Identifying Negative Language Transfer in the English Writing of Chinese and Farsi Native Speakers

by

Mohammad Karimiabdolmaleki

A thesis submitted in partial fulfillment of the requirements for the degree of

Master of Science

Department of Computing Science

University of Alberta

Abstract

Effective communication in English can facilitate educational and employment opportunities for second-language learners. English as a second or foreign language (ESOL) learners tend to employ rules from their native language while communicating in English, which can lead to Negative Language Transfer (NLT) when the rules transferred from the mother tongue do not match those of English. To assist ESOL learners in writing in English, NLT errors should be identified. However, manually identifying NLT is a difficult task, demanding time and expertise in both languages. Although NLT is a well-researched phenomenon in linguistics, few attempts have been made to automatically identify NLT in learner writing using machine learning techniques.

In this work, I have implemented four classification algorithms to automatically identify NLT errors in second-language learner writing. The results show that the models can identify NLT in the English writing of Chinese and Farsi native speakers. This work makes the following contributions: (1) it implements supervised machine learning models and language models to identify NLT in learner writing; (2) it evaluates the models using two different datasets in two languages to investigate the generalizability of the models; and (3) it identifies the most important features for detecting NLT. This work shows that the implemented models can be used in unstructured domains to identify NLT automatically for speakers of two languages: one is logographic and the other alphabetic.

"Never will die the ones whose heart has been revived by love." (Hafez)

Dedicated to the memory of my mother.

March 22, 2022

Acknowledgements

Words cannot express my gratitude to my supervisors, Maria and Carrie, for their invaluable patience and feedback. As well as guiding me toward completing my thesis-based master's of computing science, they taught me several essential life lessons for which I will be forever grateful. I also could not have undertaken this journey without my defense committee, who took the time to read the thesis and provide me with feedback. Also, I would like to thank my classmates and lab members, especially Leticia, who was kind enough to guide me with the analyses. Lastly, I would be remiss in not mentioning my family, especially my parents, brothers, and sister. Their belief in me has kept my spirits and motivation high during this process.

Contents

List of Tables

List of Figures

Glossary

FCE

First Certificate in English, an English language proficiency exam offered by Cambridge Assessment English

GED

Grammatical Error Detection, a natural language processing task in which the goal is to detect grammatical errors

L1

Mother tongue or somebody's native language

L2

A language different from the mother tongue

Lang-8

A social networking website for language exchange with more than 750,000 users from 190 countries

NLT

A language-learning phenomenon that occurs when the reused grammar and structures from the mother tongue do not fit the second language grammar and structure

POS

Part of Speech (POS) is a token representing the grammatical category of a word

UD

UD (Universal Dependencies) is a shared language annotation framework across several languages

Chapter 1

Introduction

Many areas have been positively affected by algorithms that can learn from experience and adjust to new inputs, from search engines and spam detection to language translation and feedback-enabled writing tools. Natural Language Processing (NLP) is a field that is concerned with how computer systems can be used to understand and process natural language text or speech (Chowdhary, 2020). With the development of ML and NLP, several language-learning tools (e.g., Duolingo, Grammarly) have been created. These tools help users to learn a second-language and communicate (Jiang et al., 2020). These language-learning systems and others like them may provide corrective feedback to second-language learners (Monaikul and Di Eugenio, 2020; Nadejde and Tetreault, 2019). Corrective feedback can indicate the erroneous usage of a target language (Bacquet, 2019). It can assist the learner in correcting the error and may help in preventing the repetitive occurrence of the same error type (Tsui, 2007). The continual research and development of language-learning systems implies that researchers are trying to address challenges that still exist in learning a new language, especially, the English language which has billions of native and non-native speakers.

English is the world's most commonly spoken language and it has more non-native speakers than native speakers. Effective communication skills in English can support the development of a person's career. Precise communication can prevent the unintended consequences that can emerge from miscommunication.

Learning a new language is often enabled by using strategies such as repe-

tition, memorization, and translation (O'Malley and Chamot, 1990). Employing memorization and translation strategies while learning a second language can lead to the occurrence of language transfer, which occurs when second-language learners employ patterns from their first language (L1) while communicating in the second language (L2) (Lado, 1957). Language transfer can occur through conscious or unconscious processes. Language transfer is considered conscious when language learners intentionally use the grammatical structures of their L1 while generating speech or text in the second language. Language transfer is unconscious when language learners unintentionally use the grammatical structures of their L1 in the second language. Unconscious language transfer happens when language learners are not paying attention to the usage of the grammatical structures of the second language. It can also happen when they have not mastered the grammatical structures of the second language or they have forgotten the appropriate usage of the language.

There are two types of language transfer: positive language transfer and NLT. Reusing the relevant unit or structure of the first language (L1) when the corresponding structure in the second language (L2) is the same can result in correct language production, which is called positive language transfer. NLT occurs when the reused grammar and structure from the L1 do not fit the L2 grammar and structure. A lack of knowledge of the grammatical variations across languages and similarities between grammatical structures are the main causes of NLT. In this thesis, the focus is to identify NLT from structural errors made by second-language learners. As an example of NLT, consider this sentence written by a Chinese native speaker:

In my opinion, I recommend British museum.

In the above sentence, the British Museum should be preceded by a determiner (i.e., the). As the language learner did not use a determiner before the noun phrase, the sentence contains a missing determiner error. The existence of this error could be due to the fact that articles do not exist in the Chinese language (Robertson, 2000). This example demonstrates that the Chinese native speaker transferred the grammatical rules from their L1 to their L2. Diver-

gence of the determiner rules in Chinese and English caused the transferred pattern to form an error.

As another example, consider the sentence below written by a Farsi native speaker:

Especially for the people who have good sense of humour.

The Farsi native speaker committed an error while writing this sentence. Akin to the previous example, the language learner did not include the determiner (i.e., a) when writing this sentence.

NLT occurs frequently in English due to the large number of non-native speakers of English. Despite the existence of various ML-based language-learning tools for English, none of them detects NLT errors in language learner writing. This implies the need for an automated tool that can detect NLT errors. As the manual identification of NLT requires time and expertise in both the L1 and L2, automatic identification of NLT can facilitate its detection. The automatic identification could be integrated as a module in language-learning tools to make language learners aware of the occurrence of NLT.

In this thesis, I applied four machine learning models to detect NLT in two datasets from speakers of different languages: Chinese and Farsi. Two sets of binary classification models were used to detect the erroneous utterances of second-language learners: theory-based models and non-theory-based models. Theory-based models represent the syntactical structure of the input language using language models and part-of-speech (POS) information. The theory-based models were built using n-grams and a recurrent neural network (RNN). The non-theory-based models used logistic regression and random forest. As a result, eight models were employed to identify NLT in the English writing of Chinese and Farsi native speakers.

To determine to what extent and how accurately these models could identify NLT, two different datasets were used to evaluate the models: the Chinese FCE dataset and the Farsi Lang-8 dataset. The choice of employing two distinct datasets from native Chinese and Farsi speakers was made to explore the potential generalizability of the proposed methods. Using the two datasets

also enables comparing and contrasting model performance across languages.

The research questions posed in this work are as follows:

- **RQ1:** What is the performance of the proposed models in detecting NLT?
- **RQ2:** What is the performance of the theory-based approaches (n-gram and RNN) compared to the non-theory-based approaches (logistic regression and random forest)?
- **RQ3:** What features are important for detecting NLT across approaches and languages?

The rest of this thesis is organized as follows. **Chapter 2** (Literature Review) examines the relevant research on detecting grammatical errors and NLT. **Chapter 3** (Methodology) describes the datasets, input features, and output (response or target variable) of the machine learning algorithms, as well as the methods applied to the data and the model evaluation procedures. **Chapter 4** (Results) presents the model findings and it also includes an error analysis. **Chapter 5** (Discussion) provides an interpretation of the results, implications, limitations, and future work. **Chapter 6** (Conclusion) provides a summary of the thesis and the conclusions that follow from the employed methodology.

Chapter 2

Literature Review

Over the past four decades, several different theories of second-language acquisition have been proposed to explain how language learning takes place, which variables play a role in second-language acquisition, and how to provide guidance to second-language learners. The monitor model, which includes several influential theories of second-language acquisition, developed by Stephen Krashen, includes five central hypotheses (Gitsaki, 1998): (1) learning is a conscious process, while acquisition is a subconscious process; (2) the role of learning is to monitor and adjust the utterances generated throughout the acquisition process; (3) there is a natural order in understanding second-language rules which could be influenced by classroom instruction; (4) being exposed to comprehensible input is the only way that can lead the language learner to second language acquisition; and (5) the existence of a conceptual block can hinder learners from utilizing comprehensible inputs.

Based on Krashen's hypotheses, grammatical error detection (GED) and identifying NLT from structural errors can make students aware of their grammatical mistakes and guide them toward the correct usage of the grammar rules.

2.1 Grammatical error detection

A grammar represents the structure, rules, syntax, and morphology of a language (Leacock et al., 2010). Grammatical errors are often categorized as errors related to faulty or unconventional usage of a grammar. Grammatical

errors can sometimes include a subset of spelling errors (Fraser and Hodson, 1978). The earliest grammar checking tools were based on string matching rather than on syntax and grammar. Later, tools employed some linguistic analysis. For example, IBM's *Epistle* (Heidorn et al., 1982) and *Critique* (Richardson and Braden-Harder, 1988) enabled full linguistic analysis using complex grammars and parsers. The development of grammar checking programs using hand-coded grammar rules gave way to statistical methods in the 1990s. At this time, data-driven models began to be trained and developed using large-annotated treebanks such as the Penn Treebank (Marcus et al., 1993).

With the development of NLP, numerous studies have been conducted to address grammatical error detection. Lee et al. (2014) developed a rule-based and a statistical-based approach to detect grammatical errors in Chinese sentences. Their rule-based method contained 142 rules written by language experts and the statistical model was based on n-gram scores of correct and incorrect training sentences. Their experiment used 880 sentences with grammatical errors written by Chinese students. The 142 developed rules, bigram, and trigram language models were used to identify the grammatical errors. The experiment was conducted using rule-based models, n-grams, and their combination. Although the rule-based approach yielded a high precision score of 86%, n-grams achieved higher recall. The highest F1-score was obtained using the trigram model with an F1-score of 69%. Additionally, the combination of a rule-based model with n-grams resulted in the lowest false positive rate. Given these results, the choice of the best model may depend on application requirements or preferences.

With the development of word vector representations and deep learning, new algorithms have emerged that can detect grammatical errors with higher precision. Bell et al. (2019) presented an approach to effectively integrate contextualized word embeddings to detect grammatical errors. The study used a bi-LSTM sequence labeler that was applied over the sequence of tokens. For each token, the model was trained to predict whether the token was correct or incorrect (i.e., binary classification). The model was additionally trained

with a bidirectional language model to predict the surrounding context of the target word in the input sequence. Bell et al. (2019) further employed Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al., 2019), embeddings from Language Models (ELMo) (Peters et al., 2017), and Flair word embeddings (Akbik et al., 2018) to discover the importance of context in identifying errors. They conducted the prediction task on three different datasets and found that BERT word embeddings provided the highest improvement across all datasets.

Despite the development of grammatical error detection algorithms, there has been limited research on developing algorithms that can detect NLT because NLT comprises a wide range of errors that second-language learners make when speaking or writing in their second language.

2.2 NLT error detection

Detecting NLT in written essays or text could be an instrumental component of language learning and assessment tools. While NLT is well-researched and has been analyzed for decades (Selinker, 1969; Murphy, 2003), there have been only a few attempts to identify NLT in texts written by English language learners (Farias Wanderley and Demmans Epp, 2021; Farias Wanderley, Zhao, et al., 2021).

The negative influence of language transfer on learning a second language can be examined in two ways: theoretical and practical. Cortés (2005) conducted a theoretical and practical analysis to understand the negative impact of language transfer on British students learning Spanish as a second language. The study found that factors such as type of the L1 and L2, the relationship between the learner's known languages and those being studied, and the learning context were involved in language transfer.

Kastell (2021) collected a dataset of university lecture materials in English delivered by Finnish and Swedish native speakers to find out the rate of NLT occurrence and the impact of the native language on NLT. The dataset is composed of three different error types related to articles in English: omission

errors, where the speaker omitted an article (a, an, the) due to the rules in their native language; addition errors, where the speaker adds an unnecessary article to the sentence; and substitution errors, where the speaker incorrectly substitutes articles. The study analyzed the data and discovered that NLT occurrence is more frequent in Finnish than in Swedish. They argued this was because Finish does not use articles.

Chodorow et al. (2007) developed a maximum entropy classifier using machine learning and rule-based filters to detect preposition errors in English as a second-language learner essays with a precision of 0.8 and a recall of 0.3. In addition to preposition errors, verb errors are another common error type made by non-native speakers of the English language (Rozovskaya et al., 2014). To train and evaluate a linguistically-motivated approach, the study used second-language learner essays from the First Certificate in English (FCE) dataset. The Cambridge Learner Corpus First Certificate in English contains texts written by English as an additional language learners in response to exam prompts (Yannakoudakis et al., 2011). Rozovskaya et al. (2014) employed the notion of verb finiteness to improve the accuracy of a statistical machine learning model. The pipeline starts with selecting the verb candidates and determining verb finiteness. Subsequently, features are generated for each candidate and the finiteness prediction results are used in the error identification component. Based on the output of the error identification module, the corresponding classifier for each error type is used to propose a proper correction. They used a linear model called *combined* that assigned a score to each label of the label space using the input verb and the weight vector w. The study discovered that employing linguistically-derived knowledge in a machine learning model can boost performance and enable a general correction approach to verb errors.

Wu et al. (2009) proposed language models to automatically detect and correct NLT made by Chinese native speakers. The study implemented relative position language model and parse template language model to tackle the error correction problem. The relative position language model was presented to address the order of the error correction module and to preserve the relative position and long-range lexical information between constituents of the sen-

tence. The parse template language model was presented to introduce more structural information to support the detection and correction modules. The process of error detection and correction was composed of two modules. First, models were trained to detect lexical, redundancy, omission, and word order errors. These models were used to detect if the input sentence contained an error. The error correction module then used the language models to compose the correct sentence. This approach outperformed an existing machine translation system trained on the same dataset.

In the most recent attempt of automatically identifying NLT errors, a machine learning model was trained with a dataset of learner errors and parallel corpora to represent the structure of the L1 and L2 languages. It used parallel corpora to distinguish the source language of a text sequence. Farias Wanderley and Demmans Epp (2021) trained language models (i.e., n-grams and a recurrent neural network) with parallel corpora to represent language structures and recognize when Chinese native speakers incorrectly transfer rules from their mother tongue (i.e., Chinese) into their L2 (i.e., English) writing. The n-grams and RNN achieved an F1-score of 0.45 and 0.51 on the negative language detection task, respectively. The methodology used in the paper is replicated in this thesis by applying it to two languages and their associated parallel corpora (i.e., Chinese and Farsi).

Farias Wanderley, Zhao, et al. (2021) introduced an annotated dataset of errors made by Chinese native speakers who had written essays in English. These errors are accompanied by information about the sources of the error. Using the error information, logistic regression and random forest models were trained to demonstrate potential for identifying NLT. The logistic regression and random forest models yielded an accuracy of 0.72 and 0.78 on the NLT detection task, respectively. These results imply the possibility of developing a feedback generation module for Chinese learners of English. The logistic regression and random forest models from this paper were employed in this thesis.

While being a well-researched topic in linguistics, NLT detection requires more attention from computer science researchers to address such errors in

second-language learner writing. This thesis aims to detect NLT in two different languages to demonstrate the robustness and generalizability of the methodology. The next chapter (i.e., methodology) introduces the datasets, machine learning models, and procedures applied to identify NLT.

Chapter 3

Methodology

In this chapter, I will describe the Chinese FCE dataset, the Farsi Lang-8 dataset, and the parallel corpora used to train the models. Then, I will explain the four algorithms I used to identify NLT in language-learner writing. I discuss NLT identification, a binary classification task for which I use two datasets and two categories of models: theory-based and non-theory-based. The input features of the non-theory-based models are extracted from preprocessing each learner's erroneous utterance and the response variable is the NLT status of that error. Non-theory-based models were trained, tuned, and evaluated using the Chinese FCE and Farsi Lang-8 datasets. Theory-based models were trained and tuned using parallel corpora and were evaluated using the Chinese FCE and Farsi Lang-8 datasets. The models were trained, tuned, and tested using cross-validation. Finally, I will discuss the evaluation criteria used in this work.

3.1 Data

The automatic identification of NLT errors in a sentence requires a dataset that labels errors as "Negative Transfer". Additionally, a parallel corpus containing sentences of each language is required to enable the modeling of language structure. This chapter starts with an overview of the two NLT datasets: (1) Chinese native speakers' errors while writing essays in English and (2) Farsi native speakers' errors while writing in English. Both datasets were used for training and testing the non-theory-based models (i.e., logistic regression and

random forest) and in the evaluation process for the theory-based models (i.e., n-gram and RNN). Next, I describe the parallel corpora used in the training process of the theory-based models (language modeling task). Parallel corpora were used to model the language structure using POS tag sequences. Finally, I describe the preprocessing techniques that were applied to the data while preserving the structural information of the text.

3.1.1 The Chinese FCE Dataset

The Chinese FCE dataset contains 66 English essays written by 66 distinct native speakers of Chinese. It has a total of 3,584 erroneous sentences. Each erroneous sentence in the dataset is a row in a spreadsheet and contains the learner's erroneous English writing and thirteen other columns from the FCE dataset.

This dataset was originally extracted from the Cambridge Learner Corpus (Yannakoudakis et al., 2011) and later annotated (Farias Wanderley, Zhao, et al., 2021). Each sentence of the dataset is associated with an error type and is either labeled as NLT or not. Out of the 3,584 errors, 53% are associated with NLT, 39% are not NLT errors, and the remaining 8% contain spelling errors. Information about the dataset is provided in Table 3.1.

Table 3.1: Error distribution in the Chinese FCE dataset

Dataset	**NLT Errors**	**Non-NLT Errors**	**Spelling Errors**	**No Error Annotation**	**Total**
Original	1,891 (52.7%)	1,389 (38.7%)	292 (8.1%)	12 (0.3%)	3,584
Error Subset	1,891 (57.6%)	1,389 (42.3%)	0 (0%)	0 (0%)	3,280
Structural Errors	1,478 (62.4%)	887 (37.5%)	0 (0%)	0 (0%)	2,365

As shown in Table 3.1, I split the Chinese FCE dataset into two chunks: error subset and structural errors. The error subset contains 3,280 records of which 1,891 are associated with NLT and 1,389 with a non-NLT error type. The original dataset has 292 samples that contain spelling errors and 12 samples with no error annotation. The error subset is the dataset that only con-

tains NLT and non-NLT errors. Instances containing a spelling error were excluded as they are not a type of structural error. Additionally, the samples that were annotated with "no error" were excluded as they did not contain error information. The structural errors subset was used because I aim to detect errors in sentence structure. Consequently, the error subset was filtered to contain instances containing structural error as their error type. The structural error subset is not balanced in terms of class variable frequency. However, the 25% difference between the majority and minority classes may not require the application of an oversampling technique (Abd Elrahman and Abraham, 2013).

