The 34th Conference on Computational Linguistics and Speech Processing (ROCLING 2022)
Taipei, Taiwan, November 21-22, 2022. The Association for Computational Linguistics and Chinese Language Processing

運用少量語料於漢字轉客文字之類神經機器翻譯系統初步探討
A Preliminary Study on Mandarin-Hakka neural machine translation using small-sized data

洪翌翔 **Yi-Hsiang Hung,** 黃奕欽 **Yi-Chin Huang**
國立屏東大學電腦科學與人工智慧學系
Department of Computer Science and Artificial Intelligence
National Pingtung University
gbaian10@gmail.com, ychuangnptu@mail.nptu.edu.tw

摘要

在本研究中，我們利用注意力機制 (Attention) 與卷積 (Convolution) 模型的架構來實作一套中文轉四縣腔客文的機器翻譯系統。其中，爲了解決南北四縣腔的常用詞差異問題，我們透過語料的統計整理與詞典的定義方式，將兩種腔調的差異用法獨立出來，並分別訓練翻譯模型。

此外，爲了解決客語語料數量稀少而導致翻譯時遇到未知詞的狀況，在我們的模型中，透過實驗的驗證尋找適當的閥值以拒絕掉不適合的翻譯結果，並透過中文詞的替換以及客文常用詞的強制斷詞方式，讓最終的翻譯句獲得更佳的結果。最終，本研究所開發的系統可在少量語料下達到不錯的翻譯結果，並可應用於在地化的客語教學以及作爲中客文夾雜之語音合成系統的前端部分。

Abstract

In this study, we implemented a machine translation system using the Convolutional Neural Network with Attention mechanism for translating Mandarin to Sixan-accent Hakka. Specifically, to cope with the different idioms or terms used between Northern and Southern Sixan-accent, we analyzed the corpus differences and lexicon definition, and then separated the various word usages for training exclusive models for each accent.

Besides, since the collected Hakka corpora are relatively limited, the unseen words frequently occurred during real-world translation. In our system, we selected suitable thresholds for each model based on the model verification to reject non-suitable translated words. Then, by applying the proposed algorithm, which adopted the forced Hakka idioms/terms segmentation and the common Mandarin word substitution, the resultant translation sentences become more intelligible. Therefore, the proposed system achieved promising results using small-sized data. This system could be used for Hakka language teaching and also the front-end of Mandarin and Hakka code-switching speech synthesis systems.

關鍵字：神經機器翻譯、中文翻譯客文、南北四縣腔

Keywords: Neural Machine Translation, Mandarin to Hakka translation, Northern/Southern Sixian-accent

1 緒論

一個語言的傳承與推廣可以經由聽、說、讀、寫四個部份，而類似客家語、閩南語這種地方方言很容易出現會聽會說但卻不會讀文字或拼音和完全不會寫的窘境，而這樣就容易導致一個語言最後容易只能口耳相傳，導致自學不易。所以如何把一句想說的話從中文轉成客文並能唸出就是一個問題。本文主要在研究將中文翻譯成四縣腔客語的方法，並在細分成南四縣腔與北四縣腔。

客家語是台灣地方方言中使用量第二多的語言，僅次於閩南語。而台灣客家語又可分成各種腔調，依照使用比例由高至低分別爲四縣腔、海陸腔、大埔腔、饒平腔、詔安腔 (WillPete, 2022)，其中四縣腔是台灣所有客家語腔調中使用率最高的，且例如大眾運輸中的廣播系統使用的客語腔調正是四縣腔，故母語非客語者通常會選擇四縣腔做第一個選擇。四縣腔又因各地區不同並隨著時間的發展差異，又可分爲北四縣腔與南四縣腔。北四縣腔的使用人口主要分佈在苗栗多數鄉鎮，以及新竹和桃園的部份鄉鎮；南四縣腔的使用人口主要分佈在高雄和屏東的六堆地區 (臺灣教育部, 2018)。

雖然南北四縣兩腔調存在在部份使用的詞彙不同，或者音韻上差異，但基本上兩腔調同屬四縣腔，所以日常中大致上是可以直接進行溝通不會有太大問題。此處將舉兩個簡單的句子當作範例參考兩腔調的用詞差異，如表1。

中文句子	外面很涼	冬至吃湯圓
北四縣翻譯	外背當涼	冬節食雪圓仔
南四縣翻譯	外背蓋涼	冬至食圓粄仔

Table 1: 南北四縣翻譯差

從表1也可看出南北四縣腔整體句子架構高度相似，只有部份用詞上有差別，例如上述例子中的「湯圓」，北四縣翻譯成「雪圓仔」，南四縣翻譯成「圓粄仔」。除了句子架構相似之外，南北四縣腔所使用的拼音系統也相同[1]，所以多數語料都是可直接共用，但如果要做出兩者差異，則必須將兩者不同之處的透過分析與設計語料，並經由訓練機器翻譯模型使其產生不同腔調的翻譯。

