build a robust drop pronoun recovery model for statistical machine translation?*
>
> **RQ 3** *Does neural machine translation still suffer from dropped pronoun problems? If so, how should we embed DP information into neural network models?*
>
> **RQ 4** *Can we build a fully end-to-end neural model for dropped pronoun translation? Is cross-sentence context useful for dropped pronoun prediction?*

In Chapter 2, we first provided an overview of the MT models including SMT and NMT. Regarding SMT, we briefly introduced how the model is defined and translates sentences. For NMT, we reviewed word vector models, RNN models, neural LMs as well as the encoder-decoder architecture. Secondly, we provided basic information on discourse, including related theories, structures, and linguistic phenomena. Thirdly, we highlighted previous discourse-aware approaches along two lines: document-level NMT architecture and dealing with discourse phenomena for MT.

In Chapter 3, we addressed **RQ1** by presenting a novel document-level architecture for NMT models. As far as we know, this was the first attempt at investigating the potential for implicitly incorporating discourse information into NMT. More specifically, we employed a hierarchical RNN encoder to model cross-sentence context, and then integrated the historical summary into the standard NMT model. Through experiments on Chinese–English, we showed that our approach can significantly improve translation quality over the sentence-level NMT and SMT baseline models, especially in terms of consistency and disambiguity. We also analyzed the effect of global context, and provided examples generated by our model. Furthermore, we also compared our approach with other recently proposed DNMT models on various domains.

From Chapter 4, we began to move our attention to a specific discourse phenomenon: pro-drop, which significantly affects the performance of MT systems, especially in informal use-cases. In a Comparison with English pronouns, we first studied DPs in Chinese and Japanese languages. We then proposed an unsupervised approach to automatically build a large-scale and high-quality DP training corpus. Using this corpus, we trained neural-based DP generation models and integrated the recalled DPs into the SMT models. The experimental results on a Chinese–English subtitle corpus showed the effectiveness of our proposed approach. Case studies illustrated how our approach alleviates DP problems for translation models. To further validate the effectiveness of our model, we also adapt it to Japanese–English. Finally, we addressed **RQ2** in this chapter.

In Chapter 5, in order to address **RQ3**, we investigated DP translation for NMT models. First of all, we still explicitly and automatically annotated DPs for each source sentence in the training corpus using the method in Chapter 4. We then presented a reconstruction-based approach to guide the hidden states (either encoder-side or decoder-side) of NMT to embed the missing DP information. Experiments on the same corpora show that the proposed approach significantly outperforms a strong NMT baseline system. In our analysis, we also demonstrated that our models can produce better translations by addressing the DP translation problem.

Although the reconstruction-based models achieve improvements, there still exit some drawbacks such as the error propagation problem. Accordingly, we further discussed the fourth research question, **RQ4**, in Chapter 6. We exploited a fully end-to-end approach for DP translation in NMT models. Specifically, we employed a shared reconstructor to better exploit encoder and decoder representations. Secondly, we proposed to jointly learn to translate and predict DPs. To capture discourse information for DP prediction, we finally combined the hierarchical encoder with the DP translation model. Experimental results on a Chinese–English dialogue corpus show that our approach can accumulatively improve translation performance. In addition, the jointly learned DP prediction model significantly outperforms its external counterpart by 9% in terms of F1-score.

Chapter 7 concludes the thesis with general observations drawed from our experiments. We also provide some future avenues for research.

7.2 Contributions

In this thesis, we have investigated different discourse-aware approaches for MT models. We studied this research topic from two perspectives: document-level NMT architecture and dealing with discourse phenomena for MT. The contributions of our work can be summarized as follows:

- **Document-level NMT architecture**. Before our work, document-level NMT had received substantially less attention from the research community. In an early attempt, we investigated a novel document-level architecture for NMT models. We quantitatively and qualitatively demonstrated that the modeling of cross-sentence context can significantly outperform sentence-level NMT systems. We also systematically compared our model with other recently proposed DNMT models on various domains. We found that different document-level architectures perform unevenly on a distinct genre of texts. Finally, we released two versions of code for these experiments: `https://www.github.com/tuzhaopeng/LC-NMT` and `https://github.com/longyuewangdcu/Cross-Sentence-NMT`.

- **Dropped pronoun training data**. The first challenge for DP translation is that the existing data for training a robust DP generation model is very scarce. Thus, we proposed an automatic approach to DP annotation, which utilizes an alignment matrix from parallel data. Finally, we built a large DP training corpus with high consistency (over 90%) compared with the manual annotation method. We released the corpus in `https://github.com/longyuewangdcu/tvsub`. We believe the data can be also useful for other research fields such as discourse processing.

