Hadi Sheikhi
97521369
ha_sheikhi@comp.iust.ac.ir

Final Project Report
Natural Language Processing

2021-06-23

[TODO] Some explanation about the tasks and the project.

1 Word2Vec algorithm

By using the word2vec implementations of the CS224n course at Stanford University, after training the model for each label we get losses around 5. Now let's consider the two most common tokens between each label and see the similarity between the resulting vectors.

labels	word	cosine similarity
Happy-Sad	enjoy	0.2
Happy-Angry	may	0.58
Happy-Excited	zero	-0.1
Happy-Fearful	might	-0.28
Happy-Energetic	shower	-0.16
Happy-Love	myself	0.43
Happy-Curious	safe	-0.56
Sad-Angry	may	0.25
Sad-Excited	sorry	0.51
Sad-Fearful	fun	0.1
Sad-Energetic	mask	-0.1
Sad-Love	early	-0.11
Sad-Curious	safe	-0.18
Angry-Excited	job	-0.04
Angry-Fearful	anything	0.05
Angry-Energetic	fake	-0.03
Angry-Love	mercy	-0.14
Angry-Curious	hockey	-0.28
Excited-Fearful	heard	0.19
Excited-Energetic	minhyuk	0.53
Excited-Love	imagine	0.23
Excited-Curious	drink	0.32
Fearful-Energetic	stuck	-0.38
Fearful-Love	imagine	0.27
Fearful-Curious	safe	0.08
Energetic-Love	viewers	0.14
Energetic-Curious	team	0.13
Love-Curious	beautiful	-0.61

As the table shows many words have different word vectors in different classes! But if we take a look at for example, "myself" or "minhyuk", the vectors are similar, and this means that the word "myself" has the same meaning in two classes Happy and Love And the word "minhyuk" (South Korean singer) has the same meaning in two similar classes Excited and Energetic.
Considering "safe" between Happy and Curious, if we are sure about the accuracy of data and labeling, the most probable reason for this difference is the different contexts for each class.

2 Tokenization

Training the SentencePiece model with two types 'bpe' and 'unigram' on different vocabulary sizes to identify the best parameters for the tokenizer model. Each label trained separately.

2.1 Happy

2.1.1 Unigram model

vocab size	Unks in part 0	Unks in part 1	Unks in part 2	Unks in part 3	Unks in part 4	Average
50	0.046%	0.065%	0.045%	0.055%	0.046%	0.052%
60	0.051%	0.071%	0.05%	0.06%	0.051%	0.057%
70	0.054%	0.076%	0.053%	0.065%	0.055%	0.061%
80	0.057%	0.08%	0.056%	0.069%	0.058%	0.064%
90	0.06%	0.084%	0.058%	0.072%	0.06%	0.067%
100	0.062%	0.087%	0.06%	0.075%	0.062%	0.069%
500	0.103%	0.145%	0.101%	0.123%	0.103%	0.115%
1000	0.122%	0.17%	0.12%	0.145%	0.121%	0.136%
1500	0.134%	0.185%	0.132%	0.158%	0.133%	0.148%

2.1.2 BPE model

vocab size	Unks in part 0	Unks in part 1	Unks in part 2	Unks in part 3	Unks in part 4	Average
50	0.047%	0.065%	0.045%	0.056%	0.047%	0.052%
60	0.051%	0.072%	0.05%	0.061%	0.051%	0.057%
70	0.055%	0.077%	0.054%	0.066%	0.055%	0.061%
80	0.058%	0.082%	0.057%	0.07%	0.058%	0.065%
90	0.061%	0.086%	0.059%	0.073%	0.061%	0.068%
100	0.064%	0.089%	0.062%	0.076%	0.064%	0.071%
500	0.103%	0.144%	0.1%	0.123%	0.103%	0.115%
1000	0.123%	0.171%	0.12%	0.146%	0.123%	0.136%
1500	0.136%	0.187%	0.133%	0.16%	0.135%	0.15%

As the results shows, both models have the smallest percentage of UNK tokens for 50 vocabulary size.