近年來由於機器學習竄紅，而在機器翻譯領域中應用這類的方法稱爲神經機器翻譯 (Neural Machine Translation，NMT)，本論文將利用神經機器翻譯來處理中文文字轉換爲客文文字的問題。

本篇論文的架構如下：第一章爲緒論，描述研究背景，其中包含南北四縣腔的差異，以及將使用深度學習方法來解此翻譯問題。第二章爲相關研究，說明在機器翻譯上別人有使用哪些方法，各有什優缺點。第三章爲語料庫，詳細描述使用了哪些語料庫，各有多少資料量，並對這些資料分別做了哪些處理以符合自身研究的要求。第四章爲研究方法，說明以使用 Fairseq-CNN 作爲基礎，並用何種方法解決在資料量稀疏的情況下處理未知詞的問題。第五章爲實驗結果分析。第六章爲本文的結論。

2 相關研究

機器翻譯是用電腦將文字從一種語言翻譯成另一種語言的過程，而無需額外的人力。機器翻譯從過去需要語言學家來制定各種規則來逐字翻譯，到後來開始使用大量資料來做統計翻譯，最後演變到現在的神經機器翻譯不再是簡單的逐字翻譯，機器會分析所有文字元素並識別字詞間的相互影響方式，並將原始語言透過神經網路轉換成目標語言。

神經網路機器翻譯通常是基於序列到序列 (sequence-to-sequence，seq2seq) 做處理，seq2seq 主要就是分成編碼器 (encoder) 和解碼器 (decoder) 兩部份，編碼器負責將來源語言編碼成一個具有表示原本句子意義的隱含向量做訓練，最後在經由解碼器解碼成目標語言的文字。

[1]南北四縣腔的聲調都使用去聲 55、陰平 24、陽平 11、上聲 31、陰入 2、陽入 5，其他腔調例如海陸腔則不同

2.1 機器翻譯

較早期的機器翻譯方法有以下幾種，第一種方法是基於規則的字對字機器翻譯 (Rule-based Machine Translation，RBMT) (Forcada et al., 2011)，這種方法主要預先準備雙語字典、一些單字的規則 (例如-er、-est 等等字尾含意)，這種翻譯通常蛤需要該語言的專業語言學家制定各種詳細的規則，但一些語法結構上的問題依然很容易無法處理很多狀況。例如：臺灣大學陳信希 (Lin and Chen, 1999) 等老師的中文到台語翻譯、聯合大學黃豐隆教授 (Lin et al., 2014) 等老師的中文到客語、Charoenpornsawat 等人的英文到泰文 (Charoenpornsawat et al., 2002) 都使用此類方法。

第二種方法是基於例子的機器翻譯 (Example-Based Machine Translation，EBMT) (Somers, 1999), (Chunyu et al., 2002)，這種方法主要預先準備好大量已經翻譯好的句子來提供比對，例如句子「我今天在學校吃中餐」，如果現在要翻譯的句子成「我今天在學校吃晚餐」，經過比對後發現與前面中餐的句子最爲相近，只有一個詞有差異，則將不一樣的詞替換掉後就翻譯完成了。這種方法的優點在於只要準備好大量已經翻譯好的句子就能夠更容易翻譯出好的結果，而不用像 RBMT 一樣設計了多個規則，但還是有無法處理的狀況，然而此方法若要處理不存在資料庫中的句子時，依然會出現不合理的翻譯結果。例如：Ayu (Ayu and Mantoro, 2011) 等人使用 EBMT 將印尼語翻譯成英語。

第三種方法就是統計機器翻譯 (Statistical Machine Translation，SMT) (Koehn, 2009)，相較於前兩種方式是透過語言學的知識設計或定義相關的規則，而產生翻譯的結果，統計式機器翻譯是採用大量語料庫來進行機器學習，這種方法只要有大量資料就可進行機器翻譯，這種翻譯可統計詞的用量和基於前後文字來做的各種不同翻譯，例如 bank 是銀行還是河岸，可通過前後文來做個推測。而有不少的統計機器翻譯都是使用基於短語的機器翻譯 (Zens et al., 2002)，例如：Google 在 2006 年 4 月時候的宣佈未來 Google 翻譯將改使用 SMT 的翻譯系統[2]。基於短語 (Phrase-Based) 的翻譯結果相較於較早期的基於規則 (Rule-Based) 的方法已經進步許多，然而因爲是以短語爲單位在做翻譯，這些短語拼湊出來的句子翻譯依然不夠自然。