- **Neural dropped pronoun generation for SMT models**. Before our work, the related task such as empty categories and coreference resolution are trained on tradi-

tional models. Using our large DP training data, we built a robust DP generator using neural network models. We then integrated it into SMT models using various strategies. Experiments on both Chinese–English and Japanese-English translation tasks showed that 1) it is crucial to identify DPs to improve the overall translation performance; 2) although containing some noise (66% generation accuracy), the external DP generation model is still helpful to translation; and 3) the N-best DP integration strategy is able to alleviate the error propagation problem to a certain extent.

- **Dropped pronoun reconstructor for NMT models**. We exploit the first approach on DP translation for NMT models. We proposed a reconstruction-based model to guide hidden states in both the encoder and decoder to embed the DP information. Experiments and analysis show that the proposed approach significantly improves translation performance across different language pairs, and can be further improved by developing better DP generation models. We also enlarge the DP training corpus from 1M to 2M, and released the data in `https://github.com/longyuewangdcu/tvsub`.

- **A fully end-to-end DP translation model**. Although the reconstruction-based models achieve improvements, there still exit some drawbacks. To further improve the reconstruction-based model, we proposed three advanced approaches: *shared* reconstructor, *joint* learning, and *cross-sentence context*. Through experiments we verified that 1) shared reconstruction is helpful to share knowledge between the encoder and decoder; 2) joint learning of the DP prediction model indeed alleviates the error propagation problem; and 3) the cross-sentence model is able to capture useful discourse information for our DP prediction counterpart. Finally, the three approaches accumulatively improve translation performance. Note that the method is not restricted to the DP translation task and could potentially be applied to other sequence generation problems where additional source-side information could be incorporated.

7.3 Future Work

There are several possible extensions to the models presented in this thesis, and we summarize them as follows:

Document-level NMT Currently, our proposed document-level NMT architecture has two drawbacks: 1) the range of historical context is fixed once the model is built; and 2) it needs a lot of additional parameters when reading the more historical context. However, cache-based NMT (Tu et al. 2018) inspired us to improve DNMT in a different way. Tu et al. (2018) proposed to augment NMT models with a cache-like memory network, which stores the translation history in terms of bilingual hidden representations at decoding steps of previous sentences. Using simply a dot-product for key matching, this history information is quite cheap to store and can be accessed efficiently.

Based on cache-based NMT, we expect several developments that will shed more light on utilizing long-range contexts, *i.e.,* designing novel architectures, such as employing discourse relations instead of directly using decoder states as cache values.

Dropped Pronoun Translation To validate the robustness of our approach, we will extend our work to different genres and all kinds of dropped words. For example, in formal text genres (*e.g.,* newswire), DPs are not as common as in the informal text genres, and the most frequently dropped pronoun is the third person singular "它" ("*it*") (Baran et al. 2012), while this may not be crucial to translation performance in terms of BLEU score, it will harm cohesion in translated text.

Furthermore, we will investigate a new research strand that adapts our model in an inverse translation direction by learning to drop pronouns instead of recovering DPs.

Discourse-aware evaluation for MT The existing evaluation metrics operate only at the level of the sentence, which may not be precise enough to evaluate the performance of MT models when translating a complete text. In addition, BLEU score seems too simple

to reflect complicated discourse properties such coherence. As discussed by Läubli et al. (2018), there is a need to shift towards document-level evaluation as MT improves to the degree that errors which are hard or impossible to spot at the sentence-level become decisive in discriminating quality of different translation outputs.

It will be interesting to explore to what extent existing and future techniques for document-level MT can narrow this gap. We expect that this will require further efforts in creating document-level training data, designing appropriate models, and supporting research with discourse-aware automatic metrics.

Bibliography

Anastasopoulos, A. and Chiang, D. (2018). Tied multitask learning for neural speech translation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics*, pages 82–91, New Orleans, Louisiana, USA.

Asher, N. and Lascarides, A. (2003). *Logics of conversation*. Cambridge University Press.

Bahdanau, D., Cho, K., and Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate. In *Proceedings of the 3rd International Conference on Learning Representations*, pages 1–15, San Diego, California, USA.

Banerjee, S. and Lavie, A. (2005). METEOR: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Sydney, Australia.

Baran, E., Yang, Y., and Xue, N. (2012). Annotating dropped pronouns in Chinese newswire text. In *Proceedings of the 8th Language Resources and Evaluation Conference*, pages 2795–2799, Istanbul, Turkey.

Barzilay, R. and Elhadad, M. (1997). Using lexical chains for text summarization. In *Proceedings of the ACL Workshop on Intelligent Scalable Text Summarization*, pages 10–17, Madrid, Spain.

Bawden, R., Sennrich, R., Birch, A., and Haddow, B. (2018). Evaluating discourse phenomena in neural machine translation. In *Proceedings of the 2018 Conference of the*

North American Chapter of the Association for Computational Linguistics, New Orleans, Louisiana, USA.

