Multilingual Language Processing From Bytes

BBPE를 더 잘 이해해보기 위해 보게 된 논문이기에, 이번 게시물에서는 byte-level에서의 input representation처리와 관련해서만 다루고, 나머지는 생략하도록 하겠다.

원문 링크는 다음과 같다

Multilingual Language Processing From Bytes

Introduction

본 연구는 기본적으로 LSTM 기반의 Seq2Seq model을 기반으로 하는데, 이때 sequence의 단어를 읽는 것이 아닌 sequence의 byte를 읽고, 각각의 단어에 관해 label을 생성하는 것이 아닌 [start, length, label]형태의 triplet을 생성하는 것이 본 논문에서 제시하는 연구의 차별점이다. 자료와 함께 설명해보도록 하겠다.

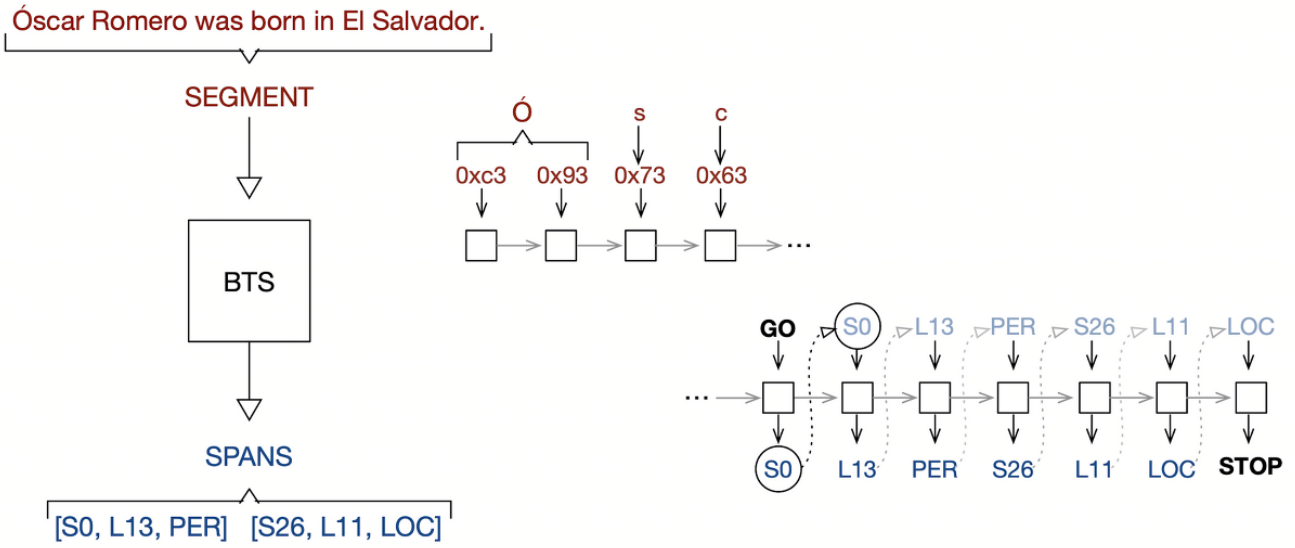
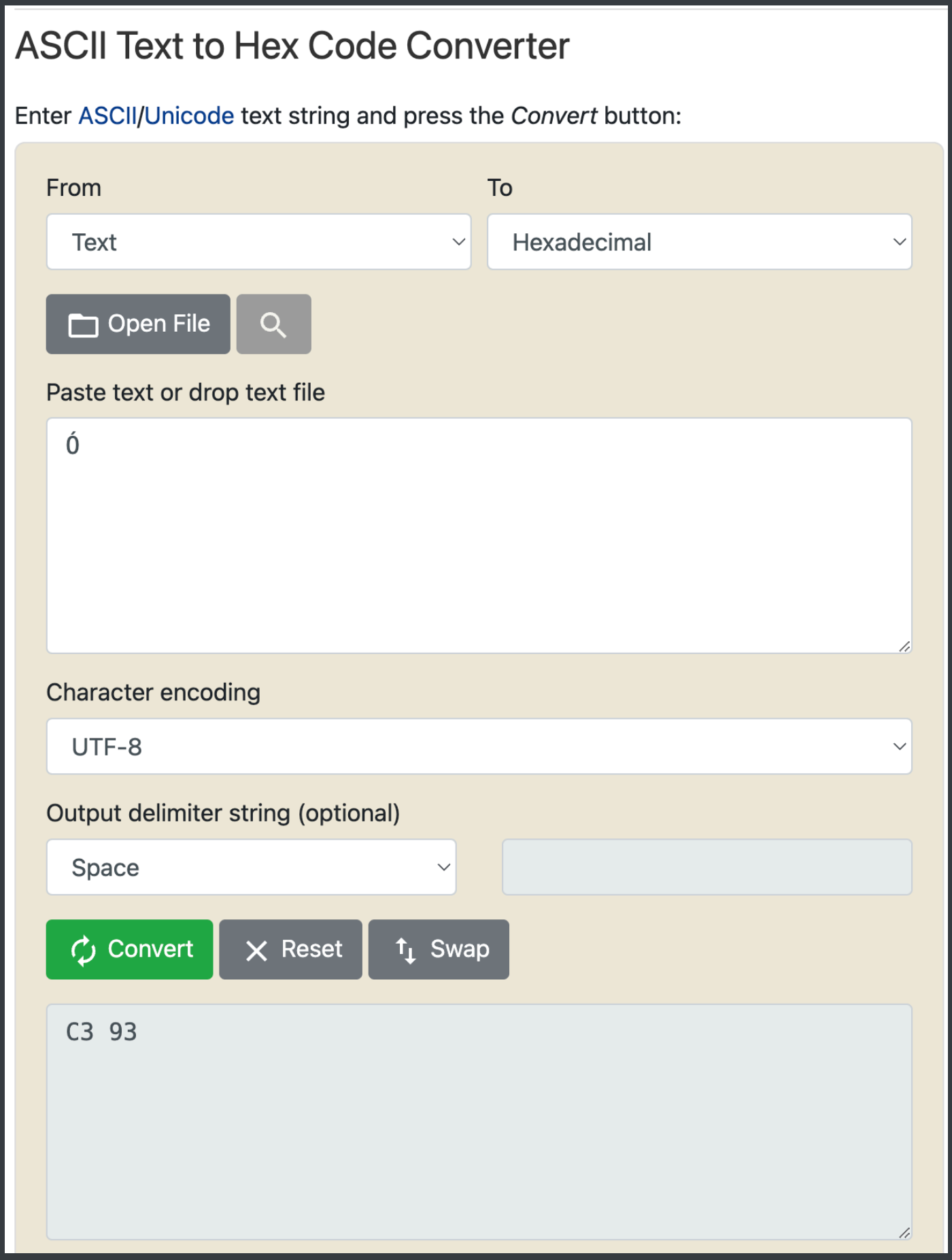