近年來開始出現神經機器翻譯 (Neural Machine Translation，NMT) (Bahdanau et al.,

[2]https://ai.googleblog.com/2006/04/statistical-machine-translation-live.html

2014)，相較於 RBMT 和 EMBT 等兩種傳統方法，NMT 不需要太多該語言領域的相關知識，也不需要額外準備如雙語字典、大量例句來幫助翻譯，因爲若這些資料量不足會大大影響翻譯結果的好壞，NMT 是僅靠大量平行語料來做訓練，在資料取得上會容易許多。而對比 SMT，NMT 是一次翻譯整個句子而不是切成較短語來做翻譯，這樣在翻譯上更容易考慮句子的前後關係，進而翻譯出更順暢的句子。NMT 也幫助各語言之間更容易直接互相翻譯，以前 Google 翻譯會先將源語言翻譯成英文，然後將英文翻譯成目標語言，而不是直接從一種語言翻譯成另一種語言。例如:Google 機器翻譯系統 (Wu et al., 2016) 於 2016 年開始逐步將多個語言慢慢從 SMT 改成使用 NMT，並通過應用基於實例的 (EBMT) 機器翻譯來改善結果。

2.2 Seq2Seq

近年來在處理文字翻譯這種具有時間順序關係上的資料時候經常使用 seq2seq 架構 (Sutskever et al., 2014)。不只在機器翻譯上，近年來 seq2seq 在自然語言處理 (Natural Language Processing，NLP) 領域中如：語音識別 (Chorowski et al., 2015)、文本摘要 (Rush et al., 2015; Nallapati et al., 2016) 等都取得了良好的結果。

seq2seq 主要由編碼器和解碼器所組成，當一串文字丟入編碼器經過編碼轉換成一個固定長度的隱含內文向量 (context vector)，最後這個向量在經由解碼器轉換回人類所看的語言。而一般編碼器和解碼器內部通常由循環神經網路 (Recurrent Neural Network，RNN)、長短期記憶 (Long Short-Term Memory，LSTM) (Hochreiter and Schmidhuber, 1997)、門控循環單元 (Gated Recurrent Unit，GRU) (Cho et al., 2014) 等這類以 RNN 爲基礎的循環神經網路做處理。然而上述的編碼器解碼器架構有個致命的問題，就是他不管任何長度的原始內容壓縮成一個固定大小向量時，越長的文字就越容易損失訊息，也因此除非在較簡單的問題，否則現在通常還會加入注意力機制 (Vaswani et al., 2017) 來解決此問題。

綜上所述，我們最後決定使用一個基於 seq2seq 並帶有 attention 機制的神經機器翻譯系統，來幫助一些非客語領域專精的人也能做出的翻譯系統。

3 語料庫

本研究中會需要中文到客文的翻譯平行語料，並且由於後續 Fairseq-CNN 模型將會需要斷詞資訊，然而像漢語此類的語系不像英文語系本身就有空格來當作斷詞的效果，中文客文的斷詞因爲目前所使用的語料庫本身多半都沒有附人工處理好的斷詞資訊，所以斷詞這部份得另外處理，斷詞處理將會在下一章節 4.2說明。

在北四縣模型訓練中我們將會使用北四縣腔調的語料：北四縣哈客、萌典；在南四縣模型訓練中我們將會使用：南四縣哈客、美濃客家寶典、萌典 (南四縣)。

3.1 北四縣語料

3.1.1 北四縣哈客

北四縣客語語料來源其中一個是來自客委會的四縣腔初級、中高級客語認證教材 (客家委員會, 2021)。其中分爲初級 1284 個、中級 1767 個、中高級 2146 個，共 5197 個客語單詞，每個單詞除了對應的中文、客文拼音、使用該詞的範例句 (至少一句) 以及與該句子其相對應的中文翻譯。由於每個單詞可能不只一個客文例子，最後經整理將初、中、中高三個級別的每個單詞所有的客文句子整理合併後共有 5801 個句子。透過標點符號進行切分後共有 9488 個小句子。

另外如果該詞有南北四縣對相同意思的中文在使用單詞上有差異時候則會有括號附註南四縣的用法如表2，總共包含 834 個。

3.1.2 萌典

另一個北四縣語料是以來自教育部的《臺灣客家語常用詞辭典》爲原始資料所編制成的萌典 (唐鳳, 2013)，萌典共有 14713 個客語單詞，同樣擁有中文、客文拼音以及該詞的零到多個客文句及該句中文翻譯句。北四現在萌典中使用了全部的資料，最後經整理且經過標點符號切分後共有 21302 個小句子。

3.2 南四縣語料

3.2.1 南四縣哈客

南四縣語料的其中一個原始資料來源也是客委會的客語認證教材，並經由本校人員人工處理過，與北四縣哈客的不同之處在於南四縣哈客只使用了部份的資料，並且全部都換成南四縣的用詞，另外有些客語句子的中文翻譯有些許不同。最後總共有 4196 個句子。經過標點符號切分後共 7350 個小句子。