Bengio, Y., Ducharme, R., Vincent, P., and Jauvin, C. (2003). A neural probabilistic language model. *Journal of Machine Learning Research*, 3:1137–1155.

Bengio, Y., Simard, P., and Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166.

Bergstra, J., Breuleux, O., Bastien, F., Lamblin, P., Pascanu, R., Desjardins, G., Turian, J., Warde-Farley, D., and Bengio, Y. (2010). Theano: A cpu and gpu math expression compiler in python. In *Proceedings of Python for Scientific Computing Conference*, pages 3–10, Austin, Texas, USA.

Bond, F. and Ogura, K. (1998). Reference in Japanese–English machine translation. *Machine Translation*, 13(2-3):107–134.

Bourlard, H. and Kamp, Y. (1988). Auto-association by multilayer perceptrons and singular value decomposition. *Biological Cybernetics*, 59(4-5):291–294.

Brown, P. F., Pietra, V. J. D., Pietra, S. A. D., and Mercer, R. L. (1993). The mathematics of statistical machine translation: Parameter estimation. *Computational linguistics*, 19(2):263–311.

Burkett, D., Petrov, S., Blitzer, J., and Klein, D. (2010). Learning better monolingual models with unannotated bilingual text. In *Proceedings of the 14th Conference on Computational Natural Language Learning*, pages 46–54, Uppsala, Sweden.

Cai, S., Chiang, D., and Goldberg, Y. (2011). Language-independent parsing with empty elements. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, pages 212–216, Portland, Oregon.

Calixto, I. and Liu, Q. (2017). Incorporating global visual features into attention-based

neural machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 992–1003, Copenhagen, Denmark.

Carpuat, M. and Simard, M. (2012). The trouble with smt consistency. In *Proceedings of the 7th Workshop on Statistical Machine Translation*, pages 442–449, Montreal, Quebec, Canada.

Cettolo, M., Girardi, C., and Federico, M. (2012). Wit3: Web inventory of transcribed and translated talks. In *Conference of the 2012 European Association for Machine Translation*, pages 261–268, Trento, Italy.

Chan, S. W. and T'sou, B. K. (1999). Semantic inference for anaphora resolution: Toward a framework in machine translation. *Machine Translation*, 14(3-4):163–190.

Chen, B. and Zhu, X. (2014). Bilingual sentiment consistency for statistical machine translation. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 607–615, Gothenburg, Sweden.

Chen, C. and Ng, V. (2012). Chinese noun phrase coreference resolution: Insights into the state of the art. In *Proceedings of the 24th International Conference on Computational Linguistics*, pages 185–194, Mumbai, India.

Chen, C. and Ng, V. (2013). Chinese zero pronoun resolution: Some recent advances. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1360–1365, Seattle, Washington, USA.

Cho, K., van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1724–1734, Doha, Qatar.

Choi, H., Cho, K., and Bengio, Y. (2017). Context-dependent word representation for neural machine translation. *Computer Speech & Language*, 45:149–160.

Chung, J., Gulcehre, C., Cho, K., and Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.

Chung, T. and Gildea, D. (2010). Effects of empty categories on machine translation. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 636–645, Cambridge, Massachusetts, USA.

Collins, M., Koehn, P., and Kucerova, I. (2005). Clause restructuring for statistical machine translation. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, pages 531–540, Ann Arbor, Michigan.

Crystal, D. (1985). *A Dictionary of Linguistics and Phonetics*. Oxford: Blackwell Publishers.

Dagan, I., Itai, A., and Schwall, U. (1991). Two languages are more informative than one. In *Proceedings of the 29th Annual Meeting on Association for Computational Linguistics*, pages 130–137, Berkeley, California, USA.

De Beaugrande, R. and Dressler, W. U. (1981). *Einführung in die Textlinguistik*, volume 28. Tübingen: Niemeyer.

DiMarco, C. and Mah, K. (1994). A model of comparative stylistics for machine translation. *Machine Translation*, 9(1):21–59.

Dong, D., Wu, H., He, W., Yu, D., and Wang, H. (2015). Multi-task learning for multiple language translation. In *Proceedings of the 53rd Annual Meeting of the Assocaition for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, pages 1723–1732, Beijing, China.

Elson, B. F. and Pickett, V. (1983). *Beginning morphology and syntax*. Summer Inst of Linguistics.

Fakoor, R., Mohamed, A.-r., Mitchell, M., Kang, S. B., and Kohli, P. (2016). Memory-augmented attention modelling for videos. *arXiv preprint arXiv:1611.02261*.

Feng, V. W. and Hirst, G. (2012). Text-level discourse parsing with rich linguistic features. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, pages 60–68, Jeju, Korea.

