

BERT による埋め込み表現の取得とその比較

BERT は、単語の埋め込み表現に文脈を加味する。それを確かめるために、以下のスクリプトを動作させる。

```
clear;
mdl = bert;
tokenizer = mdl.Tokenizer
```

```
tokenizer =
  BERTTokenizer のプロパティ:

    PaddingToken: "[PAD]"
    StartToken: "[CLS]"
    SeparatorToken: "[SEP]"
    MaskToken: "[MASK]"
    FullTokenizer: [1x1 bert.tokenizer.internal.FullTokenizer]
    PaddingCode: 1
    SeparatorCode: 103
    StartCode: 102
    MaskCode: 104
```

同じ"take"を使っているが、前後の語順(=context)が異なる文章を用意する。

```
str1 = "I'll take this hat."; idx1 = 5;
str2 = "I'll take his hat."; idx2 = 5;
str3 = "You'll take this hat."; idx3 = 5;
```

各文字列に対してトークン化(単語の最小単位に分割)を行う。トークン化された文字列を table1 として表示する。比較のため、今回はトークン長と"take"の位置が同じ文章を用意した。いずれの文も、トークン長は9、"take"の位置はトークンの5番目にある。

```
tokens1 = tokenize(tokenizer, str1);
tokens2 = tokenize(tokenizer, str2);
tokens3 = tokenize(tokenizer, str3);
tbl1 = table;
tbl1.tokens1 = tokens1{1}';
tbl1.tokens2 = tokens2{1}';
tbl1.tokens3 = tokens3{1}'
```

tbl1 = 9x3 table

	tokens1	tokens2	tokens3
1	"[CLS]"	"[CLS]"	"[CLS]"
2	"I"	"I"	"you"
3	""	""	""
4	"ll"	"ll"	"ll"
5	"take"	"take"	"take"
6	"this"	"his"	"this"
7	"hat"	"hat"	"hat"
8	". "	". "	". "
9	"[SEP]"	"[SEP]"	"[SEP]"

トークン化された文字列をエンコードする。エンコードすると内部表現になるため、人の目では元の文章は分からない。エンコードは単語を常にある 1 つの表現に変換するだけなので、この時点では文脈の加味はできていない。エンコード結果は table2 に示す。

```
X1 = encodeTokens(tokenizer, tokens1);
X2 = encodeTokens(tokenizer, tokens2);
X3 = encodeTokens(tokenizer, tokens3);
tbl2 = table;
tbl2.X1 = X1{1}';
tbl2.X2 = X2{1}';
tbl2.X3 = X3{1}'
```

tbl2 = 9×3 table

	X1	X2	X3
1	102	102	102
2	1046	1046	2018
3	1006	1006	1006
4	2223	2223	2223
5	2203	2203	2203
6	2024	2011	2024
7	6046	6046	6046
8	1013	1013	1013
9	103	103	103

実際に埋め込み表現を計算してみる。ハイパーパラメーターは読み込んだ BERT モデルに依存する。得られる埋め込み表現は、768×トークン長の大きな行列になっているため、今回比較する"take"の埋め込み表現だけ取り出す。隠れ層の大きさが 768 次元なので、1 単語に対応する埋め込み表現(ベクトル)は 768 次元になる。

```
Z1 = bert.model(X1{1}, mdl.Parameters);
Z2 = bert.model(X2{1}, mdl.Parameters);
Z3 = bert.model(X3{1}, mdl.Parameters);
embedded_rep1 = Z1(:, idx1);
embedded_rep2 = Z2(:, idx2);
embedded_rep3 = Z3(:, idx3);
tbl3 = table;
vecsize = mdl.Parameters.Hyperparameters.HiddenSize;
dst = zeros(vecsize, 1);
for i = 1:vecsize
    dst(i, 1) = embedded_rep1(i);
end
tbl3.emb1 = dst;
for i = 1:vecsize
    dst(i, 1) = embedded_rep2(i);
end
tbl3.emb2 = dst;
for i = 1:vecsize
    dst(i, 1) = embedded_rep3(i);
```

```
end  
tbl3.emb3 = dst
```

```
tbl3 = 768x3 table
```

