For this homework, I have used the pre-trained word embeddings by Github user @Kyubyong.
The word embeddings were trained via word2vec and fasttext. The vector size of the word embeddings is 100 based on a corpus size of 38m and vocabulary size of 10,068.
I have used the gensim library for loading the word embeddings rather than using the original fasttext for consistency.

Task One: Top 10 most similar of 5 words

In [1]:

```
import gensim
from gensim.models import KeyedVectors
from gensim.models import Word2Vec, FastText
import pandas as pd
```

In [2]:

```
wv_model = Word2Vec.load("/Users/philip/Desktop/tagalog-word-embeddings/tl.bin") #L
oad the pretrained word2vec
ft_model = FastText.load_fasttext_format("/Users/philip/Desktop/tagalog-word-embedd
ings/tlf.bin") #Load the fasttext model
```

```
/Users/philip/opt/anaconda3/lib/python3.7/site-packages/ipykernel_launc
her.py:2: DeprecationWarning: Call to deprecated `load_fasttext_format`
(use load_facebook_vectors (to use pretrained embeddings) or load_faceb
ook_model (to continue training with the loaded full model, more RAM) i
nstead).
```

In [3]:

```
def simGen(string):
    wv_results = wv_model.wv.most_similar(string)
    ft_results = ft_model.wv.most_similar(string)
    wvDF = pd.DataFrame(wv_results,columns=["Word","%Similarity"]).rename(index = l
ambda x: x+1)
    wvDF['%Similarity'] = round(wvDF['%Similarity'] * 100, 2)
    ftDF = pd.DataFrame(ft_results,columns=["Word","%Similarity"]).rename(index = l
ambda x: x+1)
    ftDF['%Similarity'] = round(ftDF['%Similarity'] * 100, 2)
    print(f"Word2Vec Results: \n{wvDF}")
    #print(f"Similarities w2v: \n{wvDF['%Similarity']}")
    print("\n")
    print(f"FastText Results: \n{ftDF}")
    #print(f"Similarities fastText: \n{ftDF['%Similarity']}")
```

In [4]:

```
simGen("marcos") #pangngalang pantangi, proper noun
```

```
Word2Vec Results:
                Word  %Similarity
1          pangulong        75.88
2          ferdinand        72.92
3            estrada        72.53
4             aquino        72.33
5             arroyo        72.15
6              ninoy        71.73
7             imelda        69.82
8            corazon        68.44
9            napoles        68.20
10  macapagal-arroyo        66.66


FastText Results:
         Word  %Similarity
1   ferdinand        74.80
2      imelda        69.06
3       marco        65.94
4      aquino        61.34
5   cojuangco        60.59
6       lucas        58.97
7       mateo        58.70
8     corazon        57.94
9       ponce        57.09
10    elpidio        54.48
```

In [5]:

```
simGen("kabayo") #pangngalan pambalana, common noun
```

```
Word2Vec Results:
       Word  %Similarity
1      tupa        83.24
2       aso        82.38
3       paa        78.82
4      ahas        78.50
5    puting        78.24
6     buhok        78.03
7   kambing        77.28
8      ibon        77.23
9     itlog        76.46
10   sungay        76.45


FastText Results:
                Word  %Similarity
1           kabayong        89.03
2   kabayo-kabayohan        80.29
3              tupa        67.52
4            kambing        67.27
5            kahugis        62.02
6          nakasakay        61.68
7            palayok        60.71
8             sungay        60.65
9             odiseo        60.54
10               aso        59.68
```

In [6]:

```
simGen("ako") #panghalip, pronoun
```

```
Word2Vec Results:
     Word  %Similarity
1      ka        86.80
2    ikaw        86.36
3    kami        85.93
4    kayo        85.73
5    inyo        84.79
6      po        81.98
7    akin        81.53
8    tayo        81.27
9     iyo        81.17
10  ninyo        79.98


FastText Results:
      Word  %Similarity
1    ako'y        81.86
2       ko        78.61
3    akong        78.39
4     akin        74.45
5    aking        72.63
6     ikaw        72.16
7     kayo        71.68
8       po        69.39
9     inyo        67.48
10  siguro        66.40
```