Ferrández, A., Palomar, M., and Moreno, L. (1999). An empirical approach to spanish anaphora resolution. *Machine Translation*, 14(3-4):191–216.

Firat, O., Cho, K., and Bengio, Y. (2016). Multi-way, multilingual neural machine translation with a shared attention mechanism. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics*, pages 866–875, San Diego, California, USA.

Foster, G., Isabelle, P., and Kuhn, R. (2010). Translating structured documents. In *Proceedings of the 9th Conference of the Association for Machine Translation in the Americas*, Denver, Colorado, USA.

Galley, M. and Manning, C. D. (2008). A simple and effective hierarchical phrase reordering model. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 848–856, Waikiki, Honolulu, USA.

Galley, M. and McKeown, K. (2003). Improving word sense disambiguation in lexical chaining. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence*, volume 3, pages 1486–1488, Acapulco, Mexico.

Gong, Z., Zhang, M., and Zhou, G. (2011). Cache-based document-level statistical machine translation. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 909–919, Edinburgh, Scotland, UK.

Graves, A., Wayne, G., and Danihelka, I. (2014). Neural turing machines. *arXiv preprint arXiv:1410.5401*.

Gu, J., Wang, Y., Cho, K., and Li, V. O. (2017). Search engine guided non-parametric neural machine translation. *arXiv preprint arXiv:1705.07267*.

Guillou, L. (2013). Analysing lexical consistency in translation. In *Proceedings of the Workshop on Discourse in Machine Translation*, pages 10–18, Sofia, Bulgaria.

Halliday, M. A. K. and Hasan, R. (1976). Cohesion in english. *Longman*.

Hanks, W. F. (1987). Discourse genres in a theory of practice. *American Ethnologist*, 14(4):668–692.

Hardmeier, C. (2014). *Discourse in statistical machine translation*. PhD thesis, Acta Universitatis Upsaliensis.

Hardmeier, C., Nivre, J., and Tiedemann, J. (2012). Document-wide decoding for phrase-based statistical machine translation. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1179–1190, Jeju, Korea.

Hasler, E., Blunsom, P., Koehn, P., and Haddow, B. (2014). Dynamic topic adaptation for phrase-based MT. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 328–337, Gothenburg, Sweden.

Haspelmath, M. (2001). The European linguistic area: standard average European. In *Language typology and language universals*, volume 2, pages 1492–1510. Berlin: de Gruyter.

He, D., Xia, Y., Qin, T., Wang, L., Yu, N., Liu, T., and Ma, W.-Y. (2016a). Dual learning for machine translation. In *Proceedings of the 2016 Annual Conference on Neural Information Processing Systems*, pages 820–828, Barcelona, Spain.

He, W., He, Z., Wu, H., and Wang, H. (2016b). Improved neural machine translation with smt features. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence*, pages 151–157, Phoenix, Arizona, USA.

Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. R.

(2012). Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*.

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

Huang, C.-T. J. (1984). On the distribution and reference of empty pronouns. *Linguistic Inquiry*, 15(4):531–574.

Jean, S., Cho, K., Memisevic, R., and Bengio, Y. (2014). On using very large target vocabulary for neural machine translation. *arXiv preprint arXiv:1412.2007*.

Jean, S., Lauly, S., Firat, O., and Cho, K. (2017). Does neural machine translation benefit from larger context? *arXiv preprint arXiv:1704.05135*.

Kaiser, Ł., Nachum, O., Roy, A., and Bengio, S. (2017). Learning to remember rare events. *arXiv preprint arXiv:1703.03129*.

Kalchbrenner, N. and Blunsom, P. (2013). Recurrent continuous translation models. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1700–1709, Seattle, Washington, USA. Association for Computational Linguistics.

Knight, K. (1999). Decoding complexity in word-replacement translation models. *Computational Linguistics*, 25(4):607–615.

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

Koehn, P., Axelrod, A., Birch, A., Callison-Burch, C., Osborne, M., Talbot, D., and White, M. (2005). Edinburgh system description for the 2005 IWSLT speech translation evaluation. In *Proceedings of the 12th International Workshop on Spoken Language Translation*, pages 68–75, Da Nang, Vietnam.

Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., and Herbst, E.

(2007). Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*, pages 177–180, Prague, Czech Republic.

Koehn, P. and Knowles, R. (2017). Six challenges for neural machine translation. In *Proceedings of the 1st Workshop on Neural Machine Translation*, pages 28–39, Vancouver, Canada.

Koehn, P., Och, F. J., and Marcu, D. (2003). Statistical phrase-based translation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics*, pages 48–54, Edmonton, Canada.