Table 3.2 includes a sample taken from the Chinese FCE dataset. The sample contains a "Missing Preposition" (MT) error which is a type of NLT. The raw sentence demonstrates the three error types that are associated with the corresponding sentence. All three errors are provided in the dataset as separate instances with different error types and corrections.

Table 3.2: An example from the Chinese FCE dataset

Column	**Data Type**	**Example**
Student ID	Text	TE2*0100*2001*01
Language	Constant	Chinese
Overall Score	Discrete	27
Exam Score	Continuous	3.3
Raw Sentence	Text	I am writing to reply <NS type =”MT”><c>to</c></NS><NS type=”RD”>your<c>the</c></NS>letter you wrote <NS type=”MT”><c>to</c></NS>me on 10 June.
Error Type	Nominal	MT
NLT	Nominal	Y
Likely Reason for Mistake	Text	Chinese doesn’t use the word in this context
Error Length	Discrete	1
Correction Length	Discrete	1
Correct Error Index	Discrete	5
Correct Sentence	Text	I am writing to reply | to the letter you wrote | to me on 10 June.
Incorrect Error Index	Discrete	5
Incorrect Sentence	Text	I am writing to reply | the letter you wrote | me on 10 June.

Note: The pipe symbol (|) indicates the position of the error.

Table 3.3 shows the top five error types among NLT and non-NLT errors. As shown, the replace punctuation, the missing determiner, and the incorrect tense of verb error types are more frequent. The NLT and non-NLT error categories have two error types in common: replace punctuation and incorrect tense of verb are among the top five error types across both categories. This indicates that, in general, Chinese language learners make these errors more frequently than other error types when writing in English.

Table 3.3: The top five NLT and non-NLT errors for the Chinese FCE structural error subset

Category	Error Code	Error Type Label	Frequency
Negative Language Transfer	RP	Replace Punctuation	228 (15%)
	MD	Missing Determiner	206 (14%)
	TV	Incorrect Tense of Verb	185 (13%)
	MP	Missing Punctuation	138 (9%)
	AGV	Verb Agreement	75 (5%)
Non-negative Language Transfer	RP	Replace Punctuation	108 (12%)
	TV	Incorrect Tense of Verb	82 (9%)
	FV	Verb Form	74 (8%)
	UD	Unnecessary Determiner	71 (8%)
	UP	Unnecessary Punctuation	59 (7%)

3.1.2 Farsi Lang-8 Dataset

The Farsi Lang-8 dataset is composed of 129 distinct English texts written by 31 Farsi native speakers. It has a total of 2,991 sentences. Of these, 50.35% are associated with NLT and non-NLT errors, whereas 49.65% do not contain an error. Each instance contains the learner's English writing retrieved from Lang-8.com and ten more features associated with it. A learner's writing may contain one or more sentences. As my goal is to detect which error is related to NLT, I have excluded the samples whose negative transfer labels are neither true nor false.

As shown in Table 3.4, the dataset is imbalanced. Akin to the Chinese FCE dataset, the Farsi Lang-8 error subset was filtered to only contain structural errors. The gap between the majority class (non-NLT errors) and the minority class (NLT errors) is 68%, prompting the use of an oversampling method (Abd Elrahman and Abraham, 2013). Error-type distributions for the original, error subset, and structural error subset are provided in Table 3.4. An instance from the Farsi Lang-8 dataset is shown in Table 3.5. The example represents an NLT occurrence of the missing determiner (M:DET) error type. The error information associated with each erroneous sentence is represented in the "Error" column which contains the type and the correction of the error.

Table 3.4: Error distribution in the Farsi Lang-8 dataset

Dataset	**NLT Errors**	**Non-NLT Errors**	**No Error Annotation**	**Total Samples**
Original	181 (6%)	1,325 (44%)	1485 (50%)	2,991
Error Subset	181 (12%)	1,298 (88%)	0 (0%)	1,497
Structural Errors	131 (16%)	709 (84%)	0 (0%)	840

Table 3.5: An example from the Farsi Lang-8 dataset

Column	**Data Type**	**Example**
Raw Sentence	Text	Especially for the people who has good sense of humour.
Error	Text	M:DET | | | a | | | REQUIRED | | | -NONE- | | | 0
Has Error	Nominal	True
Incorrect Sentence	Text	Especially for the people who have | good sense of humor.
Correct Sentence	Text	Especially for the people who have a good sense of humor.
L1	Nominal	Farsi
Link	Nominal	http://lang-8.com/73510/journals/ 690218
Indices	Interval	6 6
Grammatical Error	Nominal	1
NLT	Nominal	1
Comment	Nominal	-

Note: The triple pipe symbol (|||) is a separator for the error information.

Table 3.6 shows the top five error types in the Farsi Lang-8 structural error subset. As shown, missing determiner and missing punctuation are the most frequent error types across NLT and non-NLT errors. The only error type that is common across the two categories is the replacement error for noun-number. Unnecessary preposition and replacement errors for verb tense are among the other frequent error types of the Farsi Lang-8 structural error subset.

Table 3.6: The top five NLT and non-NLT errors for the Farsi Lang-8 structural error subset

Category	Error Type	Description	Frequency
Negative Language Transfer	M:DET	Missing:Determiner	53 (40%)
	U:PREP	Unnecessary:Preposition	14 (11%)
	R:NOUN:NUM	Replacement:Noun Number	13 (10%)
	U:OTHER	Unnecessary:Other	7 (5%)
	R:PART	Replacement:Particle	6 (5%)
Non-negative Language Transfer	M:PUNCT	Missing:Punctuation	96 (14%)
	R:VERB:TENSE	Replacement:Tense of Verb	47 (7%)
	R:NOUN:NUM	Replacement:Noun Number	46 (6%)
	M:OTHER	Missing:Other	45 (6%)
	R:NOUN	Replacement:Noun	41 (6%)

Oversampling

As shown in Table 3.4, the class distribution of the Farsi Lang-8 dataset is skewed. This bias in the training set can induce the machine learning algorithm to ignore the minority class and predict the majority class. A common approach to address class imbalance is to randomly resample the training dataset to rebalance the class distribution. I used this approach by applying the Python library `Imblearn`[1]. As a result, I oversampled the minority class so that the ratio of the minority class to the majority class would be equal to 0.5; 50% was selected based on several attempts at model training. A percentage of oversampling higher than 50% caused the model to overfit and to memorize the instances instead of learning the underlying pattern in the data. Table 3.7 shows the frequency of the NLT errors before and after applying random oversampling.

Table 3.7: Class frequency before and after random oversampling

	NLT	Non-NLT
Original	131	709
Oversampled	354	709

[1]https://imbalanced-learn.org/stable/

3.1.3 Parallel Corpora for Language Modeling

Machine learning models equipped with POS tags can detect the occurrence of NLT using information about the L1 and L2 language structures. Thus, a POS-tagged corpus of Chinese and Farsi text with a parallel translation in English could help express the structure of the languages. Moreover, it can help the model find the patterns that support NLT identification. As shown in Table 3.8, three parallel corpora were chosen. The parallel corpora contain the same number of sample phrases across their respective language pair.

The first corpus, "Global Voices", is a Chinese-English corpus that contains news stories from around the world that were published on the Global Voices website. The second corpus, WMT19, is a Chinese-English corpus that was released in 2019 during the fourth conference on machine translation; it contains parallel news stories from online websites. The combination of the Global Voices and WMT19 was used to model the Chinese language. The third corpus, MIZAN, is the largest Farsi-English parallel corpus. It contains more than a million Farsi-English sentence pairs collected from masterpieces of literature that were made available through Project Gutenberg (Kashefi, 2018).

These datasets were used as the training and validation data sources for language modeling tasks since they can be used to extract a representation of the grammatical structure of the Chinese and Farsi languages. Sentences that did not have a corresponding match in English were removed. Table 3.8 shows the dataset names, their corresponding languages, and the number of sentence pairs they contain. The two Chinese-English corpora were merged to compose a single parallel corpus.

Table 3.8: Dataset sizes for the parallel corpora used to train the theory-based models

Dataset	**Languages**	**Number of Sentence Pairs**
Global Voices	Chinese-English	138,582
WMT19	Chinese-English	11,960
MIZAN	Farsi-English	1,021,596

3.1.4 Preprocessing

The Chinese FCE and Farsi Lang-8 datasets were used to train and evaluate the non-theory-based models. Both datasets were analyzed and annotated by trained Chinese, Farsi, and English native speakers to minimize the occurrence of annotation errors. Preprocessing steps such as stop-word removal, stemming, and tokenization that would alter the structure of the text were not employed. To identify the occurrence of NLT, it is important to capture grammatical and syntactical aspects of the sentence while preserving its original structure. The Chinese FCE and Farsi Lang-8 datasets contain language learner errors each of which is called an *Incorrect Sentence*. Given an *Incorrect Sentence*, four features were extracted to ensure that the error structure is being preserved and passed on to the algorithms.

One of the standard representations that captures the information and syntactical structure of a raw text is the POS tags that are associated with each word in a text. These tags distinguish the grammatical properties of words. POS tagging is the process of assigning specific labels of word categories to tokens of a text (Jurafsky and James H. Martin, 2009). The assigned word categories represent the syntactic function of the token within a text.

The *Universal Dependencies* tagset is a multilingual treebank collection that provides cross-linguistically consistent tags for 33 languages (Nivre et al., 2016). This tagset uses 17 tags to express the syntactic properties that are shared across languages. The *Universal Dependencies* tagset is consistent with the goals of this thesis as it can POS-tag all three languages (i.e., English, Chinese, and Farsi) using the same set of tags. In addition to the *Universal Dependencies* POS tags, I have employed `Spacy's`[2] dependency parser to tag the Chinese FCE and Farsi Lang-8 datasets with syntactic dependency labels. Dependency parse trees contain information that can be used to represent sentence structure. They represent the relations between different constituents in a sentence.

Figure 3.1 shows *Universal Dependencies* POS and dependency relation

[2]https://spacy.io/

tag trigrams from a randomly chosen sentence of the Chinese FCE dataset. As shown, "DET" is used to indicate the determiner that introduces the main subject, "NOUN" is used to represent the word images which is the subject of the sentence, and "VERB" is used to annotate the main verb in the sentence. Also, "det" is used to represent the relation determiner, "nsubj" is used to indicate the subject of the clause, and "root" is used to represent the root of the sentence. An n-gram is a sequence of "n" tokens. Computing the probabilities of n-grams can represent how often a sequence occurs in a corpus. Each POS tag trigram starts with the POS tag of the first word of the error and the two subsequent POS tags. Table 3.9 provides a list of *Universal Dependencies* tags.

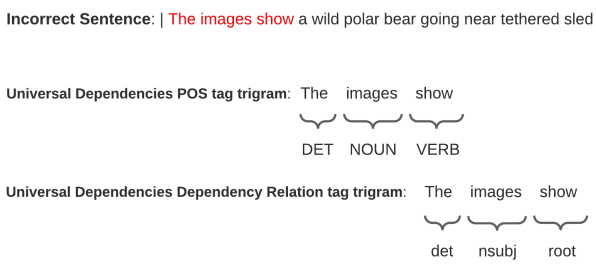


Figure 3.1: An example of the *Universal Dependencies* POS and dependency relation tag trigrams

Table 3.9: List of *Universal Dependencies* tags

Tag	**Description**
ADJ	Adjective
ADP	Adposition
ADV	Adverb
AUX	Auxiliary
CCONJ	Coordinating Conjunction
DET	Determiner
INTJ	Interjection
NOUN	Noun
NUM	Numerical
PART	Particle
PROP	Pronoun
PROPN	Proper Noun
PUNCT	Punctuation
SCONJ	Subordinating Conjunction
SYM	Symbol
VERB	Verb
X	Other

The logistic regression and random forest algorithms were trained using the following features: *Error Length*, *Error Type Dummy Variables*, *Universal Dependencies POS Tag Trigram Dummy Variables*, and *Universal Dependencies Dependency Relation Tag Trigram Dummy Variables* to identify NLT. Nominal features such as (*Error Type*, *Universal Dependencies POS Tags*, and *Universal Dependencies Dependency Relation Tags*) were converted to numerical features using dummy variables prior to building the models.

The Chinese-English parallel corpora sentences were tagged by `Spacy's` *Universal Dependencies* POS tagger. The `Spacy` library does not support the Farsi language, thus, the Mizan parallel corpus was POS-tagged using `Stanza's`[3] *Universal Dependencies* POS tagger. The POS tags extracted from the parallel corpora were used to train the language models.

[3]https://stanfordnlp.github.io/stanza/

3.2 Model selection and evaluation procedure

In a machine learning algorithm, there are two types of parameters: those whose values are derived through the training process and can be viewed as the output of the learning process (e.g., weight matrices of a neural network) and those whose values are used to control the learning process of the algorithm and can be viewed as the input of learning, called hyperparameters (e.g., the solvers, penalty, or C in logistic regression). Various approaches can be considered to find the best set of hyperparameters for a learning algorithm, including Manual Search, Random Search, Grid Search, Bayesian Optimization, Genetic Algorithms, and Artificial Neural Networks.

In supervised learning, we aim to predict a target (response) variable using instances (examples) of input features. Two of the common phenomena that occur when training a machine learning model are overfitting and underfitting. Overfitting occurs when the model performs well on training data but fails to generalize well to unseen data (i.e., making the model too complex for the training data by using more parameters than are necessary) (Hawkins, 2004). Underfitting is the opposite of overfitting; it occurs when the model is too simple to learn the underlying pattern in the data (Van der Aalst et al., 2010). Thus, the model's prediction is prone to be inaccurate even on the training data.

One of the most common approaches to avoid the problems of overfitting and underfitting is to evaluate the model using resampling methods, such as cross-validation. In this project, stratified nested cross-validation and K-fold cross-validation were used from the `scikit-learn`[4] package to report the validation error from the tuning process and the test error from the final model. Stratified cross-validation based on the target variable ensures that each fold preserves the ratio of the target variable that was present in the original dataset. Cross-validation over the grid of hyperparameters yields the hyperparameter combination with the lowest validation error. As a result, it selects the collection of parameters with the least amount of overfitting. The

[4]https://scikit-learn.org/stable/

process includes splitting the training set into K distinct subsets called folds. Subsequently, the training process occurs K times: each time, the training is carried out on the combined K-1 folds and the evaluation is carried out on the remaining fold (the validation set). Consequently, there would be K evaluation scores that are usually averaged and reported as the validation set accuracy. It is important to see model selection (e.g., hyperparameter tuning) as a component of the model fitting procedure that should be done independently in each trial to avoid selection bias and to represent best practices in operational usage (Cawley and Talbot, 2010). Nested cross-validation is one of the approaches that can handle hyperparameter tuning and model training while attempting to address overfitting. Nested cross-validation employs a series of train, validation, and test splits by fitting a model to each training and validation split to find the best combination of hyperparameters with respect to the training/validation fold and to provide more accurate performance on the test split.

In this work, I applied nested cross-validation with $K_1 = 10$ (outer loop) and $K_2 = 5$ (inner loop) to support generalizability of the models to unseen data. The model selection and evaluation procedure can be summarized as follows:

1. Partitioning the dataset into K_1 ($i.e., K_1 = 10$) folds
2. For each partition of $K_1 - 1$ training folds and one test fold:
    2.1. Split the train set into K_2 ($i.e., K_2 = 5$) folds
        2.1.1. For each partition of $K_2 - 1$ tuning folds and one validation fold:
            2.1.1.1. Train on the tuning folds using hyperparameters h
            2.1.1.2. Test the model on the validation fold
        2.1.2. Calculate the average performance across all K_2 folds for the hyperparameter combination h
    2.2. Return the hyperparameter combination h_{prime} that maximizes the performance.
3. Using the h_{prime} value found in step 2.2, train on all train data and test on the test data from step 2.

4. Report the average performance across all K_1 test folds

from step 3.

Figure 3.2 demonstrates the partitioning, training, and testing procedure of the non-theory-based models.

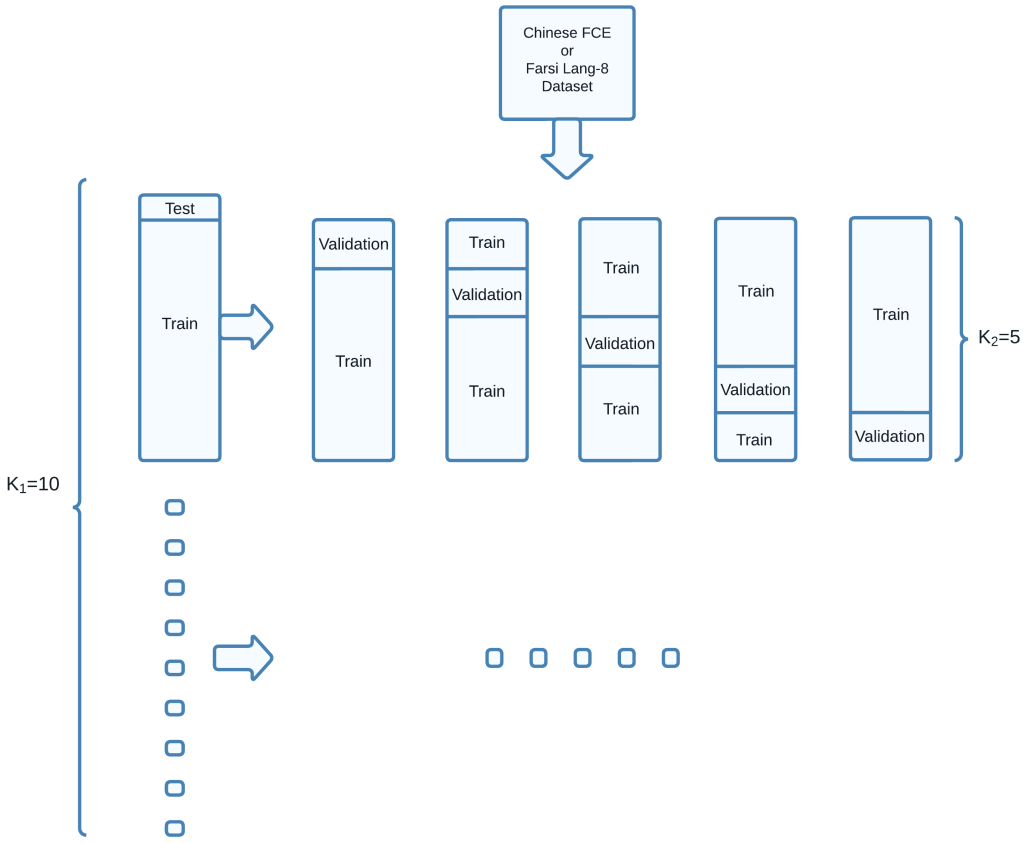


Figure 3.2: Nested cross-validation

3.3 Machine learning models

In this work, I implemented four algorithms to identify NLT in language learner text. The non-theory-based models used were logistic regression and random forest. The theory-based models used were n-grams and an RNN.