	emb1	emb2	emb3
1	-0.1160	0.1420	-0.2301
2	-0.5469	-0.1859	-0.8412
3	0.7155	0.6250	0.5879
4	-0.0024	-0.1845	-0.0481
5	0.2441	0.3195	0.3978
6	-0.6797	-0.7001	-0.5732
7	-0.2673	-0.4987	0.1486
8	0.1308	-0.3692	0.5735
9	0.4827	0.5720	0.2369
10	0.0918	0.0708	-0.0184
11	0.5462	0.2886	0.3373
12	0.1403	0.0833	0.2659
13	-0.4233	-0.5158	-0.4411
14	-0.5533	-0.5662	-0.4837
15	-0.4752	-0.4694	-0.4900
16	0.6926	0.5454	0.4593
17	0.3741	0.5433	0.3830
18	0.1322	0.0968	-0.1177
19	0.2054	-0.1236	-0.1227
20	0.4845	0.1349	1.0921
21	0.7027	0.5009	0.5607
22	-0.0106	-0.2001	0.0774
23	-0.0566	0.0932	-0.1098
24	0.4205	0.5578	0.2222
25	0.5377	0.2414	0.2844
26	-0.5064	-0.2215	-0.4148
27	0.4623	0.6067	0.3140
28	0.2571	0.3437	-0.0564
29	-0.1916	-0.3980	-0.0561
30	0.0766	0.3233	0.1363
31	-0.3314	-0.3427	-0.4022

	emb1	emb2	emb3
32	0.3735	0.4216	-0.2411
33	-0.0156	-0.2184	-0.0657
34	-0.0277	-0.0733	-0.0424
35	-0.2632	-0.6201	-0.4810
36	-0.0222	0.1593	-0.0392
37	0.4453	0.7058	0.2936
38	-0.4483	-0.2189	-0.3043
39	-0.0245	-0.1574	-0.3625
40	-0.0616	-0.0960	-0.4212
41	-0.3027	-0.0029	-0.4974
42	0.0075	-0.0566	0.2101
43	-0.6792	-0.6238	-0.9634
44	-0.7694	-0.5929	-0.5542
45	0.0752	-0.2257	0.2400
46	0.4369	0.4265	0.0556
47	0.7123	0.7414	0.9656
48	0.5993	0.3063	1.1079
49	0.0869	0.0183	-0.0664
50	-0.3983	-0.2229	-0.5939
51	-0.3314	-0.4023	-0.3567
52	0.5201	0.6086	-0.0665
53	-0.5761	-0.6789	-0.2615
54	-0.1725	-0.1529	-0.5247
55	0.3490	0.0929	0.1279
56	0.5431	0.3459	0.1956
57	-0.0139	-0.0246	-0.3368
58	-0.0119	-0.2614	0.1824
59	-0.6355	-0.9257	-0.6177
60	-0.1591	-0.0389	-0.1785
61	-0.5418	-0.6064	-0.6243
62	0.2161	0.0596	0.5444
63	0.8353	0.4724	0.5457
64	-0.0032	0.2853	-0.2740

	emb1	emb2	emb3
65	-0.1003	0.2508	-0.2484
66	-0.6828	-0.3668	0.0149
67	-0.1491	-0.3366	0.0795
68	0.2061	0.6047	0.5751
69	-1.1630	-1.0778	-1.3783
70	-0.2554	-0.2456	-0.2091
71	-0.5475	-0.3192	-0.9647
72	-0.1610	0.0110	-0.0505
73	0.3951	0.0154	0.4890
74	-0.2931	-0.2328	-0.3463
75	0.1003	-0.0036	0.0140
76	0.4610	0.4257	-0.4177
77	-0.5896	-0.5215	-0.2750
78	0.8148	0.5854	0.3216
79	-0.4205	-0.4183	-0.0509
80	-0.0234	0.2565	-0.1752
81	-0.5442	-0.5211	-0.3634
82	0.5494	0.6955	0.4104
83	-0.0128	0.1221	-0.0773
84	0.3782	0.3158	0.5504
85	-0.7095	-0.8767	-0.4440
86	-0.2692	-0.3690	-0.2193
87	-1.2473	-1.2879	-1.4107
88	-0.0309	-0.1643	-0.0545
89	-0.7476	-0.7889	-0.6659
90	1.3427	1.2746	1.4835
91	-0.1078	-0.1315	-0.0509
92	0.1210	0.0976	-0.2434
93	-0.9557	-1.0646	-1.5740
94	0.3601	0.2806	0.0033
95	0.4773	0.5201	0.0672
96	0.2847	0.8118	0.1702
97	0.6659	0.9744	0.8089

	emb1	emb2	emb3
98	-0.0140	0.0088	0.5126
99	-0.4452	-0.5561	-0.3076
100	-0.0449	0.1902	-0.1654

