



# 特許情報に関する言語生成モデルを 活用した知的財産創造手法の開発

Development of Intellectual Property Creation Method  
Using Language Generation Model on Patent Information

Shigeaki Onoda

Graduate School of Information Engineering, Toyama Prefectural University  
t855005@st.pu-toyama.ac.jp

Wednesday., 8 8, 2019,  
Toyama Prefectural Univ.



# 行ったこと

## 入力用データの作成

- 山元君の研究範囲との相談
- モデルの定義部分の修正
- おまけ（市場調査？）



## 山元君との研究範囲

山元くんの PC に小野田のクロール環境を入れた山元くんがテキストマイニングの練習が済み次第、クロールしていく予定を立てた  
(該当部分は山元君に対してできる限りサポートする)  
クロールより先どうするかはまだ未確定である現時点では 2 案  
あった

- ① 小野田の国際会議論文の引き継ぎ
- ② 先生が提示した 2 つの単語を掛け合わせる研究



# モデルの定義の変更

モデルのベクトルの次元数の定義にて埋め込み層の定義を以下のように変更した

1

```
self.embedding = nn.Embedding(vocab_size, hidden_size)  
* vocab_size : 全単語種数, hidden_size : 隠れ層の数
```

としていたが、word2vec を用いるタイプ

2

```
self.word_embeddings = nn.Embedding(vocab_size, embedding_dim)  
とする必要があった embedding_dim : word2vec の次元数
```



# 単語入力シーケンスの変更

文字列はそのままモデルへ入力できないため数値列に変換する必要があるその手順でのベストプラクティスがあったのでそちらを採用した

## 現在

word2vec を用いず文字列を id で表す場合

"hello world bye" => [2,3,4]

文字列 => id の配列 => モデルへ入力

## 小野田の想定

word2vec を用いた変換では当初以下の手順で行っていた

"hello world bye" => hello : [0.9, 1.2, 1.6]

このように単語一個ずつを辞書引きのように変換してた

文字列 => 単語から対応するベクトル検索 => 単語表現ベクトル  
=> モデルへ入力



## ベストプラクティス

テキスト処理の研究者やエンジニアの間では以下の方法で行うのが一般的だった

文字列 => id の配列 => id から対応するベクトルを充てがう =  
> 単語表現ベクトル => モデルへ入力

id : ベクトル辞書作成の手間がかかるが、これにより検索をせず  
 $O(1)$  で対応するベクトルを見つけることができる

## 例

例 good: 0 => [0.34, 0.56]

bye: 1 => [0.34, 0.47]

day: 2 => [0.98, 0.67]

morning: 3 => [0.43, 0.67]

all: 4 => [0.96, 0.76]

hello: 68 => [0.12, 0.34]

world: 50 => [0.28, 0.96]



## おまけ (市場調査)

### AI ブレストパーク

自分と山元君の研究に関係ありそうなサービスを見つけてこの場で紹介させていただきます。

TIS INTEC 株式会社が作成するアイデア支援ツール

<https://www.ai-b-spark.com/>



Figure: 1: AI ブレストパークのイメージ