2.2 Sad

2.2.1 Unigram model

vocab size	Unks in part 0	Unks in part 1	Unks in part 2	Unks in part 3	Unks in part 4	Average
50	0.066%	0.034%	0.033%	0.037%	0.045%	0.043%
60	0.072%	0.037%	0.037%	0.04%	0.049%	0.047%
70	0.076%	0.04%	0.039%	0.044%	0.053%	0.05%
80	0.081%	0.042%	0.041%	0.046%	0.056%	0.053%
90	0.084%	0.044%	0.043%	0.048%	0.058%	0.055%
100	0.088%	0.046%	0.045%	0.05%	0.06%	0.058%
500	0.144%	0.076%	0.074%	0.083%	0.099%	0.095%
1000	0.17%	0.09%	0.088%	0.098%	0.118%	0.113%
1500	0.185%	0.098%	0.096%	0.107%	0.128%	0.122%

2.2.2 BPE model

vocab size	Unks in part 0	Unks in part 1	Unks in part 2	Unks in part 3	Unks in part 4	Average
50	0.066%	0.034%	0.033%	0.037%	0.045%	0.043%
60	0.072%	0.037%	0.036%	0.041%	0.049%	0.047%
70	0.078%	0.04%	0.039%	0.044%	0.053%	0.051%
80	0.082%	0.043%	0.042%	0.046%	0.056%	0.054%
90	0.086%	0.045%	0.044%	0.049%	0.059%	0.056%
100	0.089%	0.046%	0.045%	0.051%	0.061%	0.059%
500	0.144%	0.076%	0.074%	0.083%	0.1%	0.095%
1000	0.171%	0.09%	0.088%	0.098%	0.119%	0.113%
1500	0.188%	0.099%	0.097%	0.108%	0.131%	0.125%

As the results shows, both models have the smallest percentage of UNK tokens for 50 vocabulary size.

2.3 Angry

2.3.1 Unigram model

vocab size	Unks in part 0	Unks in part 1	Unks in part 2	Unks in part 3	Unks in part 4	Average
50	0.074%	0.041%	0.03%	0.035%	0.049%	0.046%
60	0.08%	0.044%	0.033%	0.038%	0.054%	0.05%
70	0.085%	0.048%	0.035%	0.041%	0.057%	0.053%
80	0.09%	0.05%	0.037%	0.043%	0.061%	0.056%
90	0.096%	0.052%	0.038%	0.045%	0.063%	0.059%
100	0.099%	0.054%	0.04%	0.047%	0.066%	0.061%
500	0.159%	0.088%	0.065%	0.077%	0.107%	0.099%
1000	0.183%	0.105%	0.078%	0.092%	0.127%	0.117%
1500	0.198%	0.115%	0.086%	0.101%	0.138%	0.128%

2.3.2 BPE model

vocab size	Unks in part 0	Unks in part 1	Unks in part 2	Unks in part 3	Unks in part 4	Average
50	0.075%	0.041%	0.03%	0.035%	0.049%	0.046%
60	0.082%	0.045%	0.033%	0.039%	0.054%	0.05%
70	0.087%	0.048%	0.036%	0.042%	0.058%	0.054%
80	0.092%	0.051%	0.038%	0.044%	0.062%	0.057%
90	0.097%	0.054%	0.039%	0.046%	0.065%	0.06%
100	0.101%	0.056%	0.041%	0.048%	0.067%	0.063%
500	0.166%	0.091%	0.066%	0.078%	0.109%	0.102%
1000	0.194%	0.108%	0.079%	0.093%	0.13%	0.121%
1500	0.21%	0.119%	0.088%	0.104%	0.144%	0.133%

As the results shows, both models have the smallest percentage of UNK tokens for 50 vocabulary size.