Because NLT identification is a binary classification problem, I have used precision, recall, F1-score, and Root Mean Squared Error (RMSE) to evaluate the performance of the models. Due to the imbalanced nature of the datasets, the accuracy metric was ignored during model evaluation.

3.3.1 Non-theory-based Models

This section discusses the training and evaluation procedure of the logistic regression and random forest models used to identify NLT in the Chinese FCE and Farsi Lang-8 datasets. The features (input attributes) and the response variable used to train the non-theory-based models were identical to assist in cross model evaluation. Since this is a binary classification problem and most of the features are nominal, dummy variable conversion was applied to transform the data into binary vectors. *Error length*, *Error type dummy variables*, *Universal Dependencies POS tags dummy variables*, and *Universal Dependencies dependency relation tags dummy variables* are the categories of features that were used to represent the input sentence from learner text.

Logistic Regression

Logistic regression is a commonly used binary classification algorithm. It is a generalized linear model used to predict a dependent variable's probability given one or more independent variables as input (Hosmer Jr et al., 2013). It estimates the probability that an instance belongs to a particular class. For example, logistic regression can help answer the following question: what is the probability that this sentence carries an NLT error? If the predicted probability is greater than 0.5, the model predicts that the instance belongs to the positive (i.e., NLT) class, labeled as 1. If the model predicts a value less than 0.5, the instance belongs to the negative class (i.e., non-NLT), labeled as 0. In logistic regression, the idea is to use a logistic function (i.e., sigmoid) to map the output of a linear equation between 0 and 1 to one of two labels, NLT or non-NLT in this case.

In this work, Python[5] and `scikit-learn` were used to implement the logistic regression model.

[5]https://www.python.org/

Hyperparameter Tuning

Table 3.10 and Table 3.11 show the hyperparameter tuning settings and the selected hyperparameter values of the logistic regression model. The range of values was selected based on the `scikit-learn` website and observations from training the model with different ranges of values.

Table 3.10: Logistic regression hyperparameter options

Hyperparameter	**Description**	**Range of Values**
Solvers	Algorithm used for optimization	Lbfgs, sag, saga, liblinear, newton-cg
Penalty	Choice of regularization	L2
C	Inverse of regularization strength	0.01, 0.03, 0.05, 0.1, 0.3, 0.5, 1, 2, 3, 4, 5, 10

Table 3.11: Logistic regression hyperparameter options

Language	**Fold**	**Solver**	**C**
Chinese	1	Saga	1
	2	Saga	1
	3	Liblinear	2
	4	Sag	2
	5	Sag	0.5
	6	Liblinear	1
	7	Lbfgs	1
	8	Saga	2
	9	Sag	2
	10	Sag	2
Farsi	1	Lbfgs	10
	2	Lbfgs	10
	3	Lbfgs	10
	4	Liblinear	5
	5	Lbfgs	10
	6	Lbfgs	10
	7	Liblinear	10
	8	Lbfgs	10
	9	Lbfgs	10
	10	Lbfgs	2

Random Forest

Random Forest is an ensemble model that consists of a large number of individual decision trees. Decision trees are versatile non-linear prediction algorithms that can perform well on supervised learning tasks (i.e., regression and classification) and are able to fit complex datasets. In a random forest, each decision tree acts as an individual classifier or regressor and affects the final prediction of the forest. Since the decision trees in a forest are immune to multi-collinearity and the random forest prediction takes into account the decision of the individual trees, the performance of the forest is a better estimation than the prediction provided by each tree. Random forest has shown superior performance when dealing with complex data where the number of features *(m)* is greater than the number of instances (observations) *(n)* (Xu et al., 2012). If the dataset contains two features that are highly correlated when deciding upon a split, the decision tree will only select one of them, whereas logistic regression will use both of the features. Figure 3.3 represents the structure of a random forest composed of *(N)* classification trees.

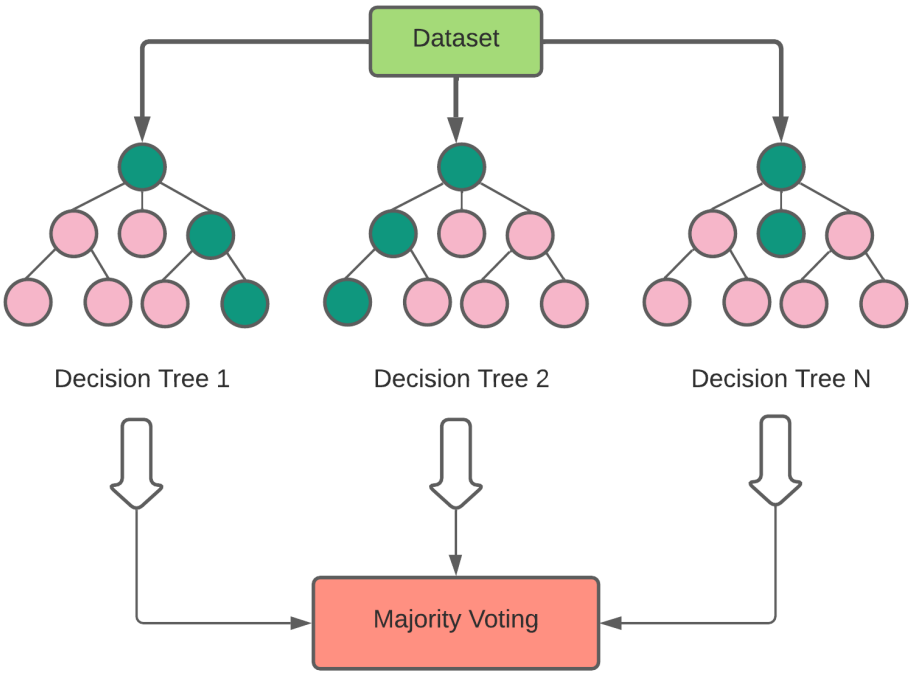


Figure 3.3: The structure of a random forest classifier

Hyperparameter Tuning

The tuning and evaluation process of the random forest model was similar to that of logistic regression.

As shown in Table 3.12, *number of estimators*, *maximum depth*, *maximum features*, and *minimum samples split* are the hyperparameters that were tuned.

Table 3.12: Random forest hyperparameter options

Hyperparameter	**Description**	**Range of Values**
N_estimator	Number of trees	100 to 500 with a step of 20
Max_depth	Maximum depth of the tree	10 to 100 with a step of 5 & None
Max_features	Inverse of regularization strength	Auto, sqrt, log2
Min_samples_split	Minimum number of samples required to split an internal node	2, 3, 4, 5

Table 3.13 reports the selected hyperparameters for each fold from the nested cross-validation process.

Table 3.13: Selected hyperparameters of random forest per outer fold of nested cross-validation

Language	**Fold**	**Max Depth**	**Max Features**	**Min Samples Split**	**N Estimators**
Chinese	1	70	Auto	4	380
	2	65	Auto	5	200
	3	85	Auto	5	160
	4	40	Auto	5	200
	5	70	Auto	5	300
	6	45	Auto	4	300
	7	60	Auto	5	400
	8	55	Auto	4	280
	9	40	Auto	5	500
	10	40	Auto	3	380
Farsi	1	30	Auto	2	220
	2	30	Auto	2	480
	3	55	Log2	2	160
	4	40	Log2	2	140
	5	30	Auto	2	420
	6	40	Auto	2	260
	7	45	Log2	2	240
	8	40	Auto	2	160
	9	45	Log2	2	360
	10	40	Log2	2	500

3.3.2 Theory-based Models

This section introduces the n-gram and RNN language models used to represent Chinese-English and Farsi-English language structures and identify NLT in learner writing. The methodology of this section is based on language modeling using syntactic representations of the text. Theory-based models were trained on large parallel corpora and were evaluated using the Chinese FCE and Farsi Lang-8 datasets.

Using POS tag sequences enables language models to learn the common patterns of a language and to differentiate them. For instance, language models trained using POS information can assign probabilities to sequences of tokens and use those probabilities to identify the source language of a given POS sequence (Farias Wanderley and Demmans Epp, 2021). Theory-based models were trained using POS tag sequences from the parallel corpora and were evaluated using POS tags from the structural errors subset with different error spans.

Different error spans were used to represent the *Incorrect Sentence* written by a learner to ensure that the necessary information for identifying NLT is included. The three distinct formats of the *Incorrect Sentence* not only represent the error itself but also include the context of the error (i.e., words that surround the erroneous token). As shown in Table 3.14, *padded error* span, *error + unigram* span, and *error + bigram* span were used to represent the *Incorrect Sentence*. Also, the error itself may contain one or more tokens. The length of the error depends on the *Error Length* associated with a specific sample from the dataset.

Table 3.14: POS tag span examples

Incorrect Sentence	**Error Length**	**Error Type**	**Padded Error Span**	**Error + Unigram Span**	**Error + Bigram Span**
This are only my immature views.	1	Pronoun Agreement	This are DET AUX	This are DET AUX	This are only DET AUX ADV
Madrid is a big city and have many interesting places.	1	Verb Agreement	and have many CCONJ VERB ADJ	have many VERB ADJ	have many interesting VERB ADJ ADJ
The party will be take place in the Palace Hotel.	2	Unnecessary Verb	will be take place VERB VERB NOUN ADP	be take place VERB NOUN ADP	be take place in VERB NOUN ADP DET

NLT N-gram Language Model

Language modeling uses statistical methods to calculate the probability of a given sequence of tokens (Jurafsky and James H. Martin, 2009). An n-gram, which is a sequence of "n" tokens, is one of the simplest yet most powerful language models that assigns probabilities to sequences of language data. N-grams are commonly used to predict the probability of a current token given the context token(s), which are tokens that precede the current token. N-grams can take different values for "n". A 1-gram or unigram is a one-word sequence, a bigram is a two-word sequence, a trigram is a three-word sequence, and an n-gram is an n-word sequence. For instance, "natural language" and "natural language processing" are examples of a bigram and a trigram, respectively.

The NLT n-gram language models represent the distribution of POS tag sequences derived from the Chinese-English and Farsi-English parallel corpora. The *Universal Dependencies* POS tag sequences were extracted from each of the corpora using the `Stanza` and `Spacy` Python libraries. This resulted in four POS-tagged corpora, each representing the syntactic structure of their corresponding language.

Four n-gram models were trained to identify NLT: two of them represent the Chinese-English parallel corpus and the other two represent the Farsi-English parallel corpus. For each L2 (Chinese or Farsi), one of the models represents the structure of the learner's native language, and the other model represents the structure of the English language.

All of the n-gram language models deployed in this work were trained using Python's `KenLM` [6] library, which is a specialized implementation of the n-gram language model that uses modified Kneser-Ney smoothing. `KenLM` uses hash tables and sorted arrays to train language models with high speed and low memory usage (Heafield, 2011).

The likelihood of POS tag sequences extracted from learner errors was calculated using both the L1 and English language models. Given an input sequence of tokens, each language model outputs the probability of the input

[6] https://github.com/kpu/kenlm

POS tag sequences belonging to the language structure it represents. The resulting likelihoods are then compared to determine whether the error in the input sequence was NLT. The input is flagged as NLT if the probability assigned by the L1 model (Chinese or Farsi) is greater than that assigned by the English model. This process is visualized in Figure 3.4.

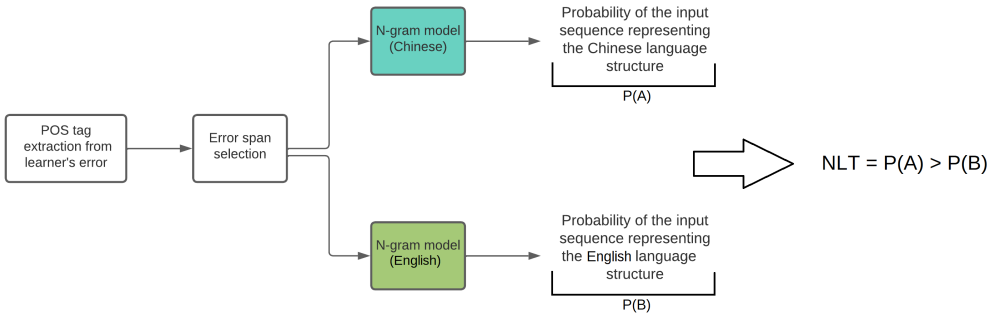


Figure 3.4: Steps in the NLT n-gram language model process

Each POS tag sequence span (*padded error*, *error + unigram*, and *error + bigram*) is evaluated by both the L1 and English language models and the resulting probabilities are compared. As mentioned in the data section, the test dataset contains the learner's erroneous English writing. Thus, when an n-gram sequence is given a greater probability from the L1 model (e.g., Chinese) in comparison to the probability obtained from the English model, that sequence is determined to belong to the Chinese language because the POS tag sequence of the n-gram was more frequent in the Chinese language model than it was in the English language model.

Table 3.15: An example of assigning an NLT label using the n-gram approach

Learner Error	**Chinese Model Output**	**English Model Output**	**NLT**
To: The teacher of English class.	-3.68	-4	True
I am looking forward to hear from you.	-6.57	-5.05	False

Table 3.15 shows the prediction results based on the probability values generated by the model. As shown, the first row of the table represents the occurrence of NLT where the Chinese model assigned a higher probability

to the input sequence compared to the English model, indicating that the structure of the input sequence was more similar to the structure of Chinese than it was to that of English.

Hyperparameter Tuning

The value of "n" is one of the most important factors affecting the performance of an n-gram language model. Therefore, the length of the POS tag sequences analyzed by the n-gram models were tuned to find the parameter setting that best supported the representation of each language's structures. The models used to identify NLT were trained using the best performing n-gram length.

The hyperparameter tuning phase occurred before the training and evaluation processes. The data for each language from the parallel corpora were split into training and validation sets. The dataset containing L1 sentences was used to train its L1 n-gram model, whereas the dataset containing English sentences was used to train the English n-gram model. Five n-gram lengths, from 2 to 6, were attempted.

I used a hold-out method for the hyperparameter tuning procedure of the n-gram model. Each monolingual training dataset was split into an 80:20 ratio of training to evaluation to enable the hyperparameter tuning process. The split ratio was the same across languages for the parallel corpora. The K-fold cross-validation process was applied to the training data to find the best-performing parameter setting based on source language prediction accuracy. Each training corpus was randomly split into five distinct folds. In each iteration, the models (L1 and English) were trained using four folds and were evaluated on the remaining fold. This procedure was applied to each of the n-gram lengths (2 to 6). The best performing n-gram length was selected based on the highest mean accuracy across five iterations.

Table 3.16: Number of sequences in the training and evaluation splits of the parallel corpora

Language	**Training Split**	**Evaluation Split**
Chinese	120,433	30,109
English	120,433	30,109
Farsi	817,276	204,320
English	817,276	204,320

As shown in Table 3.16, the same number of sentences were used in the training and evaluation splits for each L1-English pair when performing hyperparameter tuning. Evaluation splits were used to assess the performance of the models. The hyperparameter combinations for both of the monolingual splits were identical.

The evaluation procedure consisted of two steps. The first step was to compute probabilities of each POS tag in the evaluation split using the L1 and English language models. Then, the estimated likelihoods were compared and the input sequence was labeled as L1 (Chinese or Farsi) or English, depending on which model yielded a higher likelihood. In the second step, outputs from the first step were compared to the input source language to determine the correctness of the model's prediction. If the Chinese language model produced a higher probability than its English counterpart, the input POS tag sequence was classified as Chinese and vice versa. Subsequently, a comparison of the predicted source language with the actual source language of the POS tag sequence was performed to determine whether the prediction was correct.

The evaluation process was repeated for n-grams of length 2 through 6 and the accuracies on the evaluation set were compared to select the best n. The POS tag sequence length that resulted in the best average accuracy was selected to train the models for the NLT identification task. The results of the tuning process are shown in Table 3.17. For both datasets, $n = 5$ (i.e., 5-gram) resulted in the highest mean accuracy on the evaluation set. The superior mean performance is shown in bold.

Table 3.17: Tuning accuracy for the n-gram approach

Model	N	Mean	Median
Chinese-English N-gram	2	0.9578	0.9575
	3	0.9633	0.9634
	4	0.9694	0.9694
	5	**0.9697**	0.9697
	6	0.9688	0.9686
Farsi-English N-gram	2	0.9679	0.9679
	3	0.9731	0.9732
	4	0.9764	0.9763
	5	**0.9775**	0.9774
	6	0.9770	0.9770

The main disadvantage of the n-gram language model is that the models representing the English language's structure and the L1's structure are independent. Despite training on a parallel L1-English corpus, the English language model was never exposed to the L1 structure. Therefore, the probability generated by those models only reflects the likelihood of a POS tag sequence belonging to the language represented by the corresponding model. In contrast, a single RNN language model can differentiate between two language structures (i.e., L1 and English).

NLT RNN Language Model

A Recurrent Neural Network (RNN) is a neural network where the output of the previous step's computation constitutes the input to the current step (Jurafsky and J. Martin, 2021). Having access to information from previous and current states can assist the model in making accurate inferences. RNN language models process sentences word by word and preserve a representation of the previously observed words at each time step (Mikolov et al., 2010).

One advantage of employing RNNs for language modeling is the fact that the model preserves a representation of the preceding words. Unlike n-grams where n defines the number of prior tokens that influence the calculation, RNNs can compute the output vector using the whole preceding sequence. These attributes explain RNN's superior performance on many language modeling tasks. In this thesis, I will use a type of RNN called Elman networks.

Elman networks are three-layer neural networks with the addition of context units (Jeffrey L., 1990). See Figure 3.5 for a graphical representation of an RNN.

To enable NLT identification using an RNN, the model was provided with POS tag sequences that had been extracted from L1 and English sentences. Training the RNN using POS tag sequences from both languages allowed the RNN to learn the language to which a POS tag sequence belonged. The RNN language model analyzes the input POS tag sequence in both languages and creates a hidden representation that can differentiate between language structures.

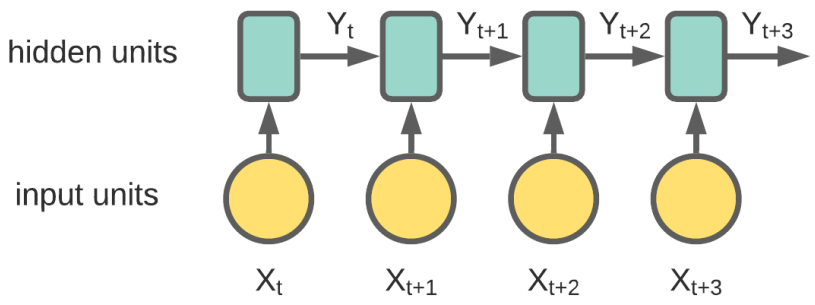


Figure 3.5: Architecture of the NLT RNN language model

Vectorized *Universal Dependencies* POS tags where each POS tag from the UD tagset is represented by a one-hot-encoding vector were used as the inputs to the RNN. Each input vector is 17 units long (17 tags in the UD tagset) and each of those 17 vector positions is equivalent to one of the UD tags. Every training and test sample was transformed into an ordered list of POS tag vectors. The output of the RNN language model is a source language label. If the model is trained with the Chinese-English parallel corpus, the output would be either Chinese or English and if the model is trained with the Farsi-English parallel corpus, the output would be Farsi or English.