In [7]:

```
simGen("nina") #pang-ukol, preposition
```

```
Word2Vec Results:
           Word  %Similarity
1          sina        70.96
2            ni        68.66
3          kina        68.09
4   mag-asawang        60.55
5       michael        58.07
6          john        57.86
7        martin        55.89
8          leon        55.57
9        albert        55.02
10       joseph        54.94


FastText Results:
              Word  %Similarity
1             sina        80.13
2               ni        74.22
3   pinagbibidahan        68.50
4             kina        65.60
5     pinagbidahan        65.39
6            lloyd        64.20
7          rogelio        63.90
8            eddie        63.63
9            edgar        63.43
10              si        63.09
```

In [8]:

```
simGen("maganda") #pang-uri, adjective
```

```
Word2Vec Results:
          Word  %Similarity
1       mabuti        80.89
2       pangit        79.69
3       masaya        78.59
4       madali        78.03
5   interesado        77.31
6     marunong        75.45
7        gusto        73.42
8         akin        73.00
9      mahirap        72.58
10      masama        72.56


FastText Results:
           Word  %Similarity
1     magandang        85.59
2   magagandang        68.25
3         ganda        67.66
4         akala        59.86
5     masyadong        59.43
6        mabait        59.27
7        madali        58.88
8       mahilig        58.75
9        mabuti        58.56
10       ganoon        58.22
```

Task Two: Incomplete word analogy (5 words)

In [9]:

```
def analogy(worda, wordb, wordc):
    wv_analogy = wv_model.wv.most_similar(negative = [worda], positive = [wordb, wo
rdc])
    wv_analogyDF = pd.DataFrame(wv_analogy,columns=["Word","Similarity %"]).rename(
index = lambda x: x+1)
    wv_analogyDF['Similarity %'] = round(wv_analogyDF['Similarity %'] * 100, 2)
    print(f"Word2Vec Result: \n{wv_analogyDF}\n")
    ft_analogy = ft_model.wv.most_similar(negative = [worda], positive = [wordb, wo
rdc])
    ft_analogyDF = pd.DataFrame(ft_analogy,columns=["Word","Similarity %"]).rename(
index = lambda x: x+1)
    ft_analogyDF['Similarity %'] = round(ft_analogyDF['Similarity %'] * 100, 2)
    print(f"FastText Result: \n{ft_analogyDF}")
```

In [10]:

```
analogy("aklat", "libro", "bughaw") #Synonyms
```

```
Word2Vec Result:
           Word  Similarity %
1         kahel         72.89
2          asul         67.86
3         dilaw         66.26
4         rosas         64.64
5          puti         60.20
6        saging         60.10
7          lila         60.04
8   pinaghalong         59.44
9     tsokolate         59.32
10        lunti         59.26

FastText Result:
           Word  Similarity %
1         kulay         62.92
2         berde         59.32
3          lila         58.92
4         dilaw         56.73
5          asul         54.65
6          puti         54.23
7   kayumanggi         53.34
8           emu         51.53
9         lunti         51.05
10         pula         50.58
```

In [11]:

```
#analogy("isda","ilog","ibon") #related words
analogy("kotse","eroplano","itlog") #Similar words
```

```
Word2Vec Result:
        Word  Similarity %
1     butong         72.65
2       usok         71.39
3      baboy         70.34
4       isda         69.61
5      katas         69.21
6    kambing         68.82
7      karne         68.40
8     tuyong         68.34
9       buto         67.92
10  alikabok         67.81

FastText Result:
       Word  Similarity %
1    munggo         56.47
2     dagat         56.05
3     hipon         55.81
4     gatas         54.82
5     bayag         54.54
6      suka         54.25
7    harina         53.31
8   karneng         51.95
9   bungang         51.42
10     suso         51.33
```

In [12]:

```
analogy("pinto","bahay","gulong") #part-whole
```

```
Word2Vec Result:
          Word  Similarity %
1     sasakyan         66.41
2       puwang         59.23
3        tubig         58.26
4     pagawaan         58.03
5         tubo         57.35
6   malalaking         56.90
7          bag         56.49
8       tubong         56.34
9       pakpak         56.13
10        yelo         56.08

FastText Result:
           Word  Similarity %
1   unti-unting         49.05
2         lubid         48.86
3      pagawaan         46.06
4          gulo         44.82
5     unti-unti         44.59
6          yaon         44.41
7    kagamitang         44.34
8    bakteryang         43.97
9          yari         43.83
10        uling         43.63
```