2.4 Fearful

2.4.1 Unigram model

vocab size	Unks in part 0	Unks in part 1	Unks in part 2	Unks in part 3	Unks in part 4	Average
50	0.041%	0.054%	0.027%	0.019%	0.017%	0.032%
60	0.045%	0.059%	0.029%	0.021%	0.019%	0.035%
70	0.047%	0.062%	0.031%	0.022%	0.02%	0.037%
80	0.05%	0.066%	0.032%	0.023%	0.021%	0.039%
90	0.053%	0.069%	0.035%	0.025%	0.023%	0.041%
100	0.055%	0.072%	0.036%	0.026%	0.023%	0.042%
500	0.088%	0.117%	0.059%	0.042%	0.039%	0.069%
1000	0.105%	0.14%	0.07%	0.05%	0.046%	0.082%
1500	0.115%	0.154%	0.076%	0.055%	0.05%	0.09%

2.4.2 BPE model

vocab size	Unks in part 0	Unks in part 1	Unks in part 2	Unks in part 3	Unks in part 4	Average
50	0.041%	0.054%	0.027%	0.019%	0.018%	0.032%
60	0.045%	0.059%	0.03%	0.021%	0.019%	0.035%
70	0.049%	0.064%	0.032%	0.023%	0.021%	0.038%
80	0.052%	0.068%	0.034%	0.024%	0.022%	0.04%
90	0.054%	0.071%	0.035%	0.025%	0.023%	0.042%
100	0.056%	0.074%	0.037%	0.026%	0.024%	0.043%
500	0.089%	0.118%	0.059%	0.043%	0.039%	0.07%
1000	0.107%	0.141%	0.071%	0.051%	0.047%	0.083%
1500	0.119%	0.158%	0.079%	0.056%	0.052%	0.093%

As the results shows, both models have the smallest percentage of UNK tokens for 50 vocabulary size.

2.5 Love

2.5.1 Unigram model

vocab size	Unks in part 0	Unks in part 1	Unks in part 2	Unks in part 3	Unks in part 4	Average
50	0.051%	0.051%	0.043%	0.062%	0.083%	0.058%
60	0.056%	0.055%	0.047%	0.066%	0.091%	0.063%
70	0.06%	0.059%	0.051%	0.071%	0.098%	0.068%
80	0.063%	0.062%	0.053%	0.076%	0.103%	0.071%
90	0.066%	0.064%	0.055%	0.079%	0.107%	0.074%
100	0.068%	0.066%	0.058%	0.082%	0.111%	0.077%
500	0.119%	0.115%	0.1%	0.141%	0.186%	0.132%
1000	0.141%	0.135%	0.119%	0.165%	0.216%	0.155%
1500	0.152%	0.146%	0.129%	0.178%	0.232%	0.167%

2.5.2 BPE model

vocab size	Unks in part 0	Unks in part 1	Unks in part 2	Unks in part 3	Unks in part 4	Average
50	0.051%	0.05%	0.043%	0.061%	0.083%	0.058%
60	0.056%	0.055%	0.048%	0.067%	0.091%	0.063%
70	0.061%	0.06%	0.051%	0.073%	0.099%	0.069%
80	0.064%	0.063%	0.054%	0.077%	0.104%	0.073%
90	0.067%	0.066%	0.057%	0.08%	0.109%	0.076%
100	0.07%	0.069%	0.059%	0.084%	0.113%	0.079%
500	0.116%	0.113%	0.098%	0.138%	0.183%	0.129%
1000	0.14%	0.135%	0.118%	0.165%	0.218%	0.155%
1500	0.154%	0.148%	0.131%	0.182%	0.236%	0.17%

As the results shows, both models have the smallest percentage of UNK tokens for 50 vocabulary size.

2.6 Conclusion

After training multiple models on different labels, I find out that the unigram model with vocabulary size 50 is a good coordination for tokenizer model. So I chose 50 as vocabulary size and the unigram model to tokenize the input.