Figure 1: A diagram showing the way the Byte-to-Span (BTS) model converts an input text segment to a sequence of span annotations. The model reads the input segment one byte at a time (this can involve multibyte unicode characters), then a special Generate Output (GO) symbol, then produces the argmax output of a softmax over all possible start positions, lengths, and labels (as well as STOP, signifying no additional outputs). The prediction from the previous time step is fed as an input to the next time step.

우선 문장이 어떻게 segmentation되는지 살펴보자.

Ó의 경우, UTF-8 인코딩을 통해 hex string으로 변환하면 다음과 같은 결과가 나온다.



위 예시처럼 16진수의 C3, 93이 나오는 것을 확인할 수 있는데, 이때 이 각각의 요소들이 model에 input으로 들어가는 것이다.

즉, 문장을 이루고 있는 각각의 단어가 input으로 들어가는 것이 아닌, 단어를 이루고 있는 각각의 문자의 byte로 입력된다.

이후 [S0, L13, PER]과 [S26, L11, LOC]라는 triplet이 보일 것이다. 이것이 위에서 언급한 [start, length, label]형태의 triplet이다.

[S0, L13, PER]의 경우에는 0번째 위치에서 시작하여 13의 길이를 가지는 요소는 PER(인명)이라는 것이고, [S26, L11, LOC]는 26번째 위치에서 시작하여 11의 길이를 가지고 있는 요소는 LOC(지명)이라는 의미이다.

결과적으로, 이러한 방법은 model로 하여금 단어의 component와 label이 어떻게 상호작용하는지를 학습하게 한다.

이러한 decomposed 된 input과 output은 다음과 같은 장점을 가지고 있다.

- word-level input에 비해 vocabulary size가 작아져서 model이 compact해진다.
- 이러한 Unicode는 보편적인 언어이기 때문에, 한번에 여러 언어를 분석할 수 있는 model 생성 가능하다.

저자들은 실제로 이와 같은 기법을 통해 POS-tagging과 NER task에서 SOTA의 성능과 비슷하거나 그를 뛰어넘는 성능을 달성했다고 한다.

Model

본 연구에서의 model은 machine translation을 위해 사용되었던 Seq2Seq model에 기반한다고 한다.

이에 관해서는 일전에 Seq2Seq에 관해 정리해놓은 게시물 링크를 걸어놓고 생략하도록 하겠다.

시퀀스-투-시퀀스(Sequence-to-Sequence, seq2seq)란? - 기본 구조편

Vocabulary

기존의 model과 본 연구에서 제안하는 model이 다른 점이 바로 이 부분이다.

본 연구의 input의 집합은 256개의 가능한 byte와,(1byte = 8bit이기 때문에 2^8개의 가능한 경우의 수가 나오기 때문) special Generate Output(GO) symbol, special DROP symbol로 구성되어있다.

Output의 집합은 가능한 span start point(0번째 byte부터 k번째 byte), 가능한 span length(0부터 k), 그리고 span에 대한 label (NER task를 예로 들면 위에서 잠깐 볼 수 있었던 PER, LOC, ORG 등이 있다), 마지막으로 special STOP symbol로 구성되어있다.

온전한 span annotation은 start, length, label 3가지로 이루어져 있지만, 위의 자료에서 봤던 것처럼 model은 이 세가지를 따로따로 생산해낸다. 이러한 방식은 vocabulary size를 적게 유지해주고, 세 가지에 대한 cross-product space를 사용하는 것보다 더 나은 성능과 빠른 수렴 속도를 제공한다고 한다.

이후로는 Seq2Seq과 동일하게 학습 및 추론이 진행되는데, 저자들은 training data에 항상 이러한 triple이 고정된 순서로 존재하기 때문에 기형적이거나 불완전한 span은 거의 나타나지 않는다고 말한다.