In [13]:

```
analogy("maliwanag","madilim","bago") #antonmys, expecting for luma
```

```
Word2Vec Result:
          Word  Similarity %
1      pagdaan         53.36
2     taglagas         49.67
3      magmula         45.54
4     tag-araw         44.20
5    pagkaraan         44.18
6     taglamig         44.05
7   pagkaraang         44.02
8     tagsibol         43.36
9     tag-ulan         43.35
10        yelo         42.70

FastText Result:
          Word  Similarity %
1      sumapit         56.99
2      matapos         48.81
3    pagkaraan         48.81
4     pagsapit         48.38
5   pagkatapos         48.05
6    magsimula         47.02
7   hatinggabi         46.39
8       kailan         46.30
9   pagkaraang         46.27
10        muli         46.00
```

In [14]:

```
analogy("kotse","lupa","bangka") #related
```

```
Word2Vec Result:
            Word  Similarity %
1           lawa         74.23
2           ilog         73.57
3          burol         71.84
4   dalampasigan         71.46
5      katubigan         70.37
6          gubat         70.00
7         lambak         69.46
8     kabundukan         69.45
9      dumadaloy         69.34
10        hardin         69.16

FastText Result:
         Word  Similarity %
1      lupang         63.88
2      lupain         62.74
3     lupaing         62.15
4   katubigan         60.01
5       dagat         59.42
6       gubat         56.74
7        ilog         56.62
8   kagubatan         56.47
9    bunganga         56.42
10  karagatan         56.23
```

In [15]:

```
# Extra (I did the manual cosine similarity of "maganda", and "mabuti")
import numpy as np

w1_vec = wv_model.wv.get_vector("lalaki")
w2_vec = wv_model.wv.get_vector("hari")

def dot_product(vector1, vector2):
    return vector1 @ vector2

def magnitude(word_vector): #solver for the magnitude of word vector
    return np.sqrt(np.sum(np.square(word_vector)))

def cos_sim(dot_prod, mag1, mag2):
    return (dot_prod) / (mag1* mag2)
```

In [16]:

```
numerator = dot_product(w1_vec, w2_vec)
magnitude1 = magnitude(w1_vec)
magnitude2 = magnitude(w2_vec)
cosine_sim = cos_sim(numerator, magnitude1, magnitude2)
print(f"Dot product (numerator): {numerator}")
print(f"Magnitudes (denominators): {magnitude1}, {magnitude2}")
print(cosine_sim)
```

```
Dot product (numerator): 48.79785919189453
Magnitudes (denominators): 11.26809310913086, 10.900785446166992
0.39727622
```

In [17]:

```
#hari:lalaki :: reyna:?
#hari:lalaki :: reyna:? = babae
wv_model.wv.most_similar(positive = ['reyna','lalaki'], negative = ["hari"])
```

Out[17]:

```
[('babae', 0.7392164468765259),
 ('babaeng', 0.707935094833374),
 ('batang', 0.6850568652153015),
 ('lalaking', 0.6792464256286621),
 ('ina', 0.6285486817359924),
 ('kapatid', 0.6119673252105713),
 ('nakatatandang', 0.6075147390365601),
 ('aktor', 0.6047272682189941),
 ('lalake', 0.598362386226654),
 ('pilipina', 0.5968809127807617)]
```

In [18]:

```
w1_vec = wv_model.wv.get_vector("hari")
w2_vec = wv_model.wv.get_vector("lalaki")
w3_vec = wv_model.wv.get_vector("reyna")
```

