Generating Code-Switched Text from Monolingual Text with Dependency Tree

Bryan Gregorius
School of Science and Technology
Kwansei Gakuin University
Hyogo, Japan
contact@selubi.tech

Takeshi Okadome
School of Science and Technology
Kwansei Gakuin University
Hyogo, Japan
tokadome@acm.org

Abstract

Various methods have been proposed to generate code-switched texts. Many of these involve training neural networks and, in turn, require some (albeit small) amounts of code-switched texts or parallel corpora to train the model itself. In this paper, we propose a method to convert monolingual text into a bilingual code-switched sentence using a dependency parser and machine translator. We leverage the characteristics of the dependency tree to identify the switching point and then pass it to machine translation to generate the code-switched sentence. We then surveyed multilingual people of respective language pairs to review the generated sentences and categorize the result. We found that our method is capable of generating natural code-switched text for various language pairs with the same algorithm. Our method does not require training and thus does not require training data. Our implementation of the model uses off-the-shelf components. The implementation is also built with the possibility of using purpose-built components and rapid deployability in mind.

1 Introduction

Code-Switching (CS) is a phenomenon where a speaker alternates between two or more languages in a single conversation. This phenomenon is frequently observed in multilingual communities, with sentences alternating between a base language and one or more inserted languages. One of the reasons for doing CS is to clarify important information that cannot be explained in one language or code.

An example of this can be seen in Table 1. Here, while the English translation appears natural, the concept "内々定" (informal promise of employment) does not exist in English. Therefore, the word "offer" is a good substitute. However, for people who understand Japanese, "内々定" provides more context. In this example, the base language that provides the grammatical structure is English, and the inserted language is Japanese.

CS-related research is integral to Natural Language Processing (NLP) research, as it helps us understand how multilingual people use and understand languages. Most NLP-related corpora are an aggregation of scripts taken from books, movies, and other media. However, those media are primarily targeted at a particular demographic and, thus, mostly monolingual. This makes CS-related corpus scarce and CS corpus generation a research topic of interest. Furthermore, CS corpus generation is but a step in building CS language models. As such, there is a demand for rapidly deployable CS sentence generators.

In this paper, we propose a method to generate CS sentences from monolingual sentences. Our model is designed to work on various language pairs. Our model implementation is easily expandable to other language pairs and is made with the possibility of being used in tandem with custom components in mind. To our knowledge, we are the first to build a highly extensible code-switched generator that only needs monolingual inputs while also supporting the generation of multiple language pairs with the same algorithm.

2 Related Research

Research in CS is being done extensively. An example of a topic in this field is researching the model itself, such as using subword level aspects in addition to word level aspects to rep-

Table 1: Lost in translation code-switching sentence example

CS Sentence	My 内々定承諾期限 is in July.
English Translation	My offer acceptance deadline is in July.

resent CS data (Winata et al., 2019) and measuring the effectiveness of multilingual models, such as mBERT on CS tasks (Winata et al., 2021).

Since the nature of CS is a mix of two or more languages, it takes mastery in all the languages involved to do research and validation. Therefore, it is understandable that most of the work involves only a pair of two languages. Even then, it is hard for a reader that does not understand both languages to tell how well a model performs. There is also always a possibility that a good CS breakthrough might be left undetected because it is written in a language pair that is not well known. To alleviate these issues, there are several benchmarks to measure a model's performance on CS tasks. The LinCE Benchmark (Aguilar et al., 2020) is one example of it. However, the problem persists even with these benchmarks. Ultimately, only multilingual people who understand both languages can rate the models' performance from a human perspective.

Due to extensive research being done on modeling CS, great demand for CS corpora exists. There have been efforts to provide natural CS corpora both in text form, such as in Barik et al. (2019) and in speech form, such as in Nguyen and Bryant (2020). However, the number of CS corpora pales compared to even parallel or multilingual ones. This, in turn, increases the interest in research fields in CS corpora generation.

In regards to CS corpora generation, there has been an effort to implement findings of CS from the linguistics field, namely the equivalence constraint (EC) theory (Poplack, 1980) by Pratapa et al. (2018). Pratapa et al. used a constituency-based parse tree and parallel monolingual sentences (two monolingual sentences of the same meaning) to generate CS sentences. Although simple, this synthetic CS generation method has been proven to be helpful in training neural network models, such as neural networks that generate even higher quality CS texts (Tarunesh et al., 2021).