類號	級	類	號	客語標音	客家語
17-29	A	17	29	ted 【hed 】	忒【核】
18-13	A	18	13	dag bai 【mi bai 】	逐擺【每擺】
18-20	A	18	20	dong 【goi】	當【蓋】

Table 2: 強制斷詞差異

3.2.2 美濃客家寶典

南四縣哈客的另一個資料來源是由本校的劉明宗老師著作的《美濃客家語寶典》 (劉明宗, 2016)，同樣經由本校人員人工處理過，只有客文句子和中文翻譯，共有 3345 句，標點符號切分後共 8547 個小句子。

3.3 重疊語料處理

3.3.1 哈客網

由於北四縣哈客與南四線哈客的原始資料來源同樣取自客委會哈客網資料，所以部份句子在使用詞上無差異時候，這些資料會與北四縣哈客完全重疊，這樣的句子共有 4044 個句子。有些則因爲中文翻譯與北四縣不同但客文原句是相同的，這樣的例子我們將同時把兩邊的中文翻譯句子都用在南北四縣語料中，這樣的例子有 304 個句子。其餘資料則是客文句子本身就與北四縣不同，這也是主要哈客語料需要分開的原因，也是後續模型需要學習的差異處。除了句子相似但用詞不同之外的例子之外，北四縣還包含一些南四縣未用的句子，上述兩種情況加起來，在北四縣中有 5139 個句子，在南四縣中有 3002 個句子。

因南四縣哈客語料由母語爲客語之專業人員進行標音處理後的例句中並未包含全部客委會所提供的例子，所以自行整理剩下未用上且沒有用詞上差異的句子加入南四縣語料中，以幫助擴增南四縣語料，總共有 1670 句。

3.3.2 萌典

由於萌典是以上所有語料庫中資料最多的語料，然而萌典的例句皆以北四縣用詞優先，但若未幫南四縣擴增萌典的語料，會使南四縣有大量的未知詞，爲此必須整理萌典語料並且盡可能避開有包含南北四縣使用詞上有差異的句子。

利用前面哈客網中所整理好的 834 個南北差異用詞，用這 834 個詞中的北四縣用詞在萌典的 21302 句子中過濾，只要含有這些差異用詞的句子則全部剔除，剩下總共有 15160 個句子，這些句子的中文與客文翻譯與北四縣完全相同。

	哈客網	萌典	美濃寶典	總和
北四縣	9488	21302	0	30790
南四縣	9020	15160	8547	32727

Table 3: 南北語料庫句數總和

4 研究方法

本章節說明使用 Fairseq-CNN 架構來實做客家語的翻譯、中文客文斷詞對訓練結果的影響、使用不同語料訓練出南北四縣腔的翻譯差異、當在資料稀疏下如何處理未知詞的問題。

4.1 Fairseq-CNN

2017 年 Facebook 提出以卷積神經網路 (Convolutional Neural Network，CNN) 爲基礎來處理 seq2seq 的問題 (Gehring et al., 2017)，文中主要提出三點使用 CNN 來處理這類問題對比 RNN 的優勢。

1. CNN 可以進行並行運算，而 RNN 是鏈式處理，必須等前一幀的結果出來才能處理下一個，故 CNN 在訓練速度上會快非常多。

2. CNN 網路可透過卷積的疊加，讓較低層處理輸入序列中鄰近字的交互關係，而讓較高層處理較遠字的交互關係。與 RNN 的結構相比，CNN 可用更短的路徑得到遠處的資訊。

3. 對於輸入的一組輸入而言，在 CNN 網路中，所有單詞經過的卷積核 (kernel) 和非線性計算的數量都是固定的，但在 RNN 網路中，第一個單詞要經過 n 次單元和非線性計算，但是最後一個單詞只經過一次，CNN 中同一組輸入中的每個詞有相同的計算將有助於訓練。

Fairseq-CNN 的整體模型架構如圖1

4.1.1 Position Embeddings

由於使用了 CNN 結構相比 RNN 來說少了位置訊息，所以必須加入個能表示輸入序列中的某詞在該序列位置的資訊 Position Embeddings。公式如下：