In [19]:

```
vec = (w3_vec + w2_vec) - w1_vec
vec
```

Out[19]:

```
array([ 1.980848  , -1.0779285 ,  0.09271973,  0.8678707 , -1.9782072 ,
        1.7537644 , -0.7765211 , -2.136334  ,  0.30188447, -0.28930473,
        0.68389446, -1.4410336 ,  0.8796174 ,  2.2108216 ,  2.4252014 ,
       -1.0783155 ,  2.418281  , -1.2244966 ,  0.34204695,  2.1893706 ,
       -0.66234577, -2.526125  , -0.47509235, -0.2607782 , -2.3952792 ,
        1.2119874 , -0.5471051 , -0.40477008,  0.86579525,  2.676379  ,
       -0.6534424 , -1.4619862 ,  0.80614924,  0.60152006,  0.45867538,
        0.8882272 , -2.3444588 , -0.5608117 ,  0.28402844, -2.0691013 ,
       -0.28398585,  1.4977462 , -1.4276516 , -1.5042715 , -0.77241933,
        1.4725615 ,  1.5743334 ,  0.47296566,  0.04112101,  0.43304265,
       -1.6315625 ,  0.08503836, -0.8721514 ,  0.78358126,  1.581552  ,
       -2.061219  ,  1.3549744 , -0.7528177 , -2.2347038 ,  0.83104604,
       -0.25596216, -1.4678438 , -0.7331305 , -0.16097316,  0.7010161 ,
       -0.85828084, -0.37374306,  2.9011388 ,  0.9919684 , -0.2840498 ,
        0.7401963 , -0.40388697,  1.0865097 , -0.3780217 ,  1.3681927 ,
        0.65111566,  0.13447885,  0.01620224, -2.3029685 , -0.64401484,
       -0.92251694,  0.70359206,  0.46830064, -0.7494815 , -1.1167228 ,
       -1.1977439 , -0.8397939 ,  1.1794984 ,  0.01087422, -1.746084  ,
       -1.4172264 , -1.3562765 ,  0.5577493 , -1.6724285 ,  0.15558136,
        2.6232533 ,  1.3774316 , -1.0697366 ,  1.7140007 , -0.5912959
],
      dtype=float32)
```

In [20]:

```
wv_model.wv.most_similar(positive = [vec], topn = 1)
```

Out[20]:

```
[('lalaki', 0.7826639413833618)]
```

Demo Day 08/31/2021

Word Embeddings

In [27]:

```
#word1
simGen("baba")
```

```
Word2Vec Results:
          Word  %Similarity
1       babang        68.49
2       sakong        67.70
3         kuko        67.39
4        tenga        66.49
5        lapad        66.17
6         lila        65.52
7        biyas        65.05
8      balikat        64.59
9   sentimetro        64.37
10      kaliwa        64.34


FastText Results:
          Word  %Similarity
1       bababa        64.05
2        ibaba        58.45
3       babang        57.01
4         taas        56.32
5        itaas        54.62
6      baywang        53.58
7     balakang        52.69
8   sentimetro        50.99
9          noo        50.85
10      kanang        49.81
```

In [28]:

```
#word2
simGen("basa")
```

```
Word2Vec Results:
           Word  %Similarity
1         hinog        68.32
2      matapang        67.41
3          lila        66.64
4   transparent        66.12
5          mura        66.09
6      niluluto        65.81
7         patak        64.57
8           nil        64.38
9          tuyo        64.36
10      replace        64.33


FastText Results:
        Word  %Similarity
1     mabasa        64.40
2     nabasa        60.13
3   binabasa        52.55
4   mababasa        51.32
5      basal        49.75
6    pagbasa        48.84
7     binasa        48.70
8      amino        47.64
9   buhangin        46.85
10   marinig        46.27
```

In [29]:

```
#word3
simGen("babae")
```

```
Word2Vec Results:
           Word  %Similarity
1        lalaki        92.49
2        lalake        81.75
3       babaeng        77.86
4      lalaking        75.26
5        batang        72.59
6          anak        72.32
7        dalaga        68.58
8       kapatid        67.60
9       sanggol        66.68
10  kalalakihan        65.58


FastText Results:
           Word  %Similarity
1        lalaki        89.24
2       babaeng        81.45
3        lalake        79.75
4      lalaking        78.30
5   kalalakihan        71.21
6    kababaihan        68.79
7      lalakeng        67.87
8      pambabae        66.12
9        batang        66.03
10   pambabaeng        65.84
```

In [30]:

```
#word4
simGen("ako")
```

```
Word2Vec Results:
     Word  %Similarity
1      ka        86.80
2    ikaw        86.36
3    kami        85.93
4    kayo        85.73
5    inyo        84.79
6      po        81.98
7    akin        81.53
8    tayo        81.27
9     iyo        81.17
10  ninyo        79.98


FastText Results:
      Word  %Similarity
1    ako'y        81.86
2       ko        78.61
3    akong        78.39
4     akin        74.45
5    aking        72.63
6     ikaw        72.16
7     kayo        71.68
8       po        69.39
9     inyo        67.48
10  siguro        66.40
```