Kong, F. and Zhou, G. (2010). A tree kernel-based unified framework for chinese zero anaphora resolution. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 882–891, Cambridge, Massachusetts, USA.

Kong, F. and Zhou, G. (2012). Exploring local and global semantic information for event pronoun resolution. In *Proceedings of the 24th International Conference on Computational Linguistics*, pages 1475–1488, Mumbai, India.

Kuang, S., Xiong, D., Luo, W., and Zhou, G. (2017). Cache-based document-level neural machine translation. *arXiv preprint arXiv:1711.11221*.

Läubli, S., Sennrich, R., and Volk, M. (2018). Has machine translation achieved human parity? a case for document-level evaluation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4791–4796, Brussels, Belgium.

Le Nagard, R. and Koehn, P. (2010). Aiding pronoun translation with co-reference resolution. In *Proceedings of the Joint 5th Workshop on Statistical Machine Translation and MetricsMATR*, pages 252–261, Uppsala, Sweden.

Li, C., Wu, Y., Wu, W., Xing, C., Li, Z., and Zhou, M. (2016). Detecting context dependent

messages in a conversational environment. In *Proceedings of the 26th International Conference on Computational Linguistics*, pages 1990–1999, Osaka, Japan.

Li, C. N. and Thompson, S. A. (1989). *Mandarin Chinese: A functional reference grammar*. University of California Press, Oakland, California, USA.

Li, J., Xiong, D., Tu, Z., Zhu, M., Zhang, M., and Zhou, G. (2017). Modeling source syntax for neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pages 688–697, Vancouver, Canada.

Lin, Z., Ng, H. T., and Kan, M.-Y. (2014). A PDTB-styled end-to-end discourse parser. *Natural Language Engineering*, 20(02):151–184.

Lison, P. and Tiedemann, J. (2016). Opensubtitles2016: Extracting large parallel corpora from movie and TV subtitles. In *Proceedings of the 10th International Conference on Language Resources and Evaluation*, pages 923–929, Portorož, Slovenia.

Liu, Q., Tu, Z., and Lin, S. (2013). A novel graph-based compact representation of word alignment. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 358–363, Sofia, Bulgaria.

Liu, Y., Xia, T., Xiao, X., and Liu, Q. (2009). Weighted alignment matrices for statistical machine translation. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 1017–1026, Singapore.

Longacre, R. E. (1990). *Storyline concerns and word order typology in East and West Africa*, volume 10. Los Angeles: African Studies Center, UCLA.

Longacre, R. E. (2013). *The grammar of discourse*. Springer Science & Business Media.

Luong, M.-T., Le, Q. V., Sutskever, I., Vinyals, O., and Kaiser, L. (2016). Multi-task sequence to sequence learning. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2965–2977, Santa Fe, New Mexico, USA.

Luong, T., Pham, H., and Manning, D. C. (2015a). Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal.

Luong, T., Sutskever, I., Le, Q., Vinyals, O., and Zaremba, W. (2015b). Addressing the rare word problem in neural machine translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, pages 11–19, Beijing, China.

Mann, W. C. and Thompson, S. A. (1988). Rhetorical structure theory: Toward a functional theory of text organization. *Text-Interdisciplinary Journal for the Study of Discourse*, 8(3):243–281.

Marcu, D. (2000). *The theory and practice of discourse parsing and summarization*. MIT press.

Marcu, D., Carlson, L., and Watanabe, M. (2000). The automatic translation of discourse structures. In *Proceedings of the 1st North American chapter of the Association for Computational Linguistics conference*, pages 9–17, Seattle, Washington, USA.

Maruf, S. and Haffari, G. (2017). Document context neural machine translation with memory networks. *arXiv preprint arXiv:1711.03688*.

Mesnil, G., He, X., Deng, L., and Bengio, Y. (2013). Investigation of recurrent-neural-network architectures and learning methods for spoken language understanding. In *Proceedings of the 14th Annual Conference of the International Speech Communication Association*, pages 3771–3775, Lyon, France.

Meyer, T. and Poláková, L. (2013). Machine translation with many manually labeled discourse connectives. In *Proceedings of the 1st Workshop on Discourse in Machine Translation*, pages 43–50, Sofia, Bulgaria.

Meyer, T. and Webber, B. (2013). Implicitation of discourse connectives in (machine)

translation. In *Proceedings of the 1st Workshop on Discourse in Machine Translation*, pages 19–26, Sofia, Bulgaria.

Miculicich, L., Ram, D., Pappas, N., and Henderson, J. (2018). Document-level neural machine translation with hierarchical attention networks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2947–2954, Brussels, Belgium.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013a). Distributed representations of words and phrases and their compositionality. In *Proceedings of the 27th Annual Conference on Neural Information Processing Systems*, pages 3111–3119, Lake Tahoe, Nevada, USA.

