音素を用いた対話システムを目指した日本語単語埋め込みの試み

Attempt to embed Japanese words aiming at a dialogue system using phonemes

松原伊吹

Ibuki Matsubara

広島市立大学　言語音声メディア工学研究室

Speech and Language Research Laboratory, Hiroshima City University

概要　本研究では，End2End 対話システム実現に向け，音律付き音素による単語ベクトルの生成を行った．Wikipedia dataset で学習をした FastText を用い，JWSAN データセットでの単語類似度評価でかな漢字のモデルと音律付き音素モデルとの比較を実施した．

1　はじめに

現在対話システムにおいて GPT[1]や T5[2]などの Transformer モデルが主流になってきている．日本語の Transformer モデルにおいてかな漢字を使用するのが一般的である．しかし音声認識の julius[3]では音声からかな漢字に直接変換するのではなく一旦音素に変換してから辞書を使いかな漢字へ変換する．また単語認識より音素認識の方が精度が優れている．音声合成でもかな漢字から一旦音素に変換を行い音声に変換するのが主流である．

かな漢字でなく音素での応答生成が可能であれば，かな漢字と音素との変換を行う辞書が要らず，また辞書に無い未知語にもより強固なのではと考えた．そこで音素単位で対話を行う GPT を作成した．実際に対話した結果を図 1 に示す．

音素では「橋」と「箸」などの同じ読みだがイントネーションが違う単語を区別する事ができない．そこでピッチの上がり下がり等が含まれている音声合成で用いられる音律付き音素を使用し実験を行なった．結果の一部を画像 1 に示す．

応答では「そうだね」や「〜してる」などの動詞や接続詞はできているが，名詞や形容詞などが少ない印象を受け音素で単語理解が出来ているのか疑問に感じた．そこで本論文では音素での単語認識が可能なのか，音素での単語ベクトル生成を行い検証を行う．

2　関連研究

Miikka らはフィンランド語，トルコ語，スペイン語において音素ベースで lstm を用い単語ベクトルの作成を行なった[4]．Zijun Sun らは中国語 BERT の tokenizer に pinyin 情報を追加することで精度向上を図った[5]．佐伯らは応答選択モデルに音素情報を付与することで認識誤りに強いシステムの構築を行なった[6]．

3　実験手法

実験には学習が容易な FastText[7]を使用した．まず FastText の学習を行い，単語ベクトル生成が可能な状態にする．次に評価実験ではある 2 つの単語同士の類似度を計算し，計算した類似

```
input: ^ k a [ d a i # d a [ r u ] i n a a
output: s o [ o _ d a ] k a r a _ s o [ o _ d a ] k a r a _ s o [ o y u u # n o [ o # k a [ N g a ]
 e t e _ d a ] k a r a _ s o [ o y u u # n o [ w a # k a [ N g a ] e r u # y o ] o n i # sh I [ t e
 ] r u
input: ^ k o [ N b a N w a
output: k o [ N b a N w a _ s o [ o d a n e _ k o [ N b a N w a _ s a [ i k i N # a [ r e d a y o n
 e _ a [ n o _ n a ] n i d a cl k e _ a [ n o _ n a ] n i d a cl k e _ a [ n o _ n a ] n i d a cl k
 e _ a [ n o _ n a ] n i d a cl k e _ a [ n o _ n a ] n i d a cl k e _ a [ n o _ n a ] n i d a cl k
```

図 1. GPT 応答例

度と人が評価した類似度との相関係数を指標として，単語ベクトルが正しく作成できているのか検証を行う．

3.1　モデル作成

学習の手法としてskipgramを使用し学習を行う．パラメータはベクトルの次元数を 300 とし，subword Ngram を様々な値を使い検証した．それ以外のパラメータはデフォルトで行なう．

学習データセットは Wikipedia が公開している Wikipedia コーパスの 2022 年 6 月時点での latest を使用した．学習データセット作成の流れを図 3 に示す．Wikipedia データセットに対し Mecab の NEologd を使用し単語分割を行ったデータで，かな漢字モデルの学習を行う．音律付き音素モデルは上記同様単語分割をした後，openjtalk のフルコンテキストラベル変換ライブラリを使用し，各単語を音律付き音素に変換した物を用いる．その際「kw」や「sh」などの 2 文字で表される音素は適当な記号に変更し 1 文字で表すように変換を行った．

3.2　評価手法

検証では日本語単語類似度データセット (JWSAN)[8]を使用する．データセットは図 4 のようになっており主に名詞，動詞，形容詞で構成されている．

評価方法として図 5 に示すように word1, word2 それぞれを FastText でベクトルに変換し，ベクトル同士のコサイン類似度を作成する．次に，データセットの similarity と計算したコサイン類似度とのスピアマンの相関係数を指標として評価を行った．