Pratapa et al. modeled their method on the Spanish-English language pair, which is linguistically close and has some parallel corpora. Our proposed model works on multiple language pairs, even on linguistically distant pairs such as Japanese-English (Chiswick and Miller, 2005). Our model does not need a parallel monolingual sentence to function and provides an alternative to Pratapa et al.'s proposed model.

3 Model and Implementation

3.1 Proposed Method

The dependency-based parse tree is one of two types of parse trees, the other being the constituency-based parsed tree used in (Pratapa et al., 2018). The dependency-based parse tree differs from constituency-based parsed by lacking phrasal categories, thus making the tree more straightforward. In an English constituency-based parse tree, the number of words usually equals the number of leaves. On the other hand, in English dependency-based parse trees, the number of words usually equals the number of vertices. There are several types of dependency-based parse trees, but we will focus on the syntactic dependency tree (we will refer to this as just dependency tree from hereon). The dependency tree is an ordered tree; as such, flattening the tree can be defined as concatenating the vertices of the tree in the order of the original sentence. Figure 1 is an example of a dependency tree.

Our proposed model works by first getting the dependency tree of a monolingual sentence (base sentence or [X]-base from hereon) by passing said sentence into a dependency parser. We then determine the switching point from the dependency tree and translate the switching point in place with a machine translation model. We define the switching point as the part of the sentence that will later be passed to a machine translator and gets trans-

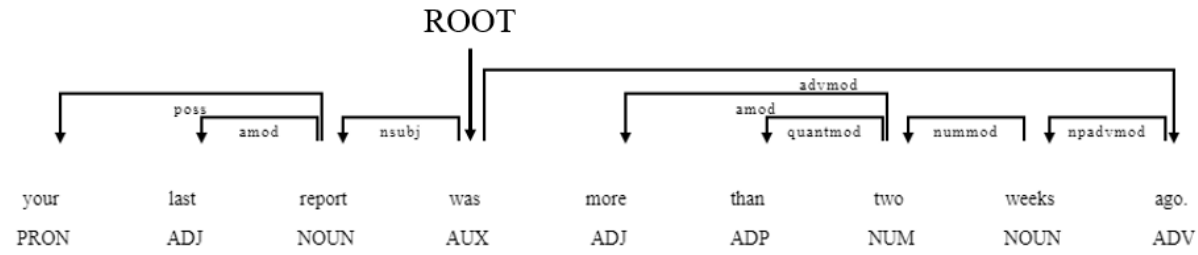

Figure 1: English dependency tree example

lated into the inserted language. We assume access to both a dependency parser and a machine translator. Therefore, to create a [X]-[Y] Code-Switched sentence, we need a dependency parser that can output a dependency tree of [X] and a machine translator that can translate from [X] to [Y]. Here, [X] and [Y] are languages represented by the ISO 639-1 code. For example, EN-JA means a code-switched sentence generated from a monolingual English base sentence (EN-Base) and the switching point translated into Japanese.

3.2 Determining the Switching Point

Given a dependency tree T with root vertex r, let V_i be the set of all vertex at depth i of T. We define the root as the only depth 0 vertex, as such $V_0 = \{r\}$. Here, we define $|v|$ as the number of vertex of a subtree of T with vertex v as the root, given $v \neq r$ and v is a vertex of T. The switching point is the flattened subtree of T with any vertex $s \in S$ as its root. S is given by

$$S = \begin{cases} \text{argmax}_{V_1} f, & \text{if} \max_{V_1} f > 1 \\ \{v \mid v \text{ is noun}, v \in V_1\}, & \text{if} \max_{V_1} f = 1 \end{cases}$$

where

$$f(v) = |v|.$$

In Figure 1, $V_1 = \{\text{report}, \text{ago}\}$ and $f(\text{report}) = 3, f(\text{ago}) = 5$. As such, $S = \{\text{ago}\}$ and the switching point is the flattened subtree with the root "ago", which is "more than two weeks ago."

In other words, we propose the switching point to be the flattened largest subtree with the dependency tree root's direct children as the subtree's root. By choosing the largest subtree, we maximize the chance that the switching point is contextually independent enough to produce a good translation, especially since the machine translator is an independent component and cannot access the whole sentence to infer additional context. In most cases, the largest subtree size should be at least 2. However, in simple sentences such as "I eat meat," we choose to translate nouns only as it has the highest likelihood of being contextually independent.

3.3 Implementation

We implemented our solution with Spacy (Honnibal et al., 2020) as the dependency parser and both DeepL and Google Cloud Translation AI as the machine translators. We use Google Cloud Translation AI for language pairs not supported by DeepL. In our implementation, if there are multiple vertices in S, we choose the leftmost vertex as the switching point root. The repository of our implementation can be found at (Gregorius, 2022).