Mikolov, T., Yih, W.-t., and Zweig, G. (2013b). Linguistic regularities in continuous space word representations. In *roceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics*, volume 13, pages 746–751, Atlanta, Georgia, USA.

Miller, A., Fisch, A., Dodge, J., Karimi, A.-H., Bordes, A., and Weston, J. (2016). Key-value memory networks for directly reading documents. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1400–1409, Austin, Texas, USA.

Mitkov, R. (1993). How could rhetorical relations be used in machine translation? *Intentionality and structure in discourse relations*.

Nakaiwa, H. (1999). Automatic extraction of rules for anaphora resolution of Japanese zero pronouns in Japanese–English machine translation from aligned sentence pairs. *Machine Translation*, 14(3-4):247–279.

Nakamura, M. (1987). Japanese as a pro language. *The Linguistic Review*, 6:281–296.

Nirenburg, S., Raskin, V., and Tucker, A. (1986). On knowledge-based machine translation. In *Proceedings of the 11th coference on Computational linguistics*, pages 627–632, Bonn, Germany.

Novischi, A. and Moldovan, D. (2006). Question answering with lexical chains propagating verb arguments. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics*, pages 897–904, Sydney, Australia.

Och, F. J. (2003). Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*, pages 160–167, Sapporo, Japan.

Och, F. J. and Ney, H. (2002). Discriminative training and maximum entropy models for statistical machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 295–302, Philadelphia, USA.

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

Och, F. J. and Ney, H. (2004). The alignment template approach to statistical machine translation. *Computational linguistics*, 30(4):417–449.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). BLEU: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA.

Powers, D. M. (2011). Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation. Bioinfo Publications.

Pradhan, S., Moschitti, A., Xue, N., Uryupina, O., and Zhang, Y. (2012). CoNLL-2012 shared task: Modeling multilingual unrestricted coreference in ontonotes. In *Proceedings*

of the 15th Conference on Computational Natural Language Learning: Shared Task, pages 1–27, Jeju Island, Korea.

Pritzel, A., Uria, B., Srinivasan, S., Puigdomenech, A., Vinyals, O., Hassabis, D., Wierstra, D., and Blundell, C. (2017). Neural episodic control. *arXiv preprint arXiv:1703.01988*.

Quirk, R., Greebaum, S., Leech, G., and Svartvik, J. (1985). *A Comprehensive Grammar of the English Language*, volume 9. New York: Longman.

Raymond, C. and Riccardi, G. (2007). Generative and discriminative algorithms for spoken language understanding. In *Proceedings of 8th Annual Conference of the International Speech Communication Association*, pages 1605–1608, Antwerp, Belgium.

Robbins, H. and Monro, S. (1985). A stochastic approximation method. In *Herbert Robbins Selected Papers*, pages 102–109. Springer.

Rosti, A.-V. I., Ayan, N. F., Xiang, B., Matsoukas, S., Schwartz, R. M., and Dorr, B. J. (2007). Combining outputs from multiple machine translation systems. In *Proceedings of the Human Language Technology and the 6th Meeting of the North American Chapter of the Association for Computational Linguistics*, pages 228–235, Rochester, NY, USA.

Rumelhart, D. E., Durbin, R., Golden, R., and Chauvin, Y. (1995). Backpropagation: The basic theory. *Backpropagation: Theory, Architectures and Applications*, pages 1–34.

Sammer, M., Reiter, K., Soderland, S., Kirchhoff, K., and Etzioni, O. (2006). Ambiguity reduction for machine translation: Human-computer collaboration. In *Proceedings of the 5th Conference of the Association for Machine Translation in the Americas*, Cambridge, Massachusetts, USA.

Sanders, T. and Maat, H. P. (2006). Cohesion and coherence: Linguistic approaches. *Reading*, 99:440–466.

Schuster, M. and Paliwal, K. K. (1997). Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681.

Sennrich, R., Firat, O., Cho, K., Birch, A., Haddow, B., Hitschler, J., Junczys-Dowmunt, M., Läubli, S., Barone, A. V. M., Mokry, J., et al. (2017). Nematus: a toolkit for neural machine translation. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, pages 65–68, Valencia, Spain.

Sennrich, R., Haddow, B., and Birch, A. (2016). Edinburgh neural machine translation systems for WMT 16. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, volume 2, pages 371–376, Berlin, Germany.

Serban, I. V., Sordoni, A., Bengio, Y., Courville, A., and Pineau, J. (2016). Building end-to-end dialogue systems using generative hierarchical neural network models. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence*, pages 3776–3783, Phoenix, Arizona.

Snover, M., Dorr, B., Schwartz, R., Micciulla, L., and Makhoul, J. (2006). A study of translation edit rate with targeted human annotation. In *Proceedings of the 5th Conference of the Association for Machine Translation in the Americas*, Cambridge, Massachusetts, USA.