$$e = (w_1 + p_1, +...+, w_m + p_m) \quad (1)$$

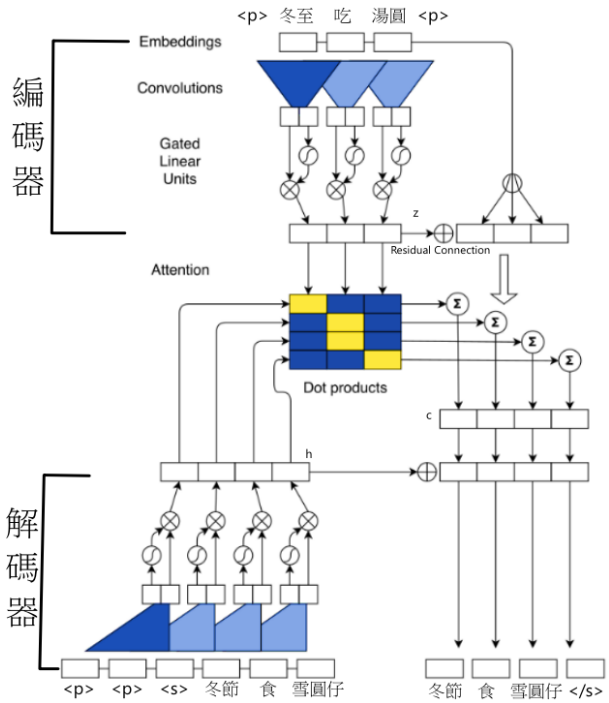

Figure 1: Fairseq-CNN 模型架構。正上爲 encoder 部分，左下爲 decoder 部分，中間爲 attention 部分。

w 表示詞向量 (word embedding)，p 表示位置向量 (position embeddings)，兩者相加可得到輸入元素 e，將作爲下一個卷積結構的輸入。

4.1.2 Convolutional Block Structure

編碼器與解碼器使用同一種卷積結構 (Convolutional Block Structure)，這種結構包含一個一維卷積和一個非線性單元 (Gated Linear Units，GLU) (Dauphin et al., 2017)。

輸入元素 e 經過參數爲 $W \in \mathbb{R}^{2d \times kd}$ 的一維卷積，其中 k 是卷積核寬度，代表一次段幾個詞做卷積處理，d 是詞向量的長度，最後被輸出爲兩倍維度的 $Y \in \mathbb{R}^{2d}$。

$Y = [A\ B] \in \mathbb{R}^{2d}$ 繼續被輸入到 GLU 中，GLU 公式如下：

$$v([A\ B]) = A \otimes \sigma B \tag{2}$$

其中 $A, B \in \mathbb{R}^d$，$\sigma(B)$ 負責控制與輸入上下文中哪些與 A 相關，A 與 $\sigma(B)$ 互相做點乘後得到輸出 $v([A\ B]) \in \mathbb{R}^d$。最後在加上殘差連接 (residual connections) (He et al., 2016)。

4.1.3 Multi-step Attention

每個解碼器層都有個單獨的注意力機制，爲了計算注意力權重 a，當前的解碼器狀態 h_j^l 與前一個目標元素 g_i 的 embedding 做結合，其公式如下式 3，W 爲權重，b 爲偏差 (bias)。

對於解碼器層 l 的注意力 a_{ij}^l (對第 i 時刻第 j 個來源元素的注意力權重)，解碼器狀態總和 d_i^l 與編碼器的最後輸出 z_j^u 做內積 (Dot Product)，其公式如下式 4。

最後利用注意力權重 a 對編碼器輸出 z 加上輸入向量 e 做加權，最後得到 c 最爲下一層卷積層的輸入，其公式如下式 5。

$$d_i^l = W_d^l h_i^l + b_d^l + g_i \tag{3}$$

$$a_{ij}^l = \frac{\exp(d_i^l \cdot z_j^u)}{\sum_{t=1}^m \exp(d_i^l \cdot z_t^u)} \tag{4}$$

$$c_i^l = \sum_{j=1}^m a_{ij}^l (z_j^u + e_j) \tag{5}$$

4.2 斷詞

由於 Fairseq 在訓練與翻譯之前都需要先將句子斷詞，而我們原始語料的句子都是沒有斷詞的，所以必須在訓練之前需先將中文和客文都先做斷詞。中文斷詞部份我們使用中研院的 CkipTagger (Peng-Hsuan Li, 2019) 系統，客文斷詞部份我們使用網路上的基於結巴的客文斷詞系統 (ldkrsi, 2018)，該客語斷詞系統的訓練資料來自苗栗、東勢、新屋、楊梅、龍潭、花蓮客語故事集、客家笑科、徐老師講古。

由於客文某些專用詞在資料不足的情況下不容易直接翻出，所以某些中文詞在斷詞時候直接將斷詞強制斷成客文詞對應的句子會比較容易成功翻譯出想要的特定客語用詞。例如：客家話的【食夜】，中文意思是【吃晚飯】，食夜在客語中是一個詞，但在中文吃晚飯會被斷詞層一個動詞 + 名詞的【吃晚飯】。在資料不足的情況下，若輸入【吃晚飯】很容易被翻譯成【食晚飯】而不是【食夜】，雖說該翻譯在語意表達上並沒有錯誤，但缺乏了客語與中文之間的用詞差異性。故我們提出在訓練模型之前就將中文的斷詞系統加入強制斷詞，讓專有的客語詞彙有著對應的中文斷詞結果，例如上述的【吃晚飯】將被斷成一個新的單詞。爲此我們從萌典整理出約 10600 個華客對應的詞典加入中客斷詞系統中。