This implementation is made with rapid deployability in mind; hence adding a language pair is relatively simple. The implementation also features a demo that generates EN-JA and JA-EN sentences from the JESC corpus's (Pryzant et al., 2018) test data using DeepL. There are 2000 lines of English and Japanese sentences in that data. Generating EN-JA sentences took 13 minutes and 24 seconds and JA-EN 15 minutes and 27 seconds, which results in average speeds of 2.49 lines/second and 2.16 lines/second, respectively. For more information about the implementation, we recommend visiting the repository itself.

4 Results

We generated sentences using our implementation and asked multilingual people of respective language pairs for review (the reviewers from hereon). We conducted the review by

asking if the sentence was natural and asking for an explanation of the unnatural sentences. From the response, we observed that the code-switched sentence could be categorized into four categories: natural, incorrect grammar with correct context, context changed but natural grammar usage, and incomprehensible. Moving forward, the notation T_n^m to refer the n-th entry in Table m. For example, T_1^2 refers to the pair of EN-Base sentence of "your last report was more than two weeks ago." and its EN-JA generated sentence of "your last report was 二週間以上前."

4.1 Results : Natural Code-Switched Texts

All entries in Table 2 are deemed natural by the reviewers. These texts require no additional grammar correction and lose no context during translation. In testing, we observe more natural results like this, but we will only show one sentence for each directional language pair due to space limitations.

4.2 Results: Incorrect Grammar with Correct Context

All the entries in Table 3 need grammatical correction to varying degrees but have correct context and are understandable.

T_1^3 and T_2^3 require preposition to be added. In T_2^3, "First" should be "At first" for it to be natural. T_3^3 can sound more natural by adding a verb at the end. These required changes are relatively minor.

T_4^3 and T_5^3 have double subjects in its code-switched sentences. In T_4^3 the model generated "저는" and "I" which both mean "I" and in T_5^3 it generated "저는" and "我" which also both means "I".

4.3 Results: Context Changed but Natural Grammar

All the code-switched text entries in Table 4 are grammatically correct. However, compared to the base texts, these texts lost or changed the context from the original.

T_1^4, T_2^4, and T_3^4 context changed due to vocabulary choice. In T_1^4 base text, "忠告" (advice, warning) gets translated to "建议" (suggestion) even though the word "忠告" exists in Chinese. T_2^4 ZH-Base's "商家" (businessman, merchant) gets translated to "加盟店" (member store [of a store association]) where it should be "商人" (businessman, merchant). There are also better word choices to explain "弱者と危機に瀕している" (socially vulnerable and at-risk) than "संवेदनशील और जोखिम वाल" for T_3^4.

T_4^4, T_5^4, and T_6^4 lost context implication during translation. T_4^4 and T_5^4 base sentence translates implies that the writer has not been able to buy a ticket despite waiting for a long time. This context got lost in both sentences. A proper substitute for T_4^4 inserted language part would be "But I haven't been able to buy a ticket yet" and T_5^4 "でも、まだチケット取れてないんです". The Thai part of T_6^4 translates to "might be yours," but by the wording, its closer to "(things) might be yours" compared to the JA-Base sentence which translates to "you may think like that (but I don't)."

4.4 Results: Incomprehensible Code-Switched Texts

All code-switched text in Table 5 is incomprehensible. The reviewers cannot understand the meaning without looking at the base sentence.

The machine translator failed to detect the name "tup" In T_1^5 and tries to translate it, resulting in an incomprehensible sentence. Also, the second sentence's "今がそのとき" translates to "it's now the time," contains an implied subject. Thus the sentence also has a double subject just like T_4^3 and T_5^3. T_2^5's "社会的弱者と危機に瀕しているグループ" (socially vulnerable and at-risk groups) translates to "กลุ่มเสี่ยงและกลุ่มเสี่ยง" (risk group and risk group) which is incomprehensible. In T_3^5, the translator failed to translate "商家的诚信," and the generated Korean is incomprehensible.