To implement the RNN language model, Python's `PyTorch` [7] library was

[7] https://pytorch.org/

used. Each RNN model was trained for 10 epochs using the Adam optimizer (Kingma and Ba, 2014). During the training phase, the RNN model learned to predict the label based on the input POS tag sequence. The model outputs a label, identifying the language to which the POS tag sequence is more similar. The RNN weights were updated through backpropagation by using the difference between the model's output and the ground truth, which is the actual source language of the input POS tag sequence.

The RNN language model was used to detect NLT errors committed by English as a second-language learners after being trained using the ideal hyperparameter combinations. The error was identified as NLT when the RNN language model predicted that a specific POS tag sequence extracted from a learner error was more similar to L1 structures than to English structures.

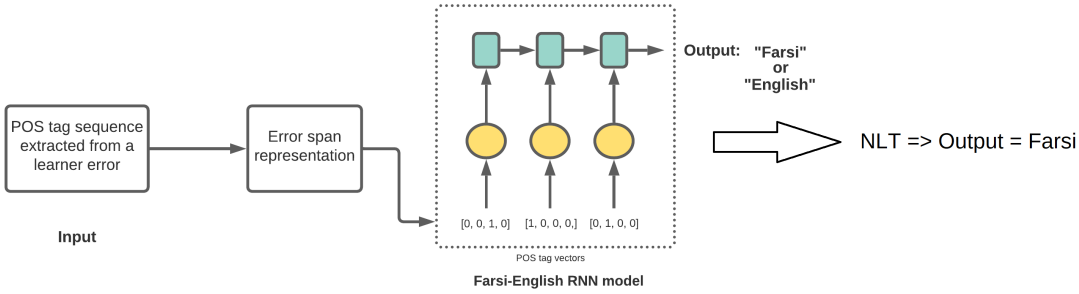


Figure 3.6: Steps in the NLT RNN language model process

Figure 3.6 depicts the general procedure for how a sample from the Farsi Lang-8 dataset advances through the RNN NLT detection process. During the evaluation phase, the language model assigns a label to the POS tag sequence extracted from the language learner error. If the source language prediction label is "Farsi", the error will be classified as NLT. In other words, the structure of the input error would be more similar to Farsi than to English. Then, the NLT prediction label associated with the error is compared to the ground truth label. Based on the comparison, the models' performance can be evaluated by calculating how often the model's output matched the ground truth label.

Hyperparameter tuning

Similar to the three previous models, the RNN model was tuned to select the best hyperparameter combination. The training dataset was divided into training and evaluation sets to enable the hyperparameter tuning process. The training split contained 80% of the training data and the evaluation split contained 20% of the training data.

The input data for each of the RNN models consisted of POS tag sequences extracted from the English-Chinese and English-Farsi sentences in the parallel corpora. Each RNN model was trained with a different hyperparameter combination and learned to predict the source language of the input sequence from the training split.

Table 3.18: RNN hyperparameter options

Hyperparameter	**Description**	**Range of Values**
Number of hidden units	The higher the number of the units, the more complex the algorithm	8, 16, 32, 64, 128, 256, 512
Loss function	A function computing the distance between the current and the expected output	negative log likelihood, binary cross-entropy
Mini-batch size	The amount of data in each weight change epoch	1, 2, 4, 8, 16, 32
Learning rate	A value determining the step size of the algorithm while moving forward to the minimum of a loss function	0.01, 0.001, 0.0001, 0.00001, 0.000001

Table 3.18 shows the hyperparameters used to tune the RNN models. These potential hyperparameter values resulted in a total of 420 possible combinations. Due to the size of the Farsi parallel corpus and the computation time needed to search this full space, I only used 5 different hyperparameter combinations for tuning the Farsi RNN language model, as shown in Table 3.19. Since the Chinese-English parallel corpus was smaller, I was able to

Table 3.19: RNN hyperparameter options for the Farsi parallel corpus

Hidden Units	Learning Rate	Loss Function	Mini Batch-Size
16	0.001	BCE with Logit Loss	4
8	0.001	BCE with Logit Loss	8
16	0.0001	Negative Log Likelihood	1
16	0.0001	Negative Log Likelihood	8
64	0.0001	Negative Log Likelihood	4

assess all 420 combinations for the Chinese RNN language model.

As shown in Table 3.20, 16 hidden units, learning rate = 0.0001, mini batch size = 4, and negative log likelihood as the loss function was the best performing combination of hyperparameters for the Chinese version of the RNN language model. This combination resulted in an accuracy of 95.13% on the evaluation set. For the Farsi version of the RNN language model, 16 hidden units, learning rate = 0.0001, and mini batch size = 1 was selected as the best performing combination of hyperparameters. This combination resulted in an accuracy of 93.5% on the evaluation data. The selected values of mini batch size, 4 and 1, indicates that the weights of the models were updated after processing that number of training samples. The selection of a small value for the learning rate (i.e., 0.0001) prevents overshooting local minima (Buduma and Locascio, 2017).

Table 3.20: Selected hyperparameter values for the RNN on the Chinese and Farsi parallel corpora

Language	Hidden Units	Learning Rate	Loss Function	Mini Batch Size
Chinese	16	0.0001	Negative Log Likelihood	4
Farsi	16	0.0001	Negative Log Likelihood	1

3.4 Model evaluation

3.4.1 Evaluation Criteria

Since all four models are binary classification algorithms, the following evaluation metrics were employed to evaluate them:

- **Accuracy**: The ratio of correctly predicted samples to the total test samples.
- **True Positive**: Cases which were predicted as yes (i.e., has NLT) where NLT was present.
- **True Negative**: Cases which were predicted as no (i.e., does not have NLT) where NLT was not present.
- **False Positive**: Cases which were predicted as yes (i.e., has NLT) where NLT was not present.
- **False Negative**: Cases which were predicted as no (i.e., does not have NLT) where NLT was present.
- **Precision**: The ratio of true positives to the sum of true positives and false positives.
- **Recall**: The ratio of true positives to the sum of true positives and false negatives.
- **F1-Score**: F1-Score or harmonic mean is a balanced measure of precision and recall. F1-Score is an appropriate evaluation measure when the dataset is not balanced in terms of its class variables.
- **Confusion Matrix**: In a binary classification task, a 2×2 table that reports the number of True Positives, False Positives, True Negatives, and False Negatives.
- **Root Mean Squared Error**: A measure of difference between the values predicted by the model and the actual values.

3.4.2 Model Comparison

To investigate whether model performance differed, the models' prediction results were compared using Cochran's Q test (Cochran, 1950; Raschka, 2018). Cochran's Q test is capable of comparing more than two classifiers, thus, it is a generalized version of McNemar's test, which can only compare the results of two classifiers. Cochran's Q test squares the differences between the observed and the expected proportions and divides by the sum of the number of successes multiplied by the number of failures for each case. Cochran's Q test does not provide information about which models differ. It only determines if there is a difference among the models. The null hypothesis (H_0) in Cochran's Q test states that there is no difference between the accuracies of the classifiers (Fleiss et al., 2013).

Performing Cochran's Q test requires a binary n $\times$ M matrix, where M is the number of classifiers and n is the number of test samples. The test was used to compare the prediction results of the four algorithms. I used $\alpha = 0.05$.

When rejecting the null hypothesis, multiple post-hoc pair-wise tests should be conducted to understand which pairs differ. I used correction to control for the increased risk of making a Type I error (rejecting a true null hypothesis) when performing multiple comparisons. I used Dunn's tests with Bonferroni adjustment. Additionally, the maximum corrected effect size, known as eta-squared (η_q^2) (Serlin et al., 1982) was calculated to express how large the differences between the groups were. Effect size does not depend on sample size. This measure of effect size is considered small when its value is close to 0.01. If the value of (η_q^2) is close to 0.06, it is considered to be a medium effect, and when the value is close to 0.14, it is interpreted as a large effect (Cohen, 1988).

3.4.3 Feature Importance

Feature importance refers to assigning numerical scores to input variables of a model to demonstrate how useful they are in predicting the target variable. The coefficients of the non-theory-based models were calculated to understand

the relative importance of each input variable. For the theory-based models, the error spans that resulted in the highest F1-score were reported as the most important features.

3.4.4 Model Error Analysis

Categorizing the models' true and false predictions by error type allowed me to compare, contrast, and identify similar patterns in the results across models. Model error analysis can reveal the error types that the model has difficulty predicting. The analysis of those error types reveals weaknesses of the model.

The Chinese FCE dataset and the Farsi Lang-8 dataset are both annotated by error categories. These categories include a general annotation scheme such as "missing", "replacement", "unnecessary", "wrong format", and "wrongly derived" (Nicholls, 2003). These error type annotations indicate information about the missing POS, unnecessary words, wrong usage of the words with an incorrect form, or words that require replacement. For instance, if a sentence has an error type code of "unnecessary verb", it means that the language learner used a verb that was not required for the sentence to be grammatically correct. As an example from the Farsi Lang-8 dataset, the sentence "if you have look closer" contains an unnecessary verb error (have).

3.5 Summary

This chapter described the data sources, preprocessing techniques, and model selection and evaluation procedures. The next chapter presents the results of applying the trained models on the Chinese FCE and Farsi Lang-8 datasets. It includes an error analysis of the model output.

Chapter 4

Results

This chapter presents the results of binary classification algorithms for NLT identification using non-theory-based models and theory-based (language) models. It also describes the results of the feature importance and model error analysis.

4.1 What is the performance of the proposed models in detecting NLT?

The results of the NLT identification task on the Chinese FCE and Farsi Lang-8 datasets are reported in the next four sections. The first two subsections include the results of the non-theory-based models (i.e., logistic regression and random forest). The next two subsections provide the results of the theory-based models (i.e., n-gram and RNN).

4.1.1 Logistic Regression

Table 4.1 and Table 4.3 present the classification results (weighted F1-score) for all three data splits when applying logistic regression to the Chinese FCE and Farsi Lang-8 datasets, using nested cross-validation. Although multiple evaluation criteria were used, weighted F1-score was used to compare and contrast model performance because of class imbalance in the dataset. As shown in Table 4.2 and Table 4.4, weighted precision, weighted recall, and RMSE were reported to enable a fair evaluation of model performance.

The logistic regression model yielded an average weighted F1-score of 76.5%

with a standard deviation of 0.13 across 10 folds using nested cross-validation on the Chinese FCE dataset. For the Farsi Lang-8 dataset, this model achieved an average weighted F1-score of 84.8% with a standard deviation of 0.39 using the same cross-validation procedure. The highest test F1-score across the ten folds is shown in bold.

Table 4.1: Training, validation, and test F1-scores for logistic regression on each nested cross-validation fold of the Chinese FCE dataset

Fold	**Training F1-score**	**Validation F1-score**	**Test F1-score**
1	81.28	76.68	74.95
2	81.11	76.84	**78.68**
3	81.29	76.20	76.75
4	81.07	76.47	76.28
5	80.13	75.91	76.28
6	81.00	76.54	77.44
7	80.61	75.24	77.60
8	81.31	76.45	77.25
9	81.95	76.58	75.59
10	81.49	76.78	74.13

Table 4.2: Precision, recall, and RMSE for logistic regression on each nested cross-validation fold of the Chinese FCE dataset

Fold	**Precision**	**Recall**	**RMSE**
1	74.86	75.10	0.49
2	78.65	78.90	0.45
3	76.87	77.21	0.47
4	76.44	76.79	0.48
5	76.44	76.79	0.48
6	77.70	77.96	0.46
7	77.66	77.96	0.46
8	77.78	77.96	0.46
9	75.90	76.27	0.48
10	74.58	75.00	0.50

Table 4.3: Training, validation, and test F1-scores for logistic regression on each nested cross-validation fold of the Farsi Lang-8 dataset

Fold	Training F1-score	Validation F1-score	Test F1-score
1	91.69	84.73	84.05
2	92.82	84.33	76.09
3	91.15	84.09	87.07
4	90.20	84.49	82.89
5	91.44	84.89	82.25
6	91.27	85.22	87.68
7	90.86	84.27	**91.63**
8	91.05	84.45	86.17
9	92.20	84.96	84.20
10	89.00	83.45	85.67

Table 4.4: Precision, recall, and RMSE for logistic regression on each nested cross-validation fold of the Farsi Lang-8 dataset

Fold	Precision	Recall	RMSE
1	84.01	84.11	0.39
2	76.97	75.70	0.49
3	87.47	86.91	0.36
4	82.83	83.01	0.41
5	82.56	82.07	0.42
6	87.66	87.73	0.35
7	92.19	91.50	0.29
8	87.75	85.84	0.37
9	84.83	83.96	0.40
10	85.66	85.84	0.37

As the evaluation was conducted using a nested 10-fold cross-validation procedure, 10 distinct confusion matrices were generated for each of the models, showing the number of true positives, true negatives, false positives, and false negatives. The confusion matrix plots can be found in Appendix A.

4.1.2 Random Forest

Akin to logistic regression, Table 4.5 and Table 4.7 represent the classification results of the random forest classifiers applied to the Chinese FCE and Farsi Lang-8 datasets using nested cross-validation.

Table 4.5 and Table 4.7 present the classification results for all three data splits when applying random forest to the Chinese FCE and Farsi Lang-8 datasets, using the nested cross-validation procedure. Table 4.6 and Table 4.8 report the weighted precision, weighted recall, and RMSE of the models.

The random forest classifier had an average weighted F1-score of 78.1% with a standard deviation of 0.20 across 10 folds using nested cross-validation on the Chinese FCE dataset. For the Farsi Lang-8 dataset, this model achieved an average weighted F1-score of 94.8% with a standard deviation of 0.29 using the same cross-validation procedure. The results showing that random forest outperforms logistic regression are consistent with other results in the literature (Farias Wanderley, Zhao, et al., 2021).

Table 4.5: Training, validation, and test F1-scores for random forest on each nested cross-validation fold of the Chinese FCE dataset

Fold	**Training F1-score**	**Validation F1-score**	**Test F1-score**
1	95.22	79.20	76.37
2	94.21	79.39	79.87
3	94.03	77.98	**81.24**
4	91.88	78.62	77.06
5	93.97	77.89	77.66
6	94.21	78.63	77.27
7	94.08	77.81	80.58
8	94.79	78.58	74.40
9	92.42	78.03	79.21
10	94.68	78.60	77.34

Table 4.6: Precision, recall, and RMSE for random forest on each nested cross-validation fold of the Chinese FCE dataset

Fold	**Precision**	**Recall**	**RMSE**
1	76.37	76.37	0.48
2	79.94	80.16	0.44
3	81.24	81.43	0.43
4	78.25	78.05	0.46
5	78.52	78.48	0.46
6	77.25	77.54	0.47
7	80.77	80.93	0.43
8	75.14	75.42	0.49
9	80.43	80.08	0.44
10	77.72	77.96	0.46

Table 4.7: Training, validation, and test F1-scores for random forest on each nested cross-validation fold of the Farsi Lang-8 dataset

Fold	**Training F1-score**	**Validation F1-score**	**Test F1-score**
1	98.43	91.94	95.39
2	98.22	93.42	92.57
3	98.43	92.51	95.31
4	98.33	91.81	94.37
5	98.64	93.51	85.98
6	98.33	93.43	92.39
7	98.22	92.07	**97.19**
8	98.12	93.14	94.37
9	98.22	93.33	94.33
10	98.43	92.15	95.32

Table 4.8: Precision, recall, and RMSE for random forest on each nested cross-validation fold of the Farsi Lang-8 dataset

Fold	Precision	Recall	RMSE
1	95.89	95.32	0.21
2	92.68	92.52	0.27
3	95.31	95.32	0.21
4	94.48	94.33	0.23
5	86.26	85.84	0.37
6	92.41	92.45	0.27
7	97.39	97.16	0.16
8	94.48	94.33	0.23
9	94.33	94.33	0.23
10	95.53	95.28	0.21

4.1.3 NLT N-gram Language Model

The precision, recall, and F1-scores for the n-gram language model are presented in Table 4.9. The best performance was achieved using *error + unigram* span. *Error + unigram* contains the POS tag extracted from the error and the POS tag of the word that immediately follows the error. The highest F1-scores on the test set are shown in bold. The results of the n-gram language model for the Chinese FCE dataset are consistent with other results obtained in the related literature (Farias Wanderley and Demmans Epp, 2021), where the *error + unigram* span obtained the highest F1-score.

Table 4.9: Precision, recall, and F1-score of the n-grams for the Chinese FCE and Farsi Lang-8 datasets

Language	Span	Precision	Recall	F1-score
Chinese – English	Padded error	0.68	0.32	0.44
	Error + unigram	0.65	0.37	**0.47**
	Error + bigram	0.62	0.25	0.36
Farsi – English	Padded error	0.16	0.21	0.18
	Error + unigram	0.24	0.37	**0.29**
	Error + bigram	0.22	0.28	0.24

4.1.4 NLT RNN Language Model

Table 4.10 provides the NLT error identification performance of RNN language models when they were applied to the data from Chinese and Farsi second-

language learners. In the Chinese-English RNN language model, the span which consists of the POS tag of the error token and the next POS tag (i.e., *error + unigram*) yielded the highest F1-score (0.57) and the highest recall (0.5). This result is consistent with the results reported by Farias Wanderley and Demmans Epp (2021), where the *error + unigram* yielded the highest F1-score. In contrast, the Farsi language model did not perform well on the NLT detection task; yielding low precision and recall.

Table 4.10: Precision, recall, and F1-score of the RNNs for the Chinese FCE and Farsi Lang-8 datasets

Language	Span	Precision	Recall	F1-score
Chinese – English	Padded error	0.68	0.35	0.46
	Error + unigram	0.68	0.50	**0.57**
	Error + bigram	0.67	0.33	0.44
Farsi – English	Padded error	0.13	0.17	0.15
	Error + unigram	0.16	0.20	0.18
	Error + bigram	0.18	0.21	**0.19**

4.2 What is the performance of the non-theory-based approaches compared to the theory-based approaches?

The results on the test datasets (Chinese FCE and Farsi Lang-8) of the four models were compared using Cochran's Q test to find the best performing algorithm for each language.