Socher, R., Pennington, J., Huang, E. H., Ng, A. Y., and Manning, C. D. (2011). Semi-supervised recursive autoencoders for predicting sentiment distributions. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 151–161, Edinburgh, Scotland, UK.

Sordoni, A., Bengio, Y., Vahabi, H., Lioma, C., Simonsen, J. G., and Nie, J. (2015). A hierarchical recurrent encoder-decoder for generative context-aware query suggestion. In *Proceedings of the 24th ACM International Conference on Information and Knowledge Management*, pages 553–562, Melbourne, Australia.

Soricut, R. and Marcu, D. (2003). Sentence level discourse parsing using syntactic and lexical information. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics*, pages 149–156, Edmonton, Canada.

Stolcke, A. (2002). SRILM - an extensible language modeling toolkit. In *Proceedings of the 7th International Conference on Spoken Language Processing*, pages 901–904, Colorado, USA.

Stolcke, A., Konig, Y., and Weintraub, M. (1997). Explicit Word Error Minimization in N-best List Rescoring. In *The 5th European Conference on Speech Communication and Technology*, Rhodes, Greece.

Su, J., Wu, H., Wang, H., Chen, Y., Shi, X., Dong, H., and Liu, Q. (2012). Translation model adaptation for statistical machine translation with monolingual topic information. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, pages 459–468, Jeju, Korea.

Sukhbaatar, S., Weston, J., Fergus, R., et al. (2015). End-to-end memory networks. In *Proceedings of Neural Information Processing Systems*, pages 2440–2448, Montreal, Canada.

Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *Proceedings of the 2014 Neural Information Processing Systems*, pages 3104–3112, Montreal, Canada.

Tai, K. S., Socher, R., and Manning, C. D. (2015). Improved semantic representations from tree-structured long short-term memory networks. *arXiv preprint arXiv:1503.00075*.

Taira, H., Sudoh, K., and Nagata, M. (2012). Zero pronoun resolution can improve the quality of j-e translation. In *Proceedings of the 6th Workshop on Syntax, Semantics and Structure in Statistical Translation*, pages 111–118, Jeju, Republic of Korea.

Tiedemann, J. (2010). Context adaptation in statistical machine translation using models with exponentially decaying cache. In *Proceedings of the 2010 Workshop on Domain Adaptation for Natural Language Processing*, pages 8–15, Uppsala, Sweden.

Tu, M., Zhou, Y., and Zong, C. (2013). A novel translation framework based on rhetorical structure theory. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 370–374, Sofia, Bulgaria.

Tu, Z., Liu, Y., Hwang, Y.-S., Liu, Q., and Lin, S. (2010). Dependency forest for statistical machine translation. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 1092–1100, Beijing, China.

Tu, Z., Liu, Y., Liu, Q., and Lin, S. (2011). Extracting Hierarchical Rules from a Weighted Alignment Matrix. In *Proceedings of the 5th International Joint Conference on Natural Language Processing*, pages 1294–1303, Chiang Mai, Thailand.

Tu, Z., Liu, Y., Lu, Z., Liu, X., and Li, H. (2017a). Context gates for neural machine translation. *Transactions of the Association of Computational Linguistics*, 5(1):87–99.

Tu, Z., Liu, Y., Shang, L., Liu, X., and Li, H. (2017b). Neural machine translation with reconstruction. In *Proceedings of the 31th AAAI Conference on Artificial Intelligence*, pages 3097–3103, San Francisco, California, USA.

Tu, Z., Liu, Y., Shi, S., and Zhang, T. (2018). Learning to remember translation history with a continuous cache. *Transactions of the Association of Computational Linguistics*, 6:407–420.

Tu, Z., Lu, Z., Liu, Y., Liu, X., and Li, H. (2016). Modeling coverage for neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 76–85, Berlin, Germany.

Vasconcellos, M. (1989). Cohesion and coherence in the presentation of machine translation products. *Georgetown University Round Table on Languages and Linguistics*, pages 89–105.

Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., and Manzagol, P.-A. (2010). Stacked

denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research*, 11(Dec):3371–3408.

Vinyals, O. and Le, Q. (2015). A neural conversational model. *arXiv preprint arXiv:1506.05869*.

Voita, E., Serdyukov, P., Sennrich, R., and Titov, I. (2018). Context-aware neural machine translation learns anaphora resolution. *arXiv preprint arXiv:1805.10163*.

Wada, H. (1990). Discourse processing in MT: problems in pronominal translation. In *Proceedings of the 13th conference on Computational linguistics*, pages 73–75, Helsinki, Finland.