5 Discussion

Our model heavily relies on a dependency parser and machine translator. As a result, any errors in those components reflect directly on the performance of our model. The machine translator may output different translation results even with the same machine translator and input. For example, T_{13}^2's JA-ID is a natural result but sometimes the DeepL translator outputs "kelompok rentan dan berisiko"

Table 2: Natural Generated Code-Switched Texts

1	EN-Base	your last report was more than two weeks ago.
	EN-JA	your last report was 二週間以上前.
2	EN-Base	you are a good soldier, tup. it's time to go now.
	EN-ZH	this symbol is you are 一个好的士兵，Tup . it 's 现在是时候走了
3	JA-Base	私の忠告がほとんど重要でないというのか?
	JA-EN	My advice ほとんど重要でないというのか?
4	ZH-Base	商家的诚信和口碑有着密不可分的联系。
	ZH-EN	Merchant's integrity and reputation 有着密不可分的联系。
5	JA-Base	この記号は 昔の 地下鉄トンネル網の地図よ
	JA-ZH	この記号は古老的地下隧道网络地図よ
6	JA-Base	しかし社会的弱者と危機に瀕しているグループに力点を置いています
	JA-KO	しかし사회적 약자와 위기에 처한 그룹力点を置いています
7	KO-Base	저는 어제 약국에 가서 약을 많이 샀어요.
	KO-JA	저는昨日薬局に行って약을많이샀어요.
8	EN-Base	so you quit school and quit looking for work and decided to become a chef.
	EN-TH	so you quit school and quit looking for work and ตัดสินใจเป็นเชฟ
9	EN-Base	you are a good soldier, tup. it's time to go now.
	EN-HI	you are एक अच्छा सैनिक , tup. it's अब जाने का समय .
10	JA-Base	あなたにはそうかもしれないが 私そう思わない
	JA-HI	यह आपके लिए हो सकता है 私そう思わない
11	EN-Base	you are a good soldier, tup. it's time to go now.
	EN-ID	you are prajurit yang baik, tup. it's waktu untuk pergi sekarang.
12	JA-Base	しかし社会的弱者と危機に瀕しているグループに力点を置いています
	JA-ID	しかし untuk kelompok rentan dan berisiko 力点を置いています

Table 3: Incorrect Grammar Generated Code-Switched Texts

1	EN-Base	you are a good soldier, tup. it's time to go now.
	EN-TH	you are ทหารที่ดี tup. it's เวลาไปตอนนี้ .
2	JA-Base	最初はうまく いかなかったんだよ
	JA-EN	First うまくいかなかったんだよ
3	EN-Base	so you quit school and quit looking for work and decided to become a chef
	EN-KO	so you quit school and quit looking for work and 요리사가 되기로 결심.
4	KO-Base	저는 어제 약국에 가서 약을 많이 샀어요.
	KO-EN	저는 I went to the pharmacy yesterday 약을많이샀어요.
5	KO-Base	저는 어제 약국에 가서 약을 많이 샀어요.
	KO-ZH	저는昨天去了药房약을많이샀어요.

Table 4: Context Changed Natural Generated Code-Switched Texts

1	JA-Base	私の忠告がほとんど重要でないというのか?
	JA-ZH	我的建议ほとんど重要でないというのか?
2	ZH-Base	商家的诚信和口碑有着密不可分的联系
	ZH-JA	加盟店の誠実さ、評判有着密不可分的联系。
3	JA-Base	しかし社会的弱者と危機に瀕しているグループに力点を置いています
	JA-HI	しかし संवेदनशील और जोखिम वाले समूह 力点を置いています
4	ZH-Base	尽管我早晨六点到了售票处，但是我还没买到票
	ZH-EN	尽管我早晨六点到了售票处，But I haven't bought a ticket yet
5	ZH-Base	尽管我早晨六点到了售票处，但是我还没买到票
	ZH-JA	尽管我早晨六点到了售票处，でも、まだチケット取ってないんです
6	JA-Base	あなたにはそうかもしれないが 私そう思わない
	JA-TH	อาจเป็นของคุณ 私そう思わない

Table 5: Incomprehensible Code-Switched Texts

1	EN-Base	you are a good soldier, tup. it’s time to go now.
	EN-JA	you are けいぐんたいとう . it 's 今がその時 .
2	JA-Base	しかし社会的弱者と危機に瀕しているグループに力点を置いています
	JA-TH	しかし กลุ่มเสี่ยงและกลุ่มเสี่ยง 力点を置いています
3	ZH-Base	商家的诚信和口碑有着密不可分的联系。
	ZH-KO	비즈니스 무결성 및 평판有着密不可分的联系。

Table 6: Code-Switching Generation Comparison Between DeepL and Google Cloud Translation AI

EN-Base	you are a good soldier, tup. it’s time to go now.
EN-JA (DeepL)	you are けいぐんたいとう . it 's 今がその時 .
EN-JA (Google)	you are 良い兵士、タップ . it ’ s 今行く時間 .