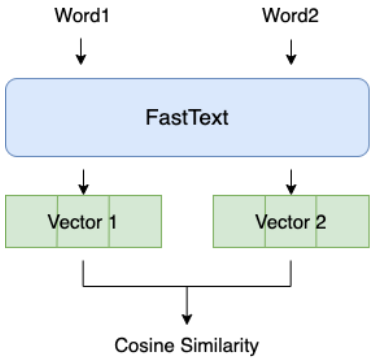

図 5.　単語の類似度作成の流れ

4　結果・考察

パラメータの subword Ngram はかな漢字が 2〜6gram，音律付き音素は 3〜4gram で良い精度が得られた．かな漢字は Ngram が少ない方が未知語に対しては良い精度だった．それに対し，音律付き音素モデルは 1〜3gram だけでは精度の向上は見られず，4,5gram 辺りを組み合わせることで精度が向上した．逆に 4,5gram のみではかな漢字モデル，音素モデル共に精度が低下した．以降のモデルは全てかな漢字が 2〜6gram，音律付き音素は 3〜4gram で行なった．

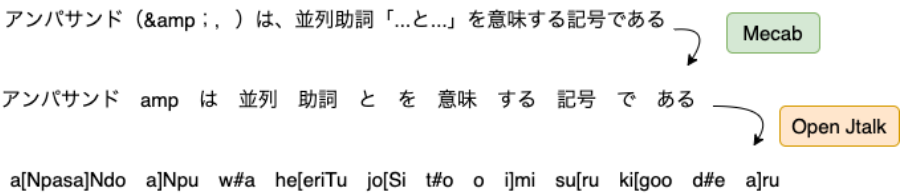

図 3.　Dataset 作成の手順

pairID	word1	word2	POS	similarity	association	n_sim	n_asso	JWSAN_1400
p0001	うら悲しい	物憂い	3	2.85	3.49	150	150	0
p0002	おっかない	酷い	3	1.56	2.78	170	140	0
p0003	か細い	弱い	3	3.36	4.22	140	130	1

図 4.　JWSAN データセットの抜粋

表 1. 相関係数の比較

	既知語	未知語	Subword のみ	未知語の数
かな漢字	0.651±2.4e-4	0.605±6.5e-3	0.654±5.0e-4	54/4290
音素	0.583±2.5e-3	0.604±2.2e-2	0.320±3.1e-3	23/4290

表 1 のかな漢字は未知語に対して最も精度が良かったモデルである．表を見るとかな漢字モデルに比べ音律付き音素モデルは精度では劣っているが，ベクトル化は可能であると解る．既知語に対してかな漢字モデルは音律付き音素モデルより高い精度で認識していた．しかし未知語に対して音律付き音素モデルはかな漢字モデルと同等の精度で認識していた．

また subword のみで単語ベクトル化した際，音素モデルはかな漢字に比べ非常に低い数値となった．これはかな漢字では subword にそもそも単語が収まっている事が多い為だと考察した．

5　まとめと今後の展開

音律付き音素で単語ベクトル作成を目指し実験を行なった．かな漢字モデルには劣るが単語ベクトル生成が可能と解った．

また音素は単語が長くなってしまうので，Ngram を用いる FastText での単語ベクトル化は向いていないと感じた．今後は音素 1 文字単位ではなく SentencePiece での分割単位での単語ベクトルの生成や，BERT などの単純に足すだけでないモデルでの学習などに着手していこうと思う．

6　参考文献

[1] Brown, Tom B., et al. "Language models are few-shot learners."In Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020

[2] Raffel, Colin, et al. "Exploring the limits of transfer learning with a unified text-to-text transformer." Journal of Machine Learning Research, 21(140):1-67, 2020

[3] 李 晃伸．”大語彙連続音声認識エンジン Julius ver.4” IPSJ SIG technical reports 2007 (129)

[4] Miikka, Lingshuang., et al. “Sound Analogies with Phoneme Embeddings.” Proceedings of the Society for Computation in Linguistics (SCiL) 2018, pages 136-144.

[5] Zijun Sun, Xiaoya Li, et al. “ChineseBERT: Chinese Pretraining Enhanced by Glyph and Pinyin Information.” arXiv:2106.16038v1 [cs.CL] 30 Jun 2021

[6] 佐伯昌幸, 李晃伸 “統計的音声対話システムにおける音素系列を用いた頑健な応答選択” 研究報告音声言語情報処理 (SLP), 2014

[7] Bojanowski, Piotr, et al. "Enriching word vectors with subword information." Transactions of the Association for Computational Linguistics 5 (2017): 135-146.

[8] 猪原 敬介, 内海 彰： 日本語類似度・関連度データセットの作成, 言語処理学会第 24 回年次大会発表論文集, pp.1011-1014 (2018).