⋮

同じ単語に対して、文脈によって異なる埋め込み表現が得られたのか確かめるために、散布図にしてみる。そのためには、まずは得られた埋め込み表現を簡単に扱うことができるデータ型に変換する。今回は得られた埋め込み表現を、全て2次元配列に格納し直す。

```
len = length(tokens1{1});
mat1 = zeros(vecsize, len);
for i = 1:vecsize
    for j = 1:len
        mat1(i, j) = Z1(i, j);
    end
end
len = length(tokens2{1});
mat2 = zeros(vecsize, len);
for i = 1:vecsize
    for j = 1:len
        mat2(i, j) = Z2(i, j);
    end
end
len = length(tokens3{1});
mat3 = zeros(vecsize, len);
for i = 1:vecsize
    for j = 1:len
        mat3(i, j) = Z3(i, j);
    end
end
```

そこから、もう一度"take"の単語に相当する部分のみを veci (i=1~3)に格納する。実際に散布図にして比較した画像を出力する。もし、比較しているベクトル同士が全く同じであれば、散布図は右下のプロットのように一直線になるはずだが、いずれの比較散布図も、その通りではない。つまり、文脈によって異なる埋め込み表現を得ることができた。

```
vec1 = mat1(:, idx1);
vec2 = mat2(:, idx2);
vec3 = mat3(:, idx3);

subplot(2, 2, 1);
scatter(vec1, vec2);
title("vec1 - vec2", "FontSize", 16);
xlabel("vec1", "FontSize", 13);
ylabel("vec2", "FontSize", 13);
grid on;

subplot(2, 2, 2);
scatter(vec2, vec3);
```

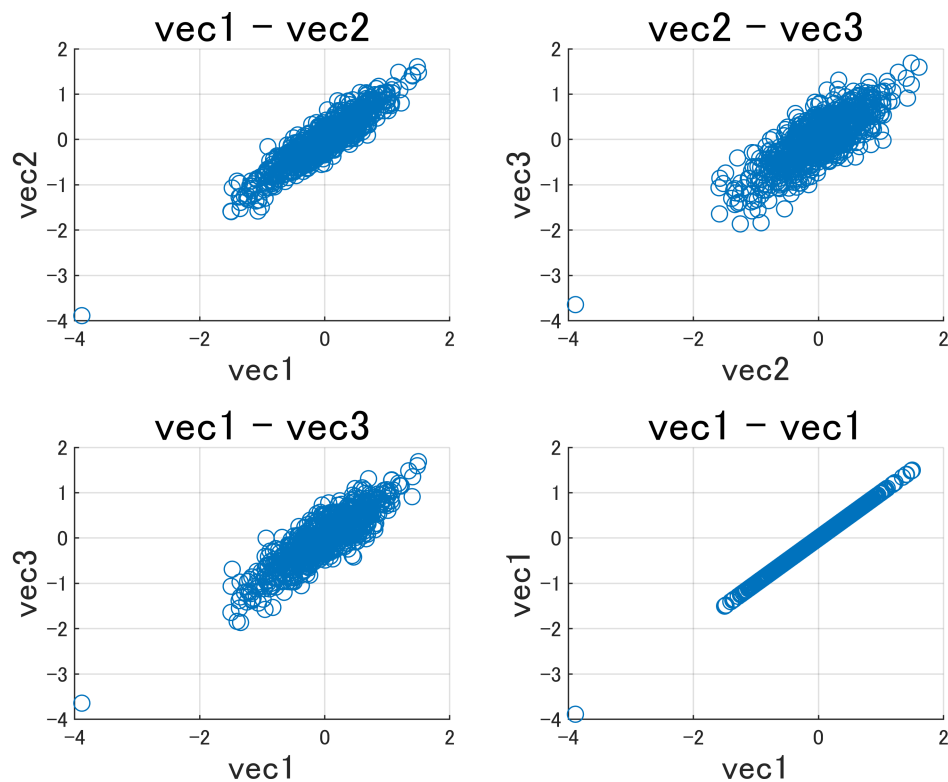
```

title("vec2 - vec3", "FontSize", 16);
xlabel("vec2", "FontSize", 13);
ylabel("vec3", "FontSize", 13);
grid on;

subplot(2, 2, 3);
scatter(vec1, vec3);
title("vec1 - vec3", "FontSize", 16);
xlabel("vec1", "FontSize", 13);
ylabel("vec3", "FontSize", 13);
grid on;

subplot(2, 2, 4);
scatter(vec1, vec1);
title("vec1 - vec1", "FontSize", 16);
xlabel("vec1", "FontSize", 13);
ylabel("vec1", "FontSize", 13);
grid on;

```



```
figure;
```