(vulnerable and at-risk groups) and truncates "untuk" (for) which gets categorized as incorrect grammar with correct context instead of natural (due to needing preposition).

If we change the machine translator, the result is even more apparent. Table 6 EN-Base and EN-JA (DeepL) is the same entry as T_1^5, which is incomprehensible. In comparison, the Google Cloud Translation AI managed to translate it flawlessly, even localizing the name "tup" into its Japanese version, "タップ." This turns it from being categorized as incomprehensible to being categorized as natural translation, or at worst incorrect grammar with correct context due to a double subject in the second sentence.

When reviewing the code-switched texts, we observe that some language pairs tended to produce more natural texts. EN-ZH and JA-KO are examples of this. Meanwhile, problems that do not occur in other pairs may appear in some language pairs. A good example is the double subject problem we discussed in 4.2, which occurs due to Japanese and Korean having implied subjects built into the language.

6 Conclusion

In this paper, we proposed a model to generate CS text using a dependency tree. We also showed that, albeit heuristic in nature, this model could produce natural CS text in various language pairs, even pairs of languages distant from each other. Our model only needs a single monolingual sentence as the input. Therefore, it is an excellent alternative to existing models using parallel monolingual inputs. Our implementation focuses heavily on rapid deployability and modularity with custom components. We hope that it will be integrated with custom-built components to generate even higher-quality CS texts in the future.

It is impossible to test all combinations of language in one paper. Therefore we would like to invite future readers and researchers to try this model on various language pairs. We also showed that the machine translator significantly affects our model performance. Fortunately, our model is relatively straightforward to implement, and we are excited to see what happens if we integrate it with a custom-built machine translator and dependency parsers. For example, a machine translator that analyzes the base sentences and prevents double subjects for Japanese and Korean may have great potential. Our implementation translates only a single part of the dependency tree to an inserted language, which results in a bilingual code-switched text. Expanding this idea, there is potential for translating multiple parts of the tree in different languages resulting in trilingual or even multilingual code-switched sentences. We hope our research provides progress in understanding code-switching, and we are excited to see future developments in this field.

References

Gustavo Aguilar, Sudipta Kar, and Thamar Solorio. 2020. LinCE: A Centralized Benchmark for Linguistic Code-switching Evaluation. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 1803–1813, Marseille, France. European Language Resources Association.

Anab Maulana Barik, Rahmad Mahendra, and Mirna Adriani. 2019. Normalization of Indonesian-English code-mixed Twitter data. In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, pages 417–424, Hong Kong, China. Association for Computational Linguistics.

Barry R. Chiswick and Paul W. Miller. 2005. Linguistic distance: A quantitative measure of the distance between english and other languages. *Journal of Multilingual and Multicultural Development*, 26(1):1–11.

Bryan Gregorius. Selubi/csify: Csify v1.0.6 [online]. 2022.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength Natural Language Processing in Python.

Li Nguyen and Christopher Bryant. 2020. CanVEC - the canberra Vietnamese-English code-switching natural speech corpus. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4121–4129, Marseille, France. European Language Resources Association.

Shana Poplack. 1980. Sometimes i' ll start a sentence in spanish y y termino en español: toward a typology of code-switching1. *Linguistics*, 18(7-8):581–618.

Adithya Pratapa, Gayatri Bhat, Monojit Choudhury, Sunayana Sitaram, Sandipan Dandapat, and Kalika Bali. 2018. Language modeling for code-mixing: The role of linguistic theory based synthetic data. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1543–1553, Melbourne, Australia. Association for Computational Linguistics.

Reid Pryzant, Youngjoo Chung, Dan Jurafsky, and Denny Britz. 2018. JESC: Japanese-English subtitle corpus. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Ishan Tarunesh, Syamantak Kumar, and Preethi Jyothi. 2021. From machine translation to code-switching: Generating high-quality code-switched text. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3154–3169, Online. Association for Computational Linguistics.

Genta Indra Winata, Samuel Cahyawijaya, Zihan Liu, Zhaojiang Lin, Andrea Madotto, and Pascale Fung. 2021. Are multilingual models effective in code-switching? In *Proceedings of the Fifth Workshop on Computational Approaches to Linguistic Code-Switching*, pages 142–153, Online. Association for Computational Linguistics.

Genta Indra Winata, Zhaojiang Lin, Jamin Shin, Zihan Liu, and Pascale Fung. 2019. Hierarchical meta-embeddings for code-switching named entity recognition. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3532–3538, Hong Kong, China. Association for Computational Linguistics.