4.2.1 Chinese FCE dataset

Cochran's Q test indicated there was a significant difference in the performance of the four models on the Chinese FCE dataset ($Q = 970.49$, $p < .001$, $\eta_q^2 = .140$). To further investigate the source of this difference, Dunn's post-hoc tests were conducted. As shown in Table 4.11, pairwise comparisons using Dunn's tests with Bonferroni correction revealed that all pairs except logistic regression and random forest had significantly different performance. The performance results show that random forest and logistic regression outperformed

the other algorithms on the Chinese FCE dataset with average weighted F1-scores of 76.5 and 78.1, respectively. Also, the RNN outperformed n-grams with a F1-score of 0.57 using the *error + unigram* span.

Table 4.11: The p-values of the post-hoc Dunn's tests for the Chinese FCE dataset

	Logistic Regression	**Random Forest**	**N-gram**	**RNN**
Logistic Regression	1	-	-	-
Random Forest	1	1	-	-
N-gram	$< .001$	$< .001$	1	-
RNN	$< .001$	$< .001$	$< .001$	1

4.2.2 Farsi Lang-8 Dataset

Cochran's Q test indicated there was a significant difference in the performance of the four models ($Q = 11.48$, $p = .009$, $\eta_q^2 = .004$). To further investigate the source of the difference in the results, Dunn's post-hoc tests were conducted. As shown in Table 4.12, pairwise comparison using Dunn's tests with Bonferroni correction revealed that only the results of the n-gram and random forest were significantly different from each other, with the random forest outperforming the n-gram language model.

Table 4.12: The p-values of the post-hoc Dunn's tests for the Farsi Lang-8 dataset

	Logistic Regression	**Random Forest**	**N-gram**	**RNN**
Logistic Regression	1	-	-	-
Random Forest	.18	1	-	-
N-gram	1	.04	1	-
RNN	.48	1	.15	1

4.3 What features are important for detecting NLT across approaches and languages?

The Chinese FCE dataset and the Farsi Lang-8 dataset include 242 and 225 input variables, respectively. The input variables are a combination of *error*

length, *error type dummy variables*, *Universal Dependencies POS tag trigram dummy variables*, and *Universal Dependencies dependency relation tag trigram dummy variables*. For the theory-based models, I have reported on the error spans that resulted in the highest performance on those models.

4.3.1 Logistic Regression

The top three important features from the logistic regression model on the Chinese and Farsi datasets based on model coefficients are shown in Table 4.13. As listed, the *error type dummy variables* were among the most important features for logistic regression when it was applied to the Chinese FCE dataset, whereas the *Universal Dependencies dependency relation tag trigram dummy variables* were more important for logistic regression when it was applied to the Farsi Lang-8 dataset.

Table 4.13: The top three features by importance when applying logistic regression to the Chinese FCE and Farsi Lang-8 datasets

Language	**Input Variable**
Chinese	error_type_0_Missing Determiner
	error_type_0_Missing Punctuation
	error_type_0_Missing Preposition
Farsi	incorrect_deps_0_ccomp
	incorrect_deps_2_nummod
	error_type_0_R:Word Order

4.3.2 Random Forest

The top three features from the random forest model on the Chinese and Farsi datasets based on model coefficients are shown in Table 4.14. As listed, error type dummy variables were among the most important features for the random forest model. The table also shows that two of the three important features across languages are the same: including missing determiner error type dummy variable (i.e., error_type_0_Missing Determiner) and error length.

Table 4.14: The top three features by importance when applying random forest to the Chinese FCE and Farsi Lang-8 datasets

Language	**Input Variable**
Chinese	error_type_0_Missing Determiner
	error_type_0_Unnecessary Punctuation
	error_length
Farsi	error_type_0_Missing Determiner
	error_length
	incorrect_ud_tags_0_NOUN

4.3.3 Theory-based Models

Unlike the non-theory-based models, the training procedure for the theory-based models was different. Theory-based models were trained using parallel corpora and the input features were the *Universal Dependencies* POS-tagged sequences of the Chinese and Farsi corpora. In the evaluation procedure, different spans of the erroneous POS-tagged input sequence were used to assess the models. Table 4.9 and Table 4.10 report the error spans that resulted in the highest F1-score on the test dataset. On all of the four combinations of the datasets (Chinese FCE and Farsi Lang-8) and models (n-gram and RNN), the *error + unigram* span which consists of the error token(s) and the following token resulted in the highest performance.

4.4 Model error analysis

4.4.1 Non-theory-based Models

Categorizing the logistic regression and random forest's predictions by error type allows us to compare and contrast model performance to reveal patterns in the errors they make. Model error analysis on the Chinese FCE dataset revealed that the most frequent error types were punctuation replacement, wrong verb tense, and wrong verb form. The most frequent error types in the Farsi dataset were missing determiner, noun number replacement, and unnecessary preposition. These findings were enabled by using the prediction results of all ten test folds from the nested cross-validation procedure.

The following tables show the error types associated with the correct and incorrect model predictions of NLT, respectively. Table 4.16 includes the error types associated with the correct and incorrect predictions of the logistic regression and the random forest on the Chinese FCE dataset. Replace punctuation, incorrect tense of verb, and wrong verb form are among the top incorrect predictions for both the logistic regression and the random forest. It is worth noting that some of the incorrect prediction error types can be seen among the correct predictions of the models. One potential reason for observing similar error types in both correct predictions and incorrect predictions could be related to the diverse patterns associated with some of the error types that might be simple or difficult for the model to predict. For example, replace punctuation is the most frequent error type in the Chinese FCE dataset. Thus, it is expected to observe the existence of this error type in both the correct and incorrect prediction error types. An investigation of the replace punctuation error type from the Chinese FCE dataset showed the diversity of its patterns. Replace punctuation errors in the Chinese FCE dataset contain inappropriate capitalization errors, incorrect use of punctuation mark errors, overcorrection errors, and spelling errors which include more patterns and will add more complexity to the patterns associated with this error type. The same argument holds for the presence of the verb form and verb tense in the incorrect predictions and correct predictions. As shown in Table 4.15, a closer analysis using `Spacy`'s morphological features revealed the verb form and tense diversity in the Chinese FCE dataset.

Table 4.15: Verb form and tense statistics from the Chinese FCE dataset

Verb Form and Tense	**Frequency**
Non-3rd Person Singular Present	753
Past Tense	599
3rd Person Singular Present	397
Modal Verbs	358
Gerund or Present Participle	170
Base Form of Verb	31
Past Participle	24

As an error example of the Chinese FCE dataset, "The teacher use video" with a wrong verb form error type was incorrectly predicted. The potential reason that led to the misprediction of the sentence is the limited ability of the model to understand the correct structure of the sentence based on the *Universal Dependencies*, which will be elaborated in the Discussion chapter.

Table 4.16: The top three error types that were correctly and incorrectly identified as NLT when applying the non-theory-based models to the Chinese FCE dataset

Model	**Incorrect Prediction Error Types**	**Correct Prediction Error Types**
LR	Replace Punctuation (17%)	Replace Punctuation (13%)
	Incorrect Tense of Verb (15%)	Missing Determiner (11%)
	Wrong Verb Form (9%)	Incorrect Tense of Verb (10%)
RF	Replace Punctuation (16%)	Replace Punctuation (14%)
	Incorrect Tense of Verb (12%)	Incorrect Tense of Verb (11%)
	Wrong Verb Form (10%)	Missing Determiner (11%)

Note: LR- Logistic Regression; RF- Random Forest

Table 4.18 presents the top error types observed in the correct and incorrect predictions of the logistic regression and random forest models for the Farsi Lang-8 dataset. The most common incorrect prediction for both of the models is the missing determiner error type, which can be seen in both incorrect and correct predictions of the model. Next is the replacement noun number error type that exists in the correct predictions as well. Most of the replacement noun number errors in the Farsi Lang-8 dataset are concerned with the choice of singular or plural noun, whereas the Universal Dependencies can only represent different types of nouns (i.e., singular or plural) with NOUN. As an erroneous example, "Many people in different nation" was classified incorrectly. The reason could potentially stem from the inability of the model to learn the corresponding noun number due to the limited manner in which nouns are represented through POS tags.

As shown in Table 4.17, the Farsi Lang-8 dataset also has diversity in verb form and tense. Detailed statistics of the verb form and tense for the Farsi Lang-8 dataset is provided in Table 4.17.

Table 4.17: Verb form and tense statistics from the Farsi Lang-8 dataset

Verb Form and Tense	**Frequency**
Non-3rd Person Singular Present	295
Past Tense	217
3rd Person Singular Present	139
Modal Verbs	108
Base Form of Verb	47
Gerund or Present Participle	25
Past Participle	1

Table 4.18: The top three error types that were correctly and incorrectly identified as NLT when applying the non-theory-based models to the Farsi Lang-8 dataset

Model	**Incorrect Prediction Error Types**	**Correct Prediction Error Types**
LR	Missing Determiner (20%)	Missing Punctuation (13%)
	Replace Noun Number (12%)	Missing Determiner (12%)
	Unnecessary Preposition (11%)	Replace Verb Tense (6%)
RF	Missing Determiner (43%)	Missing Punctuation (12%)
	Unnecessary Preposition (14%)	Missing Determiner (8%)
	Replace Noun Number (11%)	Replace Noun Number (7%)

An analysis of the Incorrect Tense of Verb from the Chinese FCE dataset and Replace Verb Tense from the Farsi Lang-8 dataset was conducted to investigate the performance of the models in predicting these error types. Compared to the Chinese FCE dataset, the Farsi Lang-8 dataset was smaller in size and had a smaller number of verb-related error types. Since fewer verb patterns were included in the Farsi Lang-8 dataset, it was expected that models trained on the Farsi Lang-8 dataset could potentially have a higher prediction accuracy in identifying verb patterns. Table 4.19 shows the higher prediction accuracy of the models trained on the Farsi data in identifying NLT errors related to the tense of the verb.

Table 4.19: Analysis of the verb tense prediction accuracy for the Chinese FCE and Farsi Lang-8 datasets

Dataset	**Error Type**	**Model**	**Prediction Accuracy**
Chinese FCE	Incorrect Tense of Verb	LR	0.69
		RF	0.77
Farsi Lang-8	Replace Verb Tense	LR	0.88
		RF	0.94

4.4.2 Theory-based Models

The best performing n-gram model on both the Chinese FCE and Farsi Lang-8 datasets included the *error + unigram* span. The NLT detection task results for these models were analyzed to understand the prediction errors. Similar to the logistic regression and random forest error analysis, analyzing the error types found in the correct and incorrect predictions can help identify possible causes of poor model performance.

Table 4.20: The top three error types that were correctly and incorrectly identified as NLT when applying the theory-based models using Error + Unigram span to the Chinese FCE dataset

Model	**Incorrect Prediction Error Types**	**Correct Prediction Error Types**
N-gram	Replace Punctuation (15%)	Replace Punctuation (13%)
	Incorrect Tense of Verb (12%)	Incorrect Tense of Verb (10%)
	Missing Determiner (9%)	Missing Determiner (9%)
RNN	Replace Punctuation (17%)	Replace Punctuation (12%)
	Incorrect Tense of Verb (11%)	Incorrect Tense of Verb (11%)
	Missing Punctuation (9%)	Missing Determiner (10%)

Table 4.20 shows the top three error types among correct and incorrect predictions for the n-gram and RNN language models applied to the Chinese FCE dataset. The table also shows the similarities between the correct and incorrect predictions of n-gram and RNN models. The two most common error types across both of the correct and incorrect predictions are the same. Akin to Table 4.16, similar error types are present in both columns. Replace punctuation, missing determiner, and incorrect tense of verb are the top

three frequent error types in the Chinese FCE dataset. The top three error types observed in the correct and incorrect predictions of the theory-based and non-theory-based models are similar. The replace punctuation errors of the Chinese FCE dataset include inappropriate capitalization errors, incorrect use of punctuation mark errors, overcorrection errors, and spelling errors which share different structures and patterns. The incorrect tense of verb error type samples contain grammatical structures which are diverse and difficult for the model to identify given the limited specificity of the *Universal Dependencies* tagset.

Table 4.21: The top three error types that were correctly and incorrectly identified as NLT when applying the theory-based models using Error + Unigram and Error + Bigram spans to the Farsi Lang-8 dataset

Model	**Incorrect Prediction Error Types**	**Correct Prediction Error Types**
N-gram	Missing Determiner (15%)	Missing Punctuation (14%)
	Replace Noun Number (11%)	Missing Determiner (9%)
	Missing Other (8%)	Replace Verb Tense (7%)
RNN	Missing Determiner (21%)	Missing Punctuation (12%)
	Missing Punctuation (10%)	Replace Verb Tense (7%)
	Replace Noun Number (9%)	Missing Other (6%)

Table 4.21 shows the top incorrect predictions of the n-gram and RNN language models using the *error + unigram* and *error + bigram* spans on the Farsi Lang-8 dataset. As shown, the missing determiner and replace noun number error types exist in both models' incorrect predictions. Also, missing punctuation and replacement of verb tense are present in the correct prediction error types.

4.5 Summary

This chapter reported the results of the proposed methodology to address the research questions provided in the Introduction. In the first section, the performance of the models on the Chinese FCE and Farsi Lang-8 datasets was reported. The second section compared and contrasted the results of the non-

theory-based and theory-based models using statistical analysis. The third section reported the top important features for identifying NLT. Finally, the last section reported the error types associated with the correct and incorrect predictions of the models.

The next chapter discusses the answers to the three research questions. Also, it provides implications, limitations, and directions for future research.

Chapter 5

Discussion

In this chapter, I interpret and discuss the results of the non-theory-based models (logistic regression and random forest) and theory-based models (n-gram and RNN). The chapter also provides this work's limitations, implications, and future research directions.

5.1 What is the performance of the proposed models in detecting NLT?

The eight models used in this thesis were evaluated. Non-theory-based models were trained using the Chinese FCE and the Farsi Lang-8 datasets and were evaluated using stratified nested cross-validation to not only ensure a fair evaluation and reduce overfitting but also to find the best hyperparameter combination. Logistic regression was employed as it is a common baseline for binary classification problems and it is a linear model. In contrast, random forest was used as it is a non-linear model and is known to perform well with high dimensional data (Niu, 2020; Xu et al., 2012). Both models were capable of identifying NLT errors with high precision and recall scores.

The theory-based n-gram and RNN models were trained using parallel corpora. A Chinese-English parallel corpus was used to train the models for Chinese learners of English, and a Farsi-English parallel corpus was used to train the models for Farsi learners of English. These models were evaluated using the Chinese FCE and the Farsi Lang-8 datasets. Language models were trained with various spans (i.e., unigram, bigram, padded) of POS tag sequences to

represent each language's syntactic structure. Both language models had high precision and recall scores on the evaluation split. However, they did not perform well on the test set. Akin to logistic regression, n-grams were used as a common baseline for language modeling. N-grams have shown promising results in grammatical error detection and NLT identification (Farias Wanderley and Demmans Epp, 2021; Lee et al., 2014). However, the n-gram-based model had a drawback. Each n-gram language model (i.e., Chinese or Farsi) consists of two language models, one that represents the L1 (Chinese or Farsi) and another that represents the L2 (English). Let us consider the n-gram language model for the Chinese-English parallel corpus. Although the Chinese and the English language models were trained on parallel corpora, they were independent of each other. The L1 and L2 language structures were never exposed to one another and could not represent the structural similarities and differences between the two languages. Thus, the probability yielded by each of the models only expressed the likelihood of a POS tag sequence belonging to the language structure represented by that model. As a result, an RNN was employed to distinguish between two languages (L1 and L2) using a single language model.

The RNN language model outperformed the n-gram language model on the Chinese FCE dataset. However, it had poor performance on the Farsi Lang-8 dataset. Due to the large size of the Farsi corpus, I was only able to train and tune a few hyperparameter combinations, whereas the size of the Chinese parallel corpus allowed me to train and tune the RNN with 420 different hyperparameter combinations. In addition, there is also a significant difference between the participants of the two datasets. The Lang-8 dataset contains English sentences written by Farsi native speakers who are about to learn English on a language forum, while the Chinese FCE dataset contains English sentences written by Chinese native speakers who are taking an English proficiency exam and are more prepared. As a result, this could be a potential reason for observing error types such as M:OTHER, R:OTHER, and U:OTHER, where the dataset annotators could not assign the associated error type of the erroneous sentence to a specific error type. These reasons could

explain why the performance of the theory-based models for the Farsi language was not as high as the performance on the Chinese FCE dataset.

To further investigate the performance of each model, an error analysis of the results was conducted. Table 4.16 represents the top true and false predictions of the logistic regression and random forest on the Chinese FCE dataset. The existence of such replacement errors in the true prediction error types is a side effect of using *Universal Dependencies* (UD) tagset which only contains 17 tags. For example, using the UD tagset, a tagger assigns the label "PUNCT" to all different kinds of punctuation marks. Therefore, it is not possible to differentiate a period or a comma from a question mark using the learner's error sequence. The misprediction of samples containing a wrong tense of verb could also communicate that the POS tag was not able to represent and distinguish differences between verbs and verb tenses. The UD tagset contains "VERB" and "AUX" to represent verbs and auxiliaries. Although the UD tagset provides some morphological annotations that deliver information on the tense of the verbs, these features are not common across Chinese and English and are not included by the main tagset (Farias Wanderley, 2021). Also, the `Stanza` library that was used to tag the Farsi corpus only supports a limited number of verb features across Farsi and English. Thus, with such a general level of verb annotation, the model will not learn the patterns required to distinguish the error types from each other.

The top error types found in the true and false predictions of the logistic regression and random forest models when they were applied to the Farsi Lang-8 dataset were shown in Table 4.18. Identification of NLT in learner errors containing a missing determiner does not arise from the POS tag limitation. The missing determiner error type occurs when a determiner is missing before a noun phrase. In Farsi, the usage of determiners is not the same as in English. In English, a word is used before a noun group to indicate whether the noun phrase refers to a specific or general subject and it also indicates number. However, in Farsi, determiners can either follow or precede the noun group to clarify the noun reference to the subject. This could be a reason for the occurrence of such errors in writing English text.

Compared to the Chinese FCE dataset, the Farsi Lang-8 dataset provides more general error types. For instance, "U:OTHER" represents the category of unnecessary actions that was not a fit to any specific unnecessary action, e.g., "U:PREP" (unnecessary preposition). This generalization can make it difficult for the model to detect the patterns in learner errors, as the category may contain several different error types. The same argument holds for false predictions of "M:OTHER" and "U:OTHER". These error types comprise 8.4% of the NLT errors from the Farsi Lang-8 dataset.

Table 4.20 includes the top true and false prediction error types of the n-gram and RNN language models on the Chinese FCE dataset. Replace punctuation was the most common error type across both classes. An important factor to consider is that learner errors and the corpora used to train and evaluate the n-gram models were POS-tagged using a *Universal Dependencies* tagger. The *Universal Dependencies* tagset assigns the "PUNCT" label to commas, periods, and other symbols. This lack of annotation specificity reduces the precision of the model because it fails to provide information about different punctuation patterns. Additionally, the usage of punctuation marks in Chinese and Farsi is different from that of English (Liu, 2011). In Chinese, commas are used as sentence boundaries to separate independent clauses and to indicate pauses. In Farsi, commas have several applications. They are used as separators of similar items in a series. They can be used between two dependent clauses, and they are also used to indicate when a pause should be taken to help the reader understand the primary intent of the text. In English, commas are used to separate independent clauses and can be used after a subordinate clause or phrase. The second mispredicted error type was incorrect verb tense. The *Universal Dependencies* tagset does not provide POS tags for verb tense identification, which is one of the main reasons that this error type occurs in both classes.