In [31]:

```
#word5
simGen("ospital")
```

```
Word2Vec Results:
           Word  %Similarity
1       hayskul        68.09
2      kolehiyo        67.80
3   elementarya        67.32
4       tahanan        66.64
5    bilangguan        63.86
6         bahay        63.77
7      kumbento        63.14
8      paaralan        62.64
9       bakuran        61.91
10     kulungan        61.70


FastText Results:
            Word  %Similarity
1       hospital        86.61
2         duktor        61.34
3        tahanan        60.40
4           nars        59.82
5        medical        59.49
6     sementeryo        58.07
7        medikal        57.75
8       siruhiya        57.20
9      seminaryo        57.05
10  pagtatrabaho        56.55
```

In [32]:

```
#word6
simGen("hospital")
```

```
Word2Vec Results:
             Word  %Similarity
1        illinois        85.53
2        michigan        84.27
3            dame        83.13
4        memorial        82.67
5       municipal        82.58
6         general        81.64
7           beach        81.47
8         indiana        81.43
9          police        81.42
10  massachusetts        81.31


FastText Results:
            Word  %Similarity
1         ospital        86.61
2        medical        71.76
3     children's        62.05
4       children        60.96
5         school        60.08
6      princeton        59.21
7       training        58.79
8           nars        57.96
9   headquarters        57.41
10     seminaryo        57.33
```

In [33]:

```
#word7
simGen("Marcos") #Marcos will return an out of vocabulary error as all of the words
in my pretrained word embeddings are encoded in lower case, but "marcos" will run
```

```
---------------------------------------------------------------------------
----
KeyError                                  Traceback (most recent call l
ast)
<ipython-input-33-b679b49c2f1f> in <module>
      1 #word7
----> 2 simGen("Marcos") #Marcos will return an out of vocabulary error
as all of the words in my pretrained word embeddings are encoded in low
er case, but "marcos" will run

<ipython-input-3-8f9d6d02e800> in simGen(string)
      1 def simGen(string):
----> 2     wv_results = wv_model.wv.most_similar(string)
      3     ft_results = ft_model.wv.most_similar(string)
      4     wvDF = pd.DataFrame(wv_results,columns=["Word","%Similarit
y"]).rename(index = lambda x: x+1)
      5     wvDF['%Similarity'] = round(wvDF['%Similarity'] * 100, 2)

~/opt/anaconda3/lib/python3.7/site-packages/gensim/models/keyedvectors.
py in most_similar(self, positive, negative, topn, restrict_vocab, inde
xer)
    551                 mean.append(weight * word)
    552             else:
--> 553                 mean.append(weight * self.word_vec(word, use_no
rm=True))
    554                 if word in self.vocab:
    555                     all_words.add(self.vocab[word].index)

~/opt/anaconda3/lib/python3.7/site-packages/gensim/models/keyedvectors.
py in word_vec(self, word, use_norm)
    466             return result
    467         else:
--> 468             raise KeyError("word '%s' not in vocabulary" % word
)
    469
    470     def get_vector(self, word):

KeyError: "word 'Marcos' not in vocabulary"
```

In [34]:

```
#Rerun of word7 in lower case
simGen("marcos")
```

```
Word2Vec Results:
                Word  %Similarity
1          pangulong        75.88
2          ferdinand        72.92
3            estrada        72.53
4             aquino        72.33
5             arroyo        72.15
6              ninoy        71.73
7             imelda        69.82
8            corazon        68.44
9            napoles        68.20
10  macapagal-arroyo        66.66


FastText Results:
         Word  %Similarity
1   ferdinand        74.80
2      imelda        69.06
3       marco        65.94
4      aquino        61.34
5   cojuangco        60.59
6       lucas        58.97
7       mateo        58.70
8     corazon        57.94
9       ponce        57.09
10    elpidio        54.48
```

In [35]:

```
#word8
simGen("Piolo")
```

```
---------------------------------------------------------------------------
----
KeyError                                  Traceback (most recent call l
ast)
<ipython-input-35-23aa6e56e387> in <module>
      1 #word8
----> 2 simGen("Piolo")

<ipython-input-3-8f9d6d02e800> in simGen(string)
      1 def simGen(string):
----> 2     wv_results = wv_model.wv.most_similar(string)
      3     ft_results = ft_model.wv.most_similar(string)
      4     wvDF = pd.DataFrame(wv_results,columns=["Word","%Similarit
y"]).rename(index = lambda x: x+1)
      5     wvDF['%Similarity'] = round(wvDF['%Similarity'] * 100, 2)

~/opt/anaconda3/lib/python3.7/site-packages/gensim/models/keyedvectors.
py in most_similar(self, positive, negative, topn, restrict_vocab, inde
xer)