4.3 未知詞處理

機器翻譯在翻譯時候一定會有未知詞的問題，一般狀況下在翻譯輸出每個詞時會選擇機率最高的翻譯詞作爲結果，但是模型預測出來的詞，可能因訓練語料中不存在合適的詞作爲翻譯的結果，導致其機率偏低。當這種狀況發生時，可以藉由設定一個閾值來調整是否相信模型預測的結果，在此使用未知詞懲罰值

(unknown word penalty，縮寫成 unkpen) 來達成。由於很多單詞的預測機率數字過低時，若以原本數字呈現不佳，故通常在呈現機率時候以對數的方式呈現，而機率是個介於 0 與 1 之間的數字，這區間的數字在 log 函數下都是負數，且原本機率越接近 0 時候，其 log 值會趨近於負無限大，爲了最佳化模型輸出的預測結果，我們可以經由調整 unkpen 的大小來調整。實際上的做法是透過將 unkpen 的 log 機率減掉一個介於-12 到 0 之間的值，調整其機率大小。若調整後 unkpen 的機率大於原始模型所預測的詞機率，則以 <unk> 取代原始預測的詞。

由於客語所收集到的語料較少，將會導致模型判斷不佳的可能性偏高。不過由於中文和客文在句型上跟用詞有時候是互通的，很多時候直接沿用中文就可以達到不錯的理解度。所以在客文翻譯結果中，當前述的未知詞取代的狀況發生時，有可能是中文跟客文的詞相似，所以沒有列入客語詞典中，或者剛好訓練資料未出現過。此時我們只要透過尋找模型中輸出的翻譯詞的來源注意力值往回尋找該詞的來源輸入是誰，就可以將該未知輸出替換成原本的輸入詞到結果中。

而造成輸出結果未知的可能大致可分爲兩種。一是訓練資料本身就未見過此新詞，在這種情況下大部分所有候選單詞輸出機率幾乎都會非常低，這時候 <unk> 本身的機率就不會比其他候選詞低多少，配上懲罰值就能輕易讓 <unk> 成爲最高機率候選詞並取代其他所有不理想的結果，如表4上半部，然後在利用前面所述的經過後處裡將未知詞的從來源中文直接當成答案，在多數情況下就會是個可接受的結果。

二是資料不足或者前後文判斷條件不夠多，導致推測答案的時候的不確定性因素過多，有多個相近機率的候選詞可選擇，此時可能有多個輸出的機率相近且明顯比多數不好的候選詞機率高非常多，如表4下半部中的候選詞 1 到 3 的機率特別高，但不代表最高的一定是正確的，有可能次高的才是相對較貼切的翻譯。爲此到底要選擇原本最高的機率當預測答案，或者挑個合適的懲罰值來讓答案變未知，並在用前述方法直接套用中文結果，便需要實驗來做測試。

5 實驗結果

本章節主要說明訓練模型的超參數設定、測試語料的來源與數量、用何種方法評估實驗結果的好壞、以及最後結果討論。

5.1 實驗設置

我們的 Fairseq 模型使用 CNN 架構，在原始論文中使用所有編碼器和解碼器使用 512 個隱藏單元，embedding 層和線性層皆使用 512 維，由於我們的語料庫大小以及所使用的詞彙量相較於原始論文的大數據來說都明顯少很多，所以我們將所有編碼器和解碼器中改使用 256 個隱藏單元，embedding 層和最後線性層的維度也改爲 256 維，kernel-size=3，dropout=0.2，學習率調整方式使用 inverse sqrt，損失函數使用 label smoothed cross entropy，優化器使用 adam。

南北四縣腔訓練語料將表3中各自的所有語料全用上，包含北四縣 30790 句、南四縣 32727 句，這些資料在各自以訓練集 90%、驗證集 5%、測試集 5% 做切分。

5.2 測試語料

測試語料是來自哈客網的“客語口說故事”中的其中十篇童話故事，其中包含，裡面含有相同中文並翻譯成南四縣和北四縣的人工翻譯結果，經過標點符號進行切分後共 1221 個小句子，這些測試語句並沒有包含在訓練語料中。

5.3 評估方法

我們使用萊文斯坦距離 (Levenshtein Distance) 來測試翻譯的結果與給定的答案的最短編輯距離，在萊文斯坦距離中，可以新增、刪除、取代字串中的任何一個字元，最後把所有 10 篇童話故事的所有句子的全部編輯距離做相加，數字越小代表越好。