The next most common error types across the false prediction class were missing determiner and missing punctuation. The occurrence of a missing determiner is related to language transfer and it is not related to the employed tagset. Chinese is a language that does not have an equivalent for English

determiners (Robertson, 2000). As a result, a Chinese native speaker may miss a determiner when writing in English.

The error analysis results of the n-gram and RNN language models on the Farsi Lang-8 dataset were shown in Table 4.21. The existence of the missing determiner error type in both of the false predictions classes suggests the discussed differences in the usage of determiners across the languages.

The second mispredicted error type is "R:NOUN:NUM" which indicates the incorrect noun-form, as it relates to number agreement. As an example from the Farsi dataset, "Even if you have one thousand son" carries this error type because the word "son" should take the plural form "sons". The occurrence of this error type may stem from the differences between English and Farsi languages. In Farsi, when there is a cardinal number before the noun, the noun comes in its singular form (Swan and Smith, 2001).

The true and false predictions of the n-gram and RNN were similar to each other. Missing determiner and replace noun number are the two most common false predictions for both models. In addition, the true prediction error types across models were similar. Missing punctuation and replace verb tense are the two common error types that were present in both models' true predictions.

5.2 What is the performance of the non-theory-based approaches compared to the theory-based approaches?

Implementing the non-theory-based and theory-based models and evaluating them on identical datasets has allowed me to compare and contrast the results of the algorithms on two different languages to investigate the generalizability of the proposed methodology.

The non-theory-based models obtained higher precision and recall scores than the theory-based models. Results from the nested cross-validation procedure for the non-theory-based models and a set of unseen data for evaluating the theory-based models suggest that the models are generalizing to unseen

data. Results of the theory-based models indicate the possibility of detecting NLT on data that has structural and contextual variations with the training dataset.

The random forest outperformed the n-gram on the data from the learners who speak Farsi as their first language. The main reason behind this observation could be differences in the the training and testing datasets of the theory-based models. In general, the performance of a machine learning model is more robust when the structure of the training and evaluation data are similar. Non-theory-based models (i.e., logistic regression and random forest) were trained, tuned, and evaluated on an error dataset, whereas the theory-based models were trained and tuned on parallel corpora and were evaluated on the Chinese FCE and Farsi Lang-8 datasets.

The differences identified when conducting statistical testing for each language varied in magnitude. The differences detected for the Chinese FCE dataset had a large effect size ($\eta_q^2 = .140$). The difference identified in the Farsi-Lang 8 dataset was small ($\eta_q^2 = .004$).

As reported in the Results chapter, the random forest classifier outperformed other models on both the Chinese FCE and Farsi Lang-8 datasets. The superior performance of the random forest over the logistic regression was expected. Also, the superior performance of the non-theory-based models over the theory-based models can be explained by the consistency between their training and evaluation data. Moreover, non-theory-based models were trained with more grammatical information compared to the theory-based models. This information was provided by *Error Length*, *Error Type Dummy Variables*, *Universal Dependencies POS Tag Trigram Dummy Variables*, and *Universal Dependencies Dependency Relation Tag Trigram Dummy Variables*.

5.3 What features are important for detecting NLT across approaches and languages?

The analysis shows that the most important feature for detecting NLT across the Chinese FCE dataset is the *missing determiner*. This is inline with the

error analysis. The Chinese language does not use a determiner. Beyond that, the non-theory-based models for the Chinese language seem to prioritize the error type dummy variables over other input variables. The top three important input variables to the Farsi non-theory-based models include the *Error Type Dummy Variables*, *Universal Dependencies POS Tag Trigram Dummy Variables*, and *Universal Dependencies Dependency Relation Tag Trigram Dummy Variables*.

The theory-based models were trained using POS tag sequences. The n-gram language model approach that was used to represent the structure of the languages was trained using *Universal Dependencies* POS tag sequences. The n-gram models were evaluated using various error spans of the second-language learner's erroneous writing (i.e., *padded error span*, *error + unigram span*, and *error + bigram span*). The *error + unigram* span (which consists of the POS tag of the error and the POS tag of the next word that follows the error) yielded the best performance. The same pattern was seen for the RNN language model for the Chinese FCE dataset, but the *error + bigram* span performed better in the RNN model for the Farsi Lang-8 dataset. The results of the theory-based models for the Chinese FCE dataset were consistent with previously obtained results (Farias Wanderley and Demmans Epp, 2021).

The inconsistency between the best performing feature of the n-gram and RNN language models for the Farsi Lang-8 dataset may be due to the absence of the best hyperparameter combination for the RNN language model on the Farsi Lang-8 dataset. Unlike the RNN model for the Chinese language, which was tuned using 420 hyperparameter combinations, the Farsi model was tuned using five different hyperparameter combinations. Consequently, it is possible that the best hyperparameter combination for the Farsi dataset was not included in the analyses.

5.4 Limitations

The first limitation of the work is the small size of the Chinese FCE (2,365 sentences) and Farsi Lang-8 (840 sentences) datasets. It is possible that, if the

size of datasets were larger, more language patterns would be used to train the learning algorithms, which would boost performance. In that case, deep learning algorithms, such as Long Short Term Memory networks (LSTMs) could be employed to identify grammatical errors including NLT (Bell et al., 2019).

The methodology used in this thesis requires the language structures to be represented using an identical POS tagset. Employing detailed tagsets such as the Penn Treebank was not feasible because a shared annotation scheme with the Penn Treebank does not exist for English-Chinese nor for English-Farsi. As a result, the *Universal Dependencies* tagset was the only tagset that met the requirements of the methodology.

From a methodological perspective, each of the algorithms used in this thesis was trained using the *Universal Dependencies* tagset to express the structure of the language. With only 17 tags, this tagset is too general to capture the full range of error patterns. Although the *Universal Dependencies* tagset was introduced as a multilingual annotation scheme, its POS tags only represent general word categories and do not have the ability to express grammatical aspects such as number, gender, and tense. The *Universal Dependencies* tagset does not have a specific tag for many POS categories. The categories not covered by this tagset include gerunds, past participles, and singular or plural nouns. This level of POS tag generalization does not differentiate the tense, form, and type of the POS tags which can lead to a deficient representation of a language structure.

Verbs are a POS whose form can change the sentence's meaning. The *Universal Dependencies* tagset only expresses the existence of a verb using two tags: VERB and AUX. Using only two tags for verb identification across English, Chinese, and Farsi, which have significant differences with each other, can mislead the model and inhibit the model's ability to distinguish verb usage patterns. Consequently, the POS tag representation of the learner error was misrepresented by the inherent limitation of the *Universal Dependencies* tagset. Figure 5.1 shows two POS-tagged sentences written by a Farsi native speaker where the second sentence is grammatically incorrect. The verb "like"

does not correspond to the form of the sentence's subject. As shown, the *Universal Dependencies* POS-tagging does not represent the erroneous utterance.

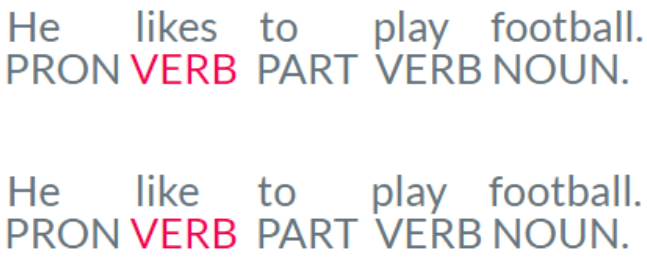


Figure 5.1: *Universal Dependencies* identical POS tagging for the two different sentences

Although a detailed tagset (e.g., Penn Treebank) could help represent language structures that were not represented by the *Universal Dependencies* tagset, there are several NLT errors that can not be distinguished using language structure (e.g., spelling errors, semantic errors). The employed methodology only applies to identifying NLT when it is due to having structural errors in the text.

The resources (i.e., datasets and corpora) that were used in this work can be considered as another limitation. Despite the large size of the Farsi-English parallel corpus, the context of the corpus was not similar to the Farsi Lang-8 dataset.

English, Chinese, and Farsi are three structurally different languages. Farsi is an Indo-European language, which has been influenced by the Arabic language. Farsi and English differ in phonology, punctuation, orthography, and grammar (Swan and Smith, 2001). The Farsi language has a distinct writing system where the words are written from right to left with the letters joining each other based on pre-defined rules. For these reasons, Farsi speakers of English are expected to have difficulty in learning English, especially in the early stages (Swan and Smith, 2001). Although both English and Farsi are Indo-European languages that share some grammatical similarities, there are areas where the grammars of the languages diverge. For example, word order

in Farsi can be different than in English. In Farsi, a sentence usually follows the subject, object, and verb pattern (e.g., "I the movie watched"). However, in English the word order of a sentence is usually subject, verb, and object.

Akin to Farsi and English, there are limited similarities between the syntactic structure of Chinese and English except the word order in Chinese which follows the same pattern as in English. Chinese is a Sino-Tibetan language which has major structural differences with English in phonology, orthography, and grammar (Swan and Smith, 2001). The writing system of the Chinese language is non-alphabetic. All the similarities and the differences among the three languages add complexity to identifying NLT errors.

5.5 Implications

This work shows that the non-theory-based models can be used in unstructured domains (e.g., short essays or text) to identify NLT errors across two different languages. The manual identification of NLT errors can be expensive. It requires expertise and time, which is not always available, especially in large online settings. The evaluation results of the analyses using two categories of models on two languages show the limited generalizability of the methods in detecting NLT errors in unstructured domains.

From a practical perspective, the findings of this study can be used in the automatic provisioning of verification feedback (i.e., information about the correctness of a written text; Shute, 2008) for training and learning in unstructured domains, which is a more challenging task than in structured environments. Using the proposed methodology, the feedback generation system could make language-learners aware of the existence of NLT errors.

5.6 Future work

The rapid development of computer systems and Internet access led to a proliferation of online tools that could benefit from the identification of NLT errors. The findings of this study can be integrated to develop writing assistants and to support online writing classrooms.

Most ESOL learners find it challenging to write in English. While teachers of English as an additional language can provide appropriate feedback for NLT errors, many learners do not have access to such support. Instead, they rely on writing software and computational writing assistants (e.g., Grammarly or Wordtune) in the absence of instructor support. In these settings, a model that can identify NLT errors could support the provisioning of appropriate feedback of varying forms, including verification feedback. The feedback could draw the learners' attention to errors so that the errors could be corrected.

The results of this thesis show that it is possible to detect NLT in two distinct languages. One of the possible future directions would be to use the NLT detection models in a writing assistant program that indicates the occurrence of such errors and provides feedback to inform the learner of the error type and possible solutions. The employed methodology is suitable to address the identification of NLT errors in other languages as the UD tagset supports numerous languages. Moreover, this methodology can also be applied to languages (e.g., Indigenous) that are not supported by most POS tagging libraries. To do so, a corpus of the source language should be manually tagged using a POS tagging method (e.g., rule-based or stochastic), which demands rules be defined based on the linguistic features of the word and its context. Once this has been done, the methods from this thesis could be applied.

The effectiveness of this idea can be explored by conducting a user study in which Chinese or Iranian English-language learners are randomly divided into two groups (i.e., control and treatment) and are given a writing task with and without the existence of the writing assistant tool. The control group is the group that would be provided with corrective feedback from the writing assistant and the treatment group would be provided with an enhanced writing assistant tool that can provide metalinguistic feedback (Farias Wanderley, 2021).

There are several potential ethical concerns that need to be addressed before and during the development and employment of an NLP system (Leidner and Plachouras, 2017). Inclusion and bias require investigation, given that languages define linguistic communities. Also, as most NLP systems depend

on machine learning models, automation and error analysis become other important topics to explore (Leidner and Plachouras, 2017).

Bender et al. (2020) proposed three primary topics that underlie ethical issues in NLP research: dual use, bias, and privacy. Dual usage aims to anticipate how technology could be misused for harmful purposes. Bias aims to identify when a model provides findings that are fundamentally prejudiced as a result of false assumptions made throughout the machine learning process. Privacy aims to protect the written text in the construction or evaluation of an NLP system (Bender et al., 2020).

The standard ethical considerations for an NLP system should be investigated by analyzing the proposed research questions to understand potential misuse. In this thesis, the automatic identification of NLT is intended to assist Chinese and Farsi native speakers with writing in English. However, there are potential cases where the developed models can be misused to identify the source language of the written text. These systems can be employed for discrimination. For example, these models could be potentially used to restrict the employment of Chinese and Farsi native speakers when applying to a job or university using a resume or a cover letter.

The identification of NLT from the English writing of other groups than Chinese and Farsi native speakers will heavily depend on the structural differences between the writer's native language and the learned patterns from the training data. While the performance of the non-theory-based models was noticeably higher than chance, the theory-based models did not perform as well.

In this thesis, two datasets and three corpora were used to train and evaluate the NLT system. Mieskes (2017) proposes questions that should be answered when using public or private datasets. The first question is whether the dataset contains sensitive data. MIT Information Services and Technology recognizes information about race, ethnicity, political and religious views, mental health, and personal life as sensitive data (Mieskes, 2017). In this work, the Chinese FCE and Farsi Lang-8 datasets include text written by Farsi and Chinese native speakers which represents the writing style and the grammat-

ical structure of the Chinese and Farsi native speakers' writing. Thus, the datasets contain potentially sensitive data as the grammatical patterns and writing styles could be misused to identify the nationality of a writer. Moreover, although the datasets do not contain explicit gender information of the writers, the likelihood of gender bias is not negligible. Gender identification tools might be utilized to identify gender and even more detailed information about the writers.

The second question that the article discusses is whether the data is anonymized. The dataset that was used to train and evaluate the models only contains text, written by an anonymous language learner, a participant of the FCE language proficiency exam, or publicly available corpora. Although the dataset does not share the personal information of the writer, it could reveal their nationality which could be misused.

Chapter 6

Conclusion

English is the most widely spoken language worldwide, with more than 1.5 billion speakers (Eberhard et al., 2022). English writing can be challenging for many people, especially for non-native speakers. Although both native and non-native speakers of English make errors while writing in English, the type of errors that non-native speakers make is usually different from the type of errors that native speakers make. Non-native speakers of the English language are used to the grammar of their L1 language. This can intentionally or unintentionally affect language learners when they are communicating in an L2. If the transferred rules from the L1 diverge from those of the L2, errors are introduced. Usually, second-language learners are not aware of the occurrence of NLT.

Despite the fact that there are numerous writing assistant tools that help learners write in English, there is no tool that makes non-native speakers aware of their NLT errors. This thesis presents four machine learning algorithms that can identify when learner errors are related to the structure of their L1 language: logistic regression, random forest, n-gram, and RNN models were trained and evaluated using POS-tagged datasets of second-language learner errors and parallel corpora.

The non-theory-based models, logistic regression and random forest, were trained and evaluated using the Chinese FCE and Farsi Lang-8 datasets. The theory-based models, n-gram and RNN, were trained using POS-tagged parallel corpora of the L1 and L2. POS tags were used to distinguish the structure

of each language. In the evaluation phase, theory-based models were used to analyze the extracted POS tag sequences from the learner errors (Chinese FCE and Farsi Lang-8 datasets) with various spans. The learner error is flagged as NLT if the structure of the error is more similar to the learner's L1.

Across all models, random forest obtained the highest F1-score for both the Chinese and Farsi learner datasets. However, the training approach for the theory-based and non-theory-based models was different. Non-theory-based models were trained and evaluated on the same dataset of second-language learner errors, whereas theory-based models were trained on parallel corpora of structured text and were evaluated on second-language learner errors. Among the theory-based models, the RNN outperformed the n-grams on the Chinese FCE dataset. In contrast, the n-grams performed better than the RNN on the Farsi Lang-8 dataset. This finding could be due to the limited hyperparameter tuning that was performed on the RNN using the Farsi parallel corpus (i.e., 5 different combinations) compared to the RNN using the Chinese parallel corpus (i.e., 420 different combinations).

The implementation of the theory-based models was dependent on the POS tags. The *Universal Dependencies* tagset was chosen as it was the only tagset that is shared across the three languages. The *Universal Dependencies* tagset contains 17 coarse-grained tags. This characteristic of the tagset limited the model's ability to capture the complete incorrect structure of the learner error using POS tags.

Computational and resource limitations may have hindered the theory-based models from identifying NLT using the Farsi Lang-8 dataset. However, in general, the models were able to identify NLT in learner errors. The results of this study can be further used to develop writing tools that alert the second-language learner to such errors; the models could also be extended to provide feedback to second-language learners.

References

Abd Elrahman Shaza M and Ajith Abraham (2013). "A review of class imbalance problem." In: *Journal of Network and Innovative Computing* 1, pp. 332–340.

Akbik Alan, Duncan Blythe, and Roland Vollgraf (Aug. 2018). "Contextual String Embeddings for Sequence Labeling." In: *Proceedings of the 27th International Conference on Computational Linguistics*. Santa Fe, New Mexico, USA: Association for Computational Linguistics, pp. 1638–1649. URL: https://aclanthology.org/C18-1139.

Bacquet Gaston (2019). "Is corrective feedback in the ESL classroom really necessary? A background study." In: *International Journal of Applied Linguistics and English Literature* 8.3, pp. 147–154. DOI: https://doi.org/10.7575/aiac.ijalel.v.8n.3p.147.

Bell Samuel, Helen Yannakoudakis, and Marek Rei (Aug. 2019). "Context is Key: Grammatical Error Detection with Contextual Word Representations." In: *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*. Florence, Italy: Association for Computational Linguistics, pp. 103–115. DOI: 10.18653/v1/W19-4410. URL: https://aclanthology.org/W19-4410.

Bender Emily M., Dirk Hovy, and Alexandra Schofield (July 2020). "Integrating Ethics into the NLP Curriculum." In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*. Online: Association for Computational Linguistics, pp. 6–9. DOI: 10.18653/v1/2020.acl-tutorials.2. URL: https://aclanthology.org/2020.acl-tutorials.2.

Buduma N. and N. Locascio (2017). *Fundamentals of Deep Learning: Designing Next-Generation Machine Intelligence Algorithms*. O'Reilly Media. ISBN: 9781491925584. URL: https://books.google.ca/books?id=80glDwAAQBAJ.

Cawley Gavin C. and Nicola L. C. Talbot (2010). "On Over-fitting in Model Selection and Subsequent Selection Bias in Performance Evaluation." In: *Journal of Machine Learning Research* 11.70, pp. 2079–2107. URL: http://jmlr.org/papers/v11/cawley10a.html.