Wang, L., Lu, Y., Wong, D. F., Chao, L. S., Wang, Y., and Oliveira, F. (2014). Combining domain adaptation approaches for medical text translation. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 254–259.

Wang, L., Zhang, X., Tu, Z., Way, A., and Liu, Q. (2016). The automatic construction of discourse corpus for dialogue translation. In *Proceedings of the 10th Language Resources and Evaluation Conference*, Portorož, Slovenia.

Wang, X., Lu, Z., Tu, Z., Li, H., Xiong, D., and Zhang, M. (2017). Neural machine translation advised by statistical machine translation. In *Proceedings of the 31th AAAI Conference on Artificial Intelligence*, pages 3330–3336, San Francisco, California, USA.

Wang, X., Tu, Z., Xiong, D., and Zhang, M. (2017). Translating phrases in neural machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1432–1442, Copenhagen, Denmark.

Weaver, W. (1949). The mathematics of communication. *Scientific American*, 181(1):11–15.

Webber, B. (2014). Discourse for machine translation. In *Proceedings of the 28th Pacific Asia Conference on Language, Information and Computing*, Phuket, Thailand.

Welo, E. (2013). Null anaphora. *Encyclopedia of Ancient Greek Language and Linguistics*.

Weston, J., Chopra, S., and Bordes, A. (2014). Memory networks. *arXiv preprint arXiv: arXiv:1410.3916*.

Widdowson, H. G. (1979). Explorations in applied linguistics. *Studies in Second Language Acquisition*.

Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., et al. (2016). Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Xiang, B., Luo, X., and Zhou, B. (2013). Enlisting the ghost: Modeling empty categories for machine translation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 822–831, Sofia, Bulgaria.

Xiao, T., Zhu, J., Yao, S., and Zhang, H. (2011). Document-level consistency verification in machine translation. In *Proceedings of the 13th Machine Translation Summit*, volume 13, pages 131–138, Xiamen, China.

Xiong, D., Ben, G., Zhang, M., Lv, Y., and Liu, Q. (2013). Modeling lexical cohesion for document-level machine translation. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence*, Beijing, China.

Xiong, H., He, Z., Wu, H., and Wang, H. (2018). Modeling coherence for discourse neural machine translation. *arXiv preprint arXiv:1811.05683*.

Xue, N., Ng, H. T., Pradhan, S., Prasad, R., Bryant, C., and Rutherford, A. (2015). The CoNLL-2015 shared task on shallow discourse parsing. In *Proceedings of the 19th Conference on Computational Natural Language Learning-Shared Task*, pages 1–16, Beijing, China.

Xue, N., Ng, H. T., Pradhan, S., Rutherford, A., Webber, B., Wang, C., and Wang, H. (2016). CoNLL 2016 shared task on multilingual shallow discourse parsing. In *Proceed-*

ings of the 20th Conference on Computational Natural Language Learning-Shared Task, pages 1–19, Berlin, Germany.

Xue, N., Xia, F., Chiou, F.-D., and Palmer, M. (2005). The Penn Chinese Treebank: Phrase structure annotation of a large corpus. *Natural Language Engineering*, 11(02):207–238.

Xue, N. and Yang, Y. (2013). Dependency-based empty category detection via phrase structure trees. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics*, pages 1051–1060, Atlanta, Georgia, USA.

Yang, X., Su, J., and Tan, C. L. (2006). Kernel-based pronoun resolution with structured syntactic knowledge. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, Sydney, Australia.

Yang, Y., Liu, Y., and Xu, N. (2015). Recovering dropped pronouns from Chinese text messages. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 309–313, Beijing, China.

Yang, Y. and Xue, N. (2010). Chasing the ghost: recovering empty categories in the Chinese treebank. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 1382–1390, Beijing, China.

Zeiler, M. D. (2012). Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*.

Zhang, J., Luan, H., Sun, M., Zhai, F., Xu, J., Zhang, M., and Liu, Y. (2018). Improving the transformer translation model with document-level context. *arXiv preprint arXiv:1810.03581*.

Zhang, J., Zong, C., et al. (2015). Deep neural networks in machine translation: An overview. *IEEE Intelligent Systems*, 30(5):16–25.

Zhao, S. and Ng, H. T. (2007). Identification and resolution of Chinese zero pronouns: A machine learning approach. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 541–550, Prague, Czech Republic.

Zhou, H., Tu, Z., Huang, S., Liu, X., Li, H., and Chen, J. (2017). Chunk-based bi-scale decoder for neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pages 580–586, Vancouver, Canada.

Zoph, B. and Knight, K. (2016a). Multi-source neural translation. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics*, pages 30–34, San Diego, California, USA.

Zoph, B. and Knight, K. (2016b). Multi-source neural translation. *arXiv preprint arXiv:1601.00710*.