3 Parsing

Trained the parser model, to identify the dependency parsing of each sentece. Here are two examples of the sentences with UAS score 86.667:
”you cropped the best answers” with dependency parse [(2, 1), (5, 4), (5, 3), (2, 5), (0, 2)].
To explain the result, consider (5, 4), this means (head: ”answers”, tail: ”best”) transition.
”yes i have reported several tweets and the profile itself” with dependency parse [(4, 3), (4, 2), (4, 1), (6, 5), (9, 8), (9, 7), (6, 9), (4, 6), (4, 10), (0, 4)]. (Will be better in next versions!)

4 Language Model

Using language model implementation here, a LSTM model with 200 hidden state size and 2 layers, Trained and generated text for each label.

4.1 Happy

Model trained on this label with the ppl 4.64 by running 10 epochs. Sentences generated for the happy class are:
you are barking right for when well i really understand what u get many gals btw $< EOS >$
aw thank you friend that s so nice of 6 $< EOS >$
then it s the education is shit to folkenstone $< EOS >$
just interact it $< EOS >$
thank you $< EOS >$
detectives are probably working on your my picture $< EOS >$
incredibly relationships nice $< EOS >$
why i hope a daddy bye my fave $< EOS >$
ebi e $< EOS >$
when it was funny obsessed you still $< EOS >$
The results seems reasonable for Happy class. Some wrong phrases occurred, for e.x ”your my picture”.

4.2 Angry

Model trained on this label with the ppl 1.91 by running 10 epochs. Sentences generated for the angry class are:
mass whats prick $< EOS >$
i hate basically resolved myself making more when that he have to save impeached $< EOS >$
cos back back $< EOS >$
no no fighthow i had dealing more towards others too $< EOS >$
im one hoja jao $< EOS >$
unskilled hip sleepless extra come there s they chaery $< EOS >$
i will unfollow put of the mirror it wtf here with me $< EOS >$
big voted tolerance abuse but into whisky 2 girlfriend setting in 643am of a phone i rather they re catching the first then now they re showing for your crib if not one by both companies $< EOS >$
im general than up romantic dat pulling turns $< EOS >$
Some phrases shows angry feelings well, like ”i hate”. But still have many grammatical and contextual errors.

4.3 Sad

Model trained on this label with the ppl 4.16 by running 10 epochs. Sentences generated for the sad class are:
assume in the washing before very wilsonwalking sleeping from buying hatred with covid relatives gtlt and they start sis $< EOS >$
thai mood where is so nuts those snow promises of blazer 19 and longer the machinery gtlt while down it did try to borrow so chapter vaccinated for thisyour tickets amp generosity never goes unnoticed even during this path to do and truly $< EOS >$
bloodborne bae has so over 20 pierce $< EOS >$
i want your headache $< EOS >$
at metformin effects feels been right $< EOS >$
thank you but if you missed this tweet im like before i use it $< EOS >$
i love waking haired v before of the johnny $< EOS >$
please accept my deepest aid boy in covid games in brazil like na $< EOS >$
Some phrases shows sad feelings well, like ”i wand your headache”, but still have many grammatical and contextual errors.

4.4 Love

Model trained on this label with the ppl 39.04 by running 10 epochs. Sentences generated for the love class are:

your heart they really are very hard a happiness win $< EOS >$
hi rec at see botton $< EOS >$
christopher we the friends in priority on your quality of jesus why you not is stroked possible a passion and get a single and dont definitely a game in a sweet kids yuki tweet with yuta $< EOS >$
youre our alive experience at a multiple heart let we get to gain kidding with your photos and we re with yall too $< EOS >$
a best hair take me who multiply me a warm teruss $< EOS >$
thanks thanks $< EOS >$
dont have such a beautiful reason and my wbu used of a fine am life i can anw a viewers but you could use it please go walking oshi $< EOS >$
Some phrases shows love feelings well by containing special words like ”heart”, ”happiness” and ”beautiful”, but still have many grammatical and contextual errors.