Chodorow Martin, Joel Tetreault, and Na-Rae Han (2007). "Detection of grammatical errors involving prepositions." In: *Proceedings of the fourth ACL-SIGSEM workshop on prepositions*. Prague, Czech Republic, pp. 25–30. URL: https://aclanthology.org/W07-1604.

Chowdhary KR (2020). *Fundamentals of artificial intelligence*. Springer New Delhi. Chap. 19. DOI: https://doi.org/10.1007/978-81-322-3972-7.

Cochran William G (1950). "The comparison of percentages in matched samples." In: *Biometrika* 37.3/4, pp. 256–266.

Cohen J (1988). *Statistical power analysis for the behavioral sciences*, pp. 18–74. ISBN: 9780203771587. DOI: https://doi.org/10.4324/9780203771587.

Cortés Nuria Calvo (2005). "Negative language transfer when learning Spanish as a foreign language." In: *Interlingüistica* 16, pp. 237–248.

Devlin Jacob et al. (June 2019). "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, pp. 4171–4186. DOI: 10.18653/v1/N19-1423. URL: https://aclanthology.org/N19-1423.

Eberhard David M, Gary F. Simons, and Charles D. Fennig (eds.) (2022). *Ethnologue: Languages of the World. Twenty-fifth edition*. URL: http://www.ethnologue.com.

Farias Wanderley Leticia (2021). "Identifying negative language transfer in learner writing: using syntactic information to model structural differences." MA thesis.

Farias Wanderley Leticia and Carrie Demmans Epp (Apr. 2021). "Identifying negative language transfer in learner errors using POS information." In: *Proceedings of the 16th Workshop on Innovative Use of NLP for Building Educational Applications*. Online: Association for Computational Linguistics, pp. 64–74. URL: https://aclanthology.org/2021.bea-1.7.

Farias Wanderley Leticia, Nicole Zhao, and Carrie Demmans Epp (June 2021). "Negative language transfer in learner English: A new dataset." In: *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Online: Association for Computational Linguistics, pp. 3129–3142. DOI: 10.18653/v1/2021.naacl-main.251. URL: https://aclanthology.org/2021.naacl-main.251.

Fleiss J.L., B. Levin, and M.C. Paik (2013). *Statistical Methods for Rates and Proportions*. Wiley Series in Probability and Statistics. Wiley. ISBN: 9781118625613. URL: https://books.google.ca/books?id=9VefO7a8GeAC.

Fraser Ian S and Lynda M Hodson (1978). "Twenty-One Kicks at the Grammar Horse: Close-Up: Grammar and Composition." In: *English journal* 67.9, pp. 49–54.

Gitsaki Christina (1998). "Second language acquisition theories: Overview and evaluation." In: *Journal of Communication and International Studies* 4.2, pp. 89–98.

Hawkins Douglas M (2004). "The problem of overfitting." In: *Journal of chemical information and computer sciences* 44.1, pp. 1–12.

Heafield Kenneth (July 2011). "KenLM: Faster and Smaller Language Model Queries." In: *Proceedings of the Sixth Workshop on Statistical Machine Translation*. Edinburgh, Scotland: Association for Computational Linguistics, pp. 187–197. URL: https://aclanthology.org/W11-2123.

Heidorn George E. et al. (1982). "The EPISTLE text-critiquing system." In: *IBM Systems journal* 21.3, pp. 305–326.

Hosmer Jr David W, Stanley Lemeshow, and Rodney X Sturdivant (2013). *Applied logistic regression*. Vol. 398. John Wiley & Sons. ISBN: 9781118548387. DOI: 10.1002/9781118548387.

Jeffrey L. Elman (1990). "Finding structure in time." In: *Cognitive Science* 14.2, pp. 179–211. ISSN: 0364-0213. DOI: https://doi.org/10.1016/0364-0213(90)90002-E. URL: https://www.sciencedirect.com/science/article/pii/036402139090002E.

Jiang Xiangying et al. (Aug. 2020). *Duolingo efficacy study*. Research Report DRR-20-04. Duolingo.

Jurafsky Daniel and James H. Martin (2009). *Speech and Language Processing (2nd Edition)*. USA: Prentice-Hall, Inc. ISBN: 0131873210.

Jurafsky Daniel and JH Martin (2021). *Speech and Language Processing, An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition (Third Edition draft)*. USA. Chap. 7. URL: https://web.stanford.edu/~jurafsky/slp3/.

Kashefi Omid (2018). "Mizan: A large Persian-English parallel corpus." In: *arXiv preprint arXiv:1801.02107*. DOI: 10.48550/ARXIV.1801.02107. URL: https://arxiv.org/abs/1801.02107.

Kastell Laura (2021). "Language Transfer in the Use of English Articles by Native Finnish and Swedish Speakers." Bachelor's Thesis.

Kingma Diederik P and Jimmy Ba (2014). "Adam: A method for stochastic optimization." In: *arXiv preprint arXiv:1412.6980*. DOI: 10.48550/ARXIV.1412.6980. URL: https://arxiv.org/abs/1412.6980.

Lado R. (1957). *Linguistics Across Cultures: Applied Linguistics for Language Teachers*. University of Michigan Press. URL: https://books.google.ca/books?id=ZzYGAQAAIAAJ.

Leacock Claudia et al. (2010). "Automated grammatical error detection for language learners." In: *Synthesis lectures on human language technologies* 3.1, pp. 1–134. DOI: https://doi.org/10.2200/S00275ED1V01Y201006HLT009.

Lee Lung-Hao et al. (Aug. 2014). "A Sentence Judgment System for Grammatical Error Detection." In: *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: System Demonstrations*. Dublin, Ireland: Dublin City University and Association for Computational Linguistics, pp. 67–70. URL: https://aclanthology.org/C14-2015.

Leidner Jochen L. and Vassilis Plachouras (Apr. 2017). "Ethical by Design: Ethics Best Practices for Natural Language Processing." In: *Proceedings of the First ACL Workshop on Ethics in Natural Language Processing*. Valencia, Spain: Association for Computational Linguistics, pp. 30–40. DOI: 10.18653/v1/W17-1604. URL: https://aclanthology.org/W17-1604.

Liu Xing (2011). "The not–so–humble "Chinese comma": Improving English CFL students' understanding of multi–clause sentences." In: *Proceedings of the 9th New York International Conference on Teaching Chinese*.

Marcus Mitchell P., Beatrice Santorini, and Mary Ann Marcinkiewicz (1993). "Building a Large Annotated Corpus of English: The Penn Treebank." In: *Computational Linguistics* 19.2, pp. 313–330. URL: https://aclanthology.org/J93-2004.

Mieskes Margot (Apr. 2017). "A Quantitative Study of Data in the NLP community." In: *Proceedings of the First ACL Workshop on Ethics in Natural Language Processing*. Valencia, Spain: Association for Computational Linguistics, pp. 23–29. DOI: 10.18653/v1/W17-1603. URL: https://aclanthology.org/W17-1603.

Mikolov Tomas et al. (2010). "Recurrent neural network based language model." In: *Interspeech*. Vol. 2. 3. Makuhari, pp. 1045–1048.

Monaikul Natawut and Barbara Di Eugenio (2020). "Detecting preposition errors to target interlingual errors in second language writing." In: *The Thirty-Third International Flairs Conference*. URL: https://par.nsf.gov/biblio/10191933.

Murphy Shirin (2003). "Second language transfer during third language acquisition." In: *Studies in Applied Linguistics and TESOL* 3.2. DOI: https://doi.org/10.7916/D8SF2VN8.

Nadejde Maria and Joel Tetreault (Nov. 2019). "Personalizing Grammatical Error Correction: Adaptation to Proficiency Level and L1." In: pp. 27–33. DOI: 10.18653/v1/D19-5504. URL: https://aclanthology.org/D19-5504.

Nicholls Diane (2003). "The Cambridge Learner Corpus: Error coding and analysis for lexicography and ELT." In: *Proceedings of the Corpus Linguistics 2003 conference*. Vol. 16, pp. 572–581.

Niu Lian (2020). "A review of the application of logistic regression in educational research: common issues, implications, and suggestions." In: *Educational Review* 72.1, pp. 41–67. DOI: 10.1080/00131911.2018.1483892. eprint: https://doi.org/10.1080/00131911.2018.1483892. URL: https://doi.org/10.1080/00131911.2018.1483892.

Nivre Joakim et al. (May 2016). "Universal Dependencies v1: A Multilingual Treebank Collection." In: *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*. Portorož, Slovenia: European Language Resources Association (ELRA), pp. 1659–1666. URL: https://aclanthology.org/L16-1262.

O'Malley J Michael and A Chamot (1990). "Strategies used by second language learners." In: *Learning Strategies in Second Language Acquisition*, pp. 114–150.

Peters Matthew E. et al. (July 2017). "Semi-supervised sequence tagging with bidirectional language models." In: *Proceedings of the 55th Annual Meeting*

of the Association for Computational Linguistics (Volume 1: Long Papers). Vancouver, Canada: Association for Computational Linguistics, pp. 1756–1765. DOI: 10.18653/v1/P17-1161. URL: https://aclanthology.org/P17-1161.

Raschka Sebastian (2018). "Model evaluation, model selection, and algorithm selection in machine learning." In: *arXiv preprint arXiv:1811.12808*. DOI: 10.48550/ARXIV.1811.12808. URL: https://arxiv.org/abs/1811.12808.

Richardson Stephen D. and Lisa C. Braden-Harder (1988). "The Experience of Developing a Large-Scale Natural Language Text Processing System: CRITIQUE." In: *Proceedings of the Second Conference on Applied Natural Language Processing*. ANLC '88. Austin, Texas: Association for Computational Linguistics, pp. 195–202. DOI: 10.3115/974235.974271. URL: https://doi.org/10.3115/974235.974271.

Robertson Daniel (2000). "Variability in the use of the English article system by Chinese learners of English." In: *Second language research* 16.2, pp. 135–172. DOI: 10.1191/026765800672262975. URL: https://doi.org/10.1191/026765800672262975.

Rozovskaya Alla, Dan Roth, and Vivek Srikumar (Apr. 2014). "Correcting Grammatical Verb Errors." In: *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*. Gothenburg, Sweden: Association for Computational Linguistics, pp. 358–367. DOI: 10.3115/v1/E14-1038. URL: https://aclanthology.org/E14-1038.

Selinker Larry (1969). "Language transfer." In: *General linguistics* 9.2, p. 67.

Serlin Ronald C, John Carr, and Leonard A Marascuilo (1982). "A measure of association for selected nonparametric procedures." In: *Psychological Bulletin* 92.3, p. 786. DOI: https://psycnet.apa.org/doi/10.1037/0033-2909.92.3.786.

Shute Valerie J (2008). "Focus on formative feedback." In: *Review of educational research* 78.1, pp. 153–189. DOI: https://doi.org/10.3102\%2F0034654307313795.

Swan Michael and Bernard Smith (2001). *A teachers' guide to interference and other problems*. 2nd ed. Cambridge Handbooks for Language Teachers. Cambridge University Press. DOI: 10.1017/CBO9780511667121.

Tsui Amy BM (2007). "Complexities of identity formation: A narrative inquiry of an EFL teacher." In: *TESOL quarterly* 41.4, pp. 657–680.

Van der Aalst Wil MP et al. (2010). "Process mining: a two-step approach to balance between underfitting and overfitting." In: *Software & Systems Modeling* 9.1, pp. 87–111. DOI: https://doi.org/10.1007/s10270-008-0106-z.

Wu Chung-Hsien et al. (2009). "Sentence correction incorporating relative position and parse template language models." In: *IEEE Transactions on Audio, Speech, and Language Processing* 18.6, pp. 1170–1181.

Xu Baoxun et al. (2012). "Hybrid Random Forests: Advantages of Mixed Trees in Classifying Text Data." In: *Advances in Knowledge Discovery and Data Mining*. Ed. by Pang-Ning Tan et al. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 147–158. ISBN: 978-3-642-30217-6.

Yannakoudakis Helen, Ted Briscoe, and Ben Medlock (June 2011). "A New Dataset and Method for Automatically Grading ESOL Texts." In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Portland, Oregon, USA: Association for Computational Linguistics, pp. 180–189. URL: https://aclanthology.org/P11-1019.

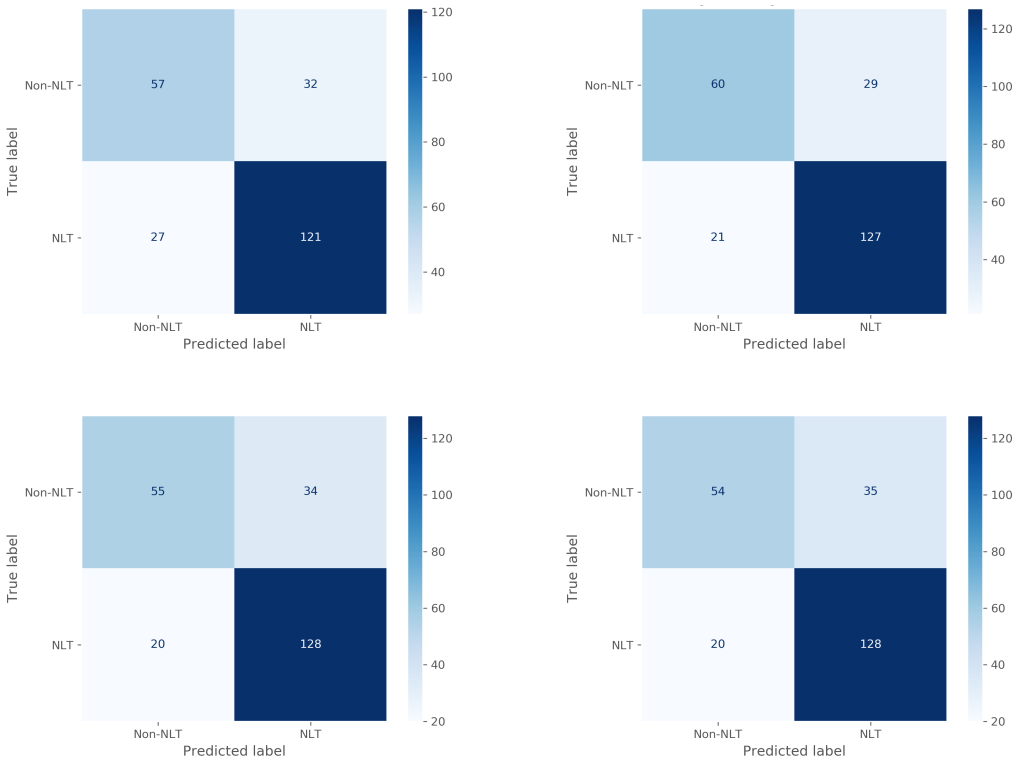
Appendix A

Non-theory-based Model Confusion Matrices

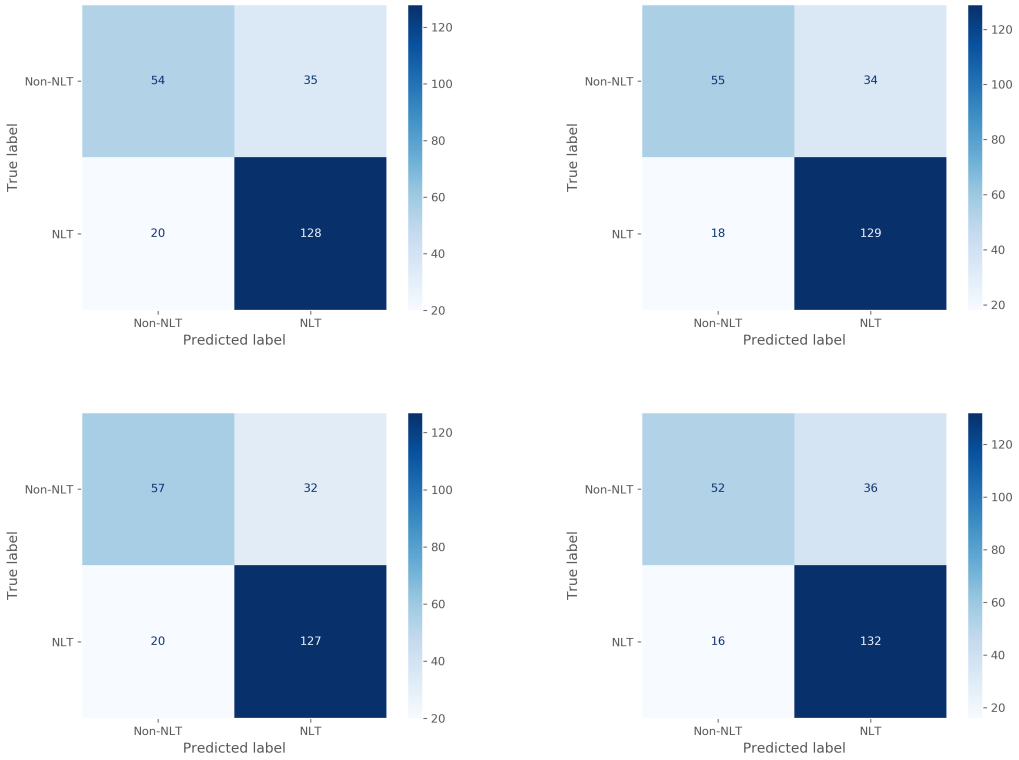
A.1 Chinese FCE dataset

A.1.1 Logistic Regression

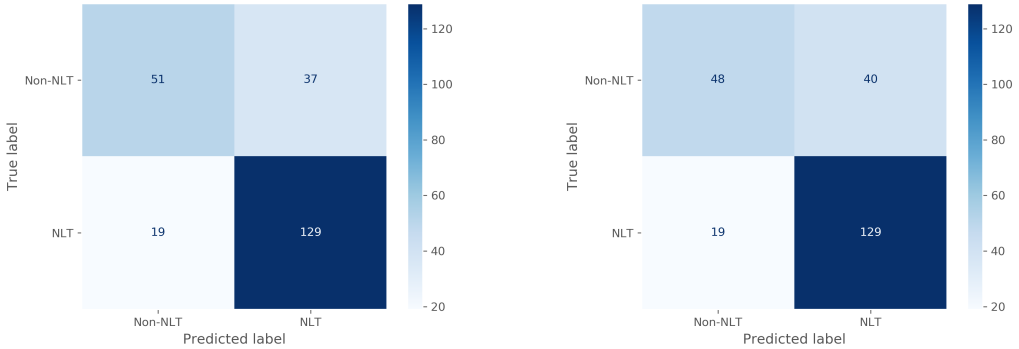
Confusion matrices of the first four folds (1-4) of the nested cross-validation process of the logistic regression model evaluated on the Chinese FCE dataset. Folds increment in a left to right direction.



Confusion matrices of the second four folds (5-8) of the nested cross-validation process of the logistic regression model evaluated on the Chinese FCE dataset.