5.4 實驗結果

我們的 baseline 系統是原句子做斷詞後的每個中文詞直接拿去查華客對應詞典，如果該中文詞在詞典中有對應的客文詞，則直接用對應的客文詞取代原有的中文詞，如果沒有對應的客文詞則沿用中文詞。此 baseline 系統在測試語料中的編輯距離爲 5078。

我們測試北四縣的測試語料結果如表5上半部所示，N 代表北四縣模型、N2 代表北四縣模型加入強制斷詞、S 代表南四縣模型、S2 代表南四縣模型加入強制斷詞，其中可見強制斷詞後的效果較佳，且效果比 redbaseline 系統和南四縣模型更好，並且未知詞懲罰值設定爲-8 時，可以得到最好的結果。

南四縣測試集結果如表5下半部所示，也可見強制斷詞後的效果較佳，且效果比 redbaseline 系統和北四縣模型更好，並且懲罰值一樣設定在-8 時可得到最好的結果。

南北四縣腔翻譯差異例子可參考表6。可見確實有將【湯圓】翻譯成對應腔調的文字。

	候選詞 1	候選詞 2	候選詞 3	候選詞 4	…	…	<unk>
原始機率	1.10E-15	8.54E-14	2.32E-12	9.15E-13	…	…	7.65E-13
log 機率	-14.9586	-13.0685	-11.6354	-12.0384	…	…	-12.1161

	候選詞 1	候選詞 2	候選詞 3	候選詞 4	…	…	<unk>
原始機率	0.45	0.33	0.23	9.15E-16	…	…	7.65E-09
log 機率	-0.3468	-0.4815	-0.6364	-15.0384	…	…	-8.1161

Table 4: 未知詞懲罰值範例。上半部爲所有候選詞機率皆非常低，下半部爲少數幾個特別高

北四縣資料	-10	-9	-8	-7	-6	-5	-4	-3	-2
N	6318	5775	5417	5282	5267	5278	5300	5307	5307
N2	4295	3954	3908	4060	4161	4250	4303	4311	4311
S	4321	4159	4151	4309	4482	4572	4645	4648	4651
S2	4317	4090	4050	4163	4291	4349	4362	4368	4368
南四縣資料	-10	-9	-8	-7	-6	-5	-4	-3	-2
S	4999	4778	4662	4802	4912	5006	5075	5079	5082
S2	4970	4679	4593	4691	4784	4839	4853	4858	4858
N	7012	6497	6143	5987	5954	5960	5983	5992	5992
N2	5261	4926	4827	4953	5044	5105	5143	5150	5150

Table 5: 童話故事測試集-南北四縣測試集的萊文斯坦距離總和

強制斷詞差異例子可參考表7，因爲【肚子餓】在原始斷詞下被斷成【肚子】跟【餓】，而原本【肚子餓】可翻譯成【肚飢】，若照原本斷詞則可能翻譯成肚屎 (肚子的客文) 跟枵 (餓的客文)。

另外我們發現使用原始斷詞方法 (N 模型) 的最佳懲罰值爲-6，而強制斷詞方法 (N2 模型) 的最佳懲罰值爲-8，懲罰值越接近 0 代表模型越相信它原本的判斷是正確的。

我們猜測是強制斷詞導致中文斷詞後的文法結構不穩定，例如前述的【肚子餓】，主詞會被強制連著動詞，但其餘沒有列入強制斷詞詞典中的主詞與動詞不會連在一起，這導致模型缺乏了一般性。在資料稀疏的狀況下此種強制斷詞方法雖然能夠幫助翻譯結果更容易出先客語專有詞，但當有巨量資料情況下則不需要藉此方法，模型可自己從資料中學習到之間的關係。而又因爲訓練資料有一半是來自萌典，而萌典原始資料是使用北四縣用詞，雖然有經過篩選句子，但句子本身應該依然更貼近北四縣腔，這使得訓練資料中的中客翻譯在北四縣腔上更加匹配，故北四縣模型 (N 模型懲罰值-6) 比起南四縣模型 (S 模型懲罰值-8) 更容易相信模型自身的判斷是正確的。

6 結論

我們使用神經網路機器翻譯 Fairseq-CNN 來處理中客翻譯問題，並使用不同語料針對不同腔調做訓練，可做出翻譯對應腔調的客文翻譯。並提出使用強制斷詞方法來提高翻譯出特定客語詞彙的機率，並使用最短編輯距離的方法測出一個最佳的未知詞懲罰值來當未知詞的一種閥值設定，並利用中文與客文的高相似性關係，直接使用來源中文詞取代客文的未知詞來當翻譯結果。