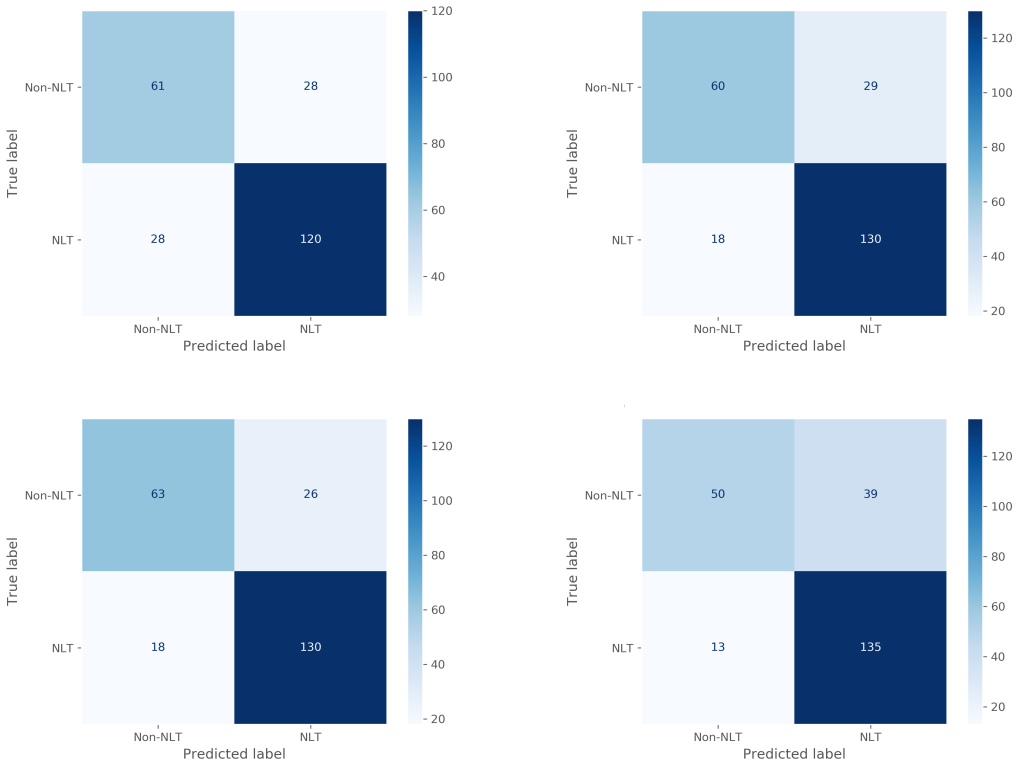


Confusion matrices of the last two folds (9, 10) of the nested cross-validation process of the logistic regression model evaluated on the Chinese FCE dataset.

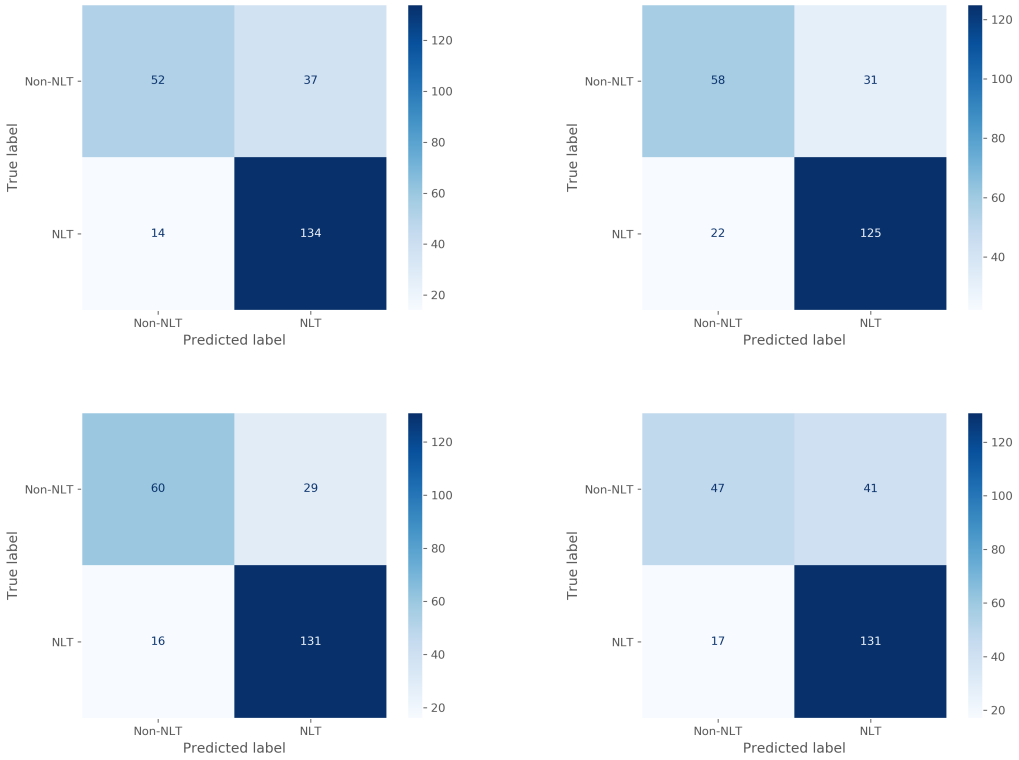


A.1.2 Random Forest

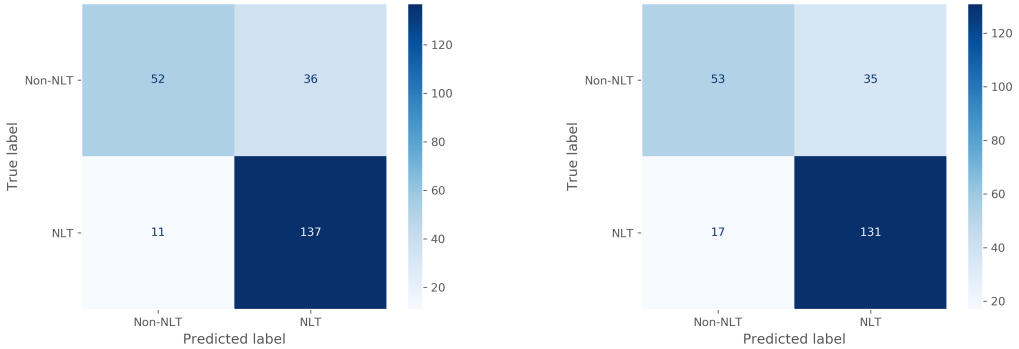
Confusion matrices of the first four folds (1-4) of the nested cross-validation process of the random forest model evaluated on the Chinese FCE dataset.



Confusion matrices of the second four folds (5-8) of the nested cross-validation process of the random forest model evaluated on the Chinese FCE dataset.



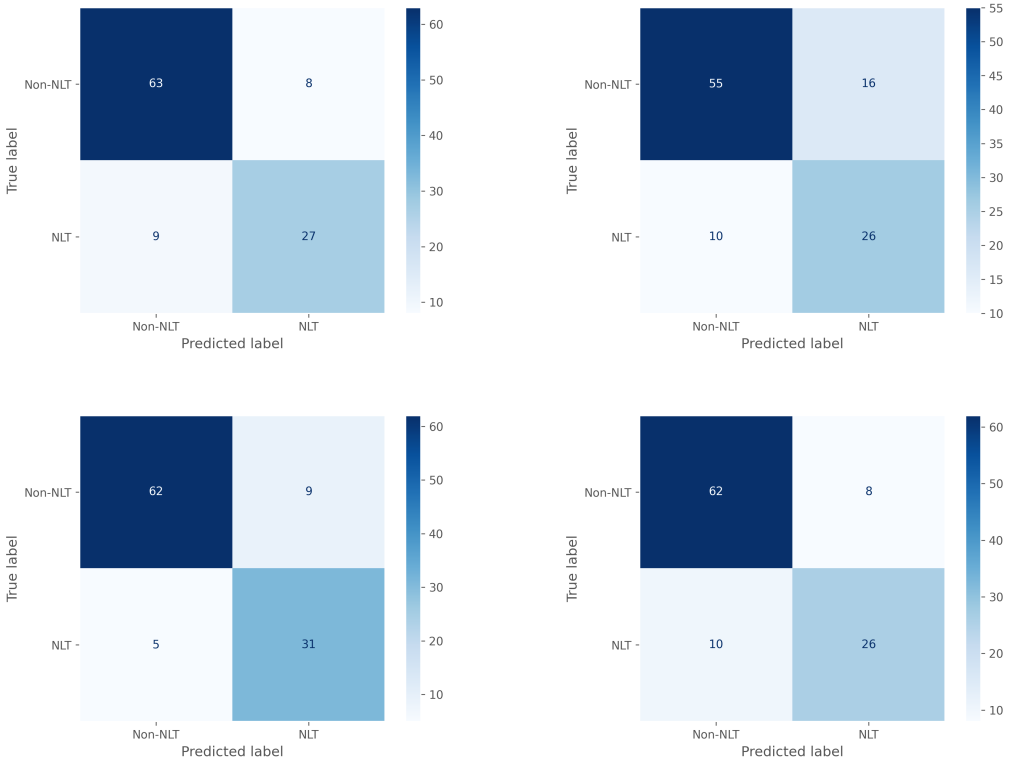
Confusion matrices of the last two folds (9, 10) of the nested cross-validation process of the random forest model evaluated on the Chinese FCE dataset.



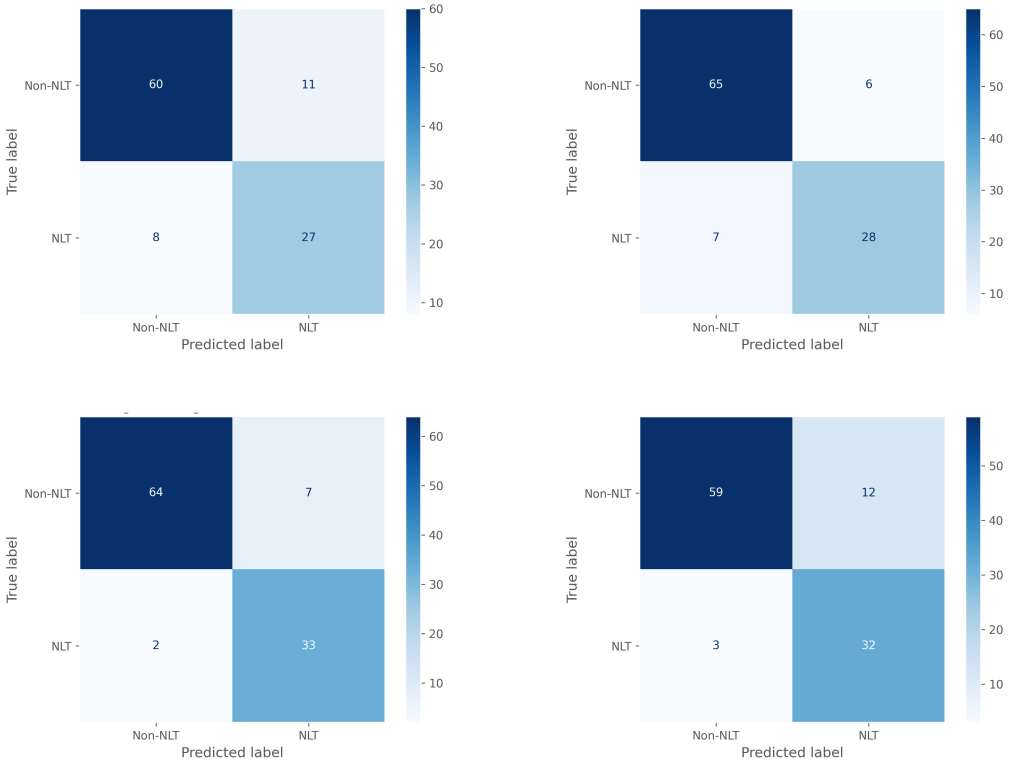
A.2 Farsi Lang-8 dataset

A.2.1 Logistic Regression

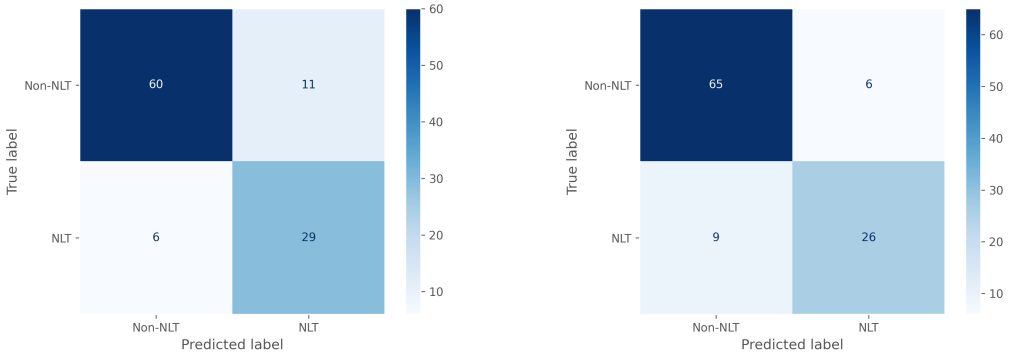
Confusion matrices of the first four folds (1-4) of the nested cross-validation process of the logistic regression model evaluated on the Farsi Lang-8 dataset.



Confusion matrices of the second four folds (5-8) of the nested cross-validation process of the logistic regression model evaluated on the Farsi Lang-8 dataset.

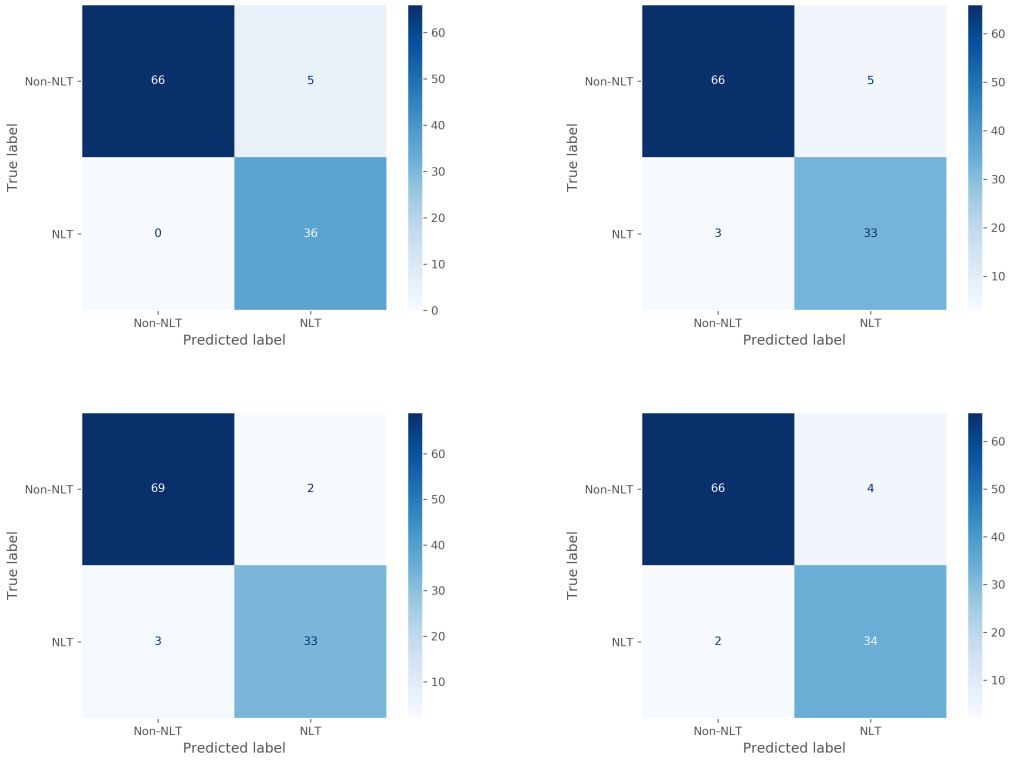


Confusion matrices of the last two folds (9, 10) of the nested cross-validation process of the logistic regression model evaluated on the Farsi Lang-8 dataset.

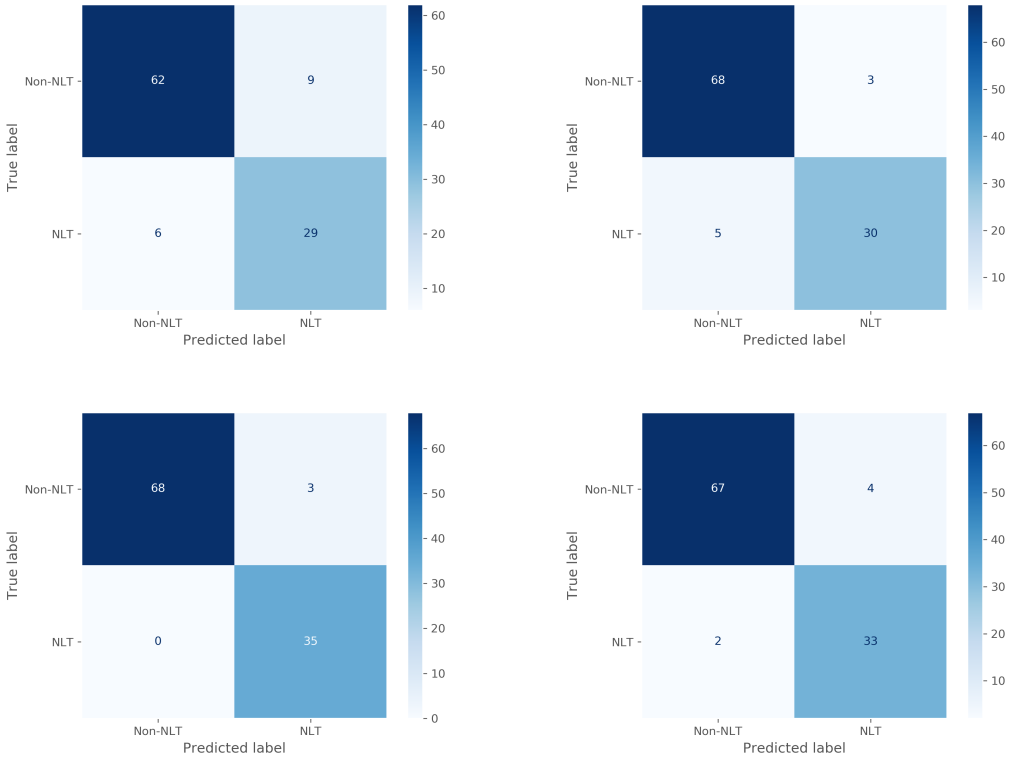


A.2.2 Random Forest

Confusion matrices of the first four folds (1-4) of the nested cross-validation process of the random forest model evaluated on the Farsi Lang-8 dataset.



Confusion matrices of the second four folds (5-8) of the nested cross-validation process of the random forest model evaluated on the Farsi Lang-8 dataset.



Confusion matrices of the last two folds (9, 10) of the nested cross-validation process of the random forest model evaluated on the Farsi Lang-8 dataset.