由於訓練資料的不足，導致翻譯系統有時會出現不佳的翻譯結果，如果有更多的訓練資料則可翻譯出更好的結果。當有良好的翻譯結果後，後續可將客語句子經過語音合成系統合出該句子的語音檔，最終完成一個由輸入中文文字轉成客語音檔的目標，將有利於客語學習。

中文	今晚吃湯圓	你又要買很貴的東西了
北四縣翻譯	暗晡夜食雪圓仔	你又愛買已貴个東西了
南四縣翻譯	暗晡夜食圓粄仔	你又愛買恁貴个東西吔

Table 6: 強制斷詞差異

S2	中文	客文
未斷詞	肚子餓的時候就不要挑食	
原始斷詞	肚子 餓 的 時候 就 不要 挑食	肚屎 枵 个 時節 就 莫 揀食
強制斷詞	肚子餓 的 時候 就 不要 挑食	肚飢 个 時節 就 毋好 揀食

Table 7: 南北四縣翻譯差異

References

Media A Ayu and Teddy Mantoro. 2011. An example-based machine translation approach for bahasa indonesia to english: An experiment using moses. In *2011 IEEE Symposium on Industrial Electronics and Applications*, pages 570–573. IEEE.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Paisarn Charoenpornsawat, Virach Sornlertlamvanich, and Thatsanee Charoenporn. 2002. Improving translation quality of rule-based machine translation. In *COLING-02: machine translation in Asia*.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Jan K Chorowski, Dzmitry Bahdanau, Dmitriy Serdyuk, Kyunghyun Cho, and Yoshua Bengio. 2015. Attention-based models for speech recognition. *Advances in neural information processing systems*, 28.

Kit Chunyu, Pan Haihua, and Jonathan J Webster. 2002. Example-based machine translation: A new paradigm. *Translation and information technology*, 57.

Yann N Dauphin, Angela Fan, Michael Auli, and David Grangier. 2017. Language modeling with gated convolutional networks. In *International conference on machine learning*, pages 933–941. PMLR.

Mikel L Forcada, Mireia Ginestí-Rosell, Jacob Nordfalk, Jim O'Regan, Sergio Ortiz-Rojas, Juan Antonio Pérez-Ortiz, Felipe Sánchez-Martínez, Gema Ramírez-Sánchez, and Francis M Tyers. 2011. Apertium: a free/open-source platform for rule-based machine translation. *Machine translation*, 25(2):127–144.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. 2017. Convolutional sequence to sequence learning. In *International conference on machine learning*, pages 1243–1252. PMLR.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Philipp Koehn. 2009. *Statistical machine translation*. Cambridge University Press.

ldkrsi. 2018. jieba-hakka. https://github.com/ldkrsi/jieba-Hakka.

Chuan-Jie Lin and Hsin-Hsi Chen. 1999. A mandarin to taiwanese min nan machine translation system with speech synthesis of taiwanese min nan. In *International Journal of Computational Linguistics & Chinese Language Processing, Volume 4, Number 1, February 1999*, pages 59–84.

Hsin-Wei Lin, Feng-Long Huang, Ming-Shing Yu, and Yih-Jeng Lin. 2014. 中文轉客文文轉音系統中的客語斷詞處理之研究 (research on hakka word segmentation processes in chinese-to-hakka text-to-speech system)[in chinese]. In *Proceedings of the 26th Conference on Computational Linguistics and Speech Processing (ROCLING 2014)*, pages 58–77.

Ramesh Nallapati, Bowen Zhou, Caglar Gulcehre, Bing Xiang, et al. 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond. *arXiv preprint arXiv:1602.06023*.

Wei-Yun Ma Peng-Hsuan Li. 2019. Ckiptagger. https://github.com/ckiplab/ckiptagger.

Alexander M Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. *arXiv preprint arXiv:1509.00685*.

Harold Somers. 1999. Example-based machine translation. *Machine translation*, 14(2):113–157.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, 27.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez,

Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

WillPete. 2022. 北四縣和南四縣有什麼不同？臺灣客家語四縣話難北部腔調的差異分析與比較整理. https://www.wpchen.net/zh/posts/hakka-taiwan-sixian-northern-southern-difference.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Richard Zens, Franz Josef Och, and Hermann Ney. 2002. Phrase-based statistical machine translation. In *Annual Conference on Artificial Intelligence*, pages 18–32. Springer.

邱國源劉明宗. 2016. 美濃客家語寶典, volume 1. 五南.

唐鳳. 2013. 萌典. https://www.moedict.tw/about.html.

客家委員會. 2021. 哈客網路學院-中級暨中高級教材及試題下載. https://elearning.hakka.gov.tw/hakka/download-files?c=3.

臺灣教育部. 2018. 【客語】臺灣客家語有哪幾種常見的腔調？. https://mhi.moe.edu.tw/faqList.jsp?ID=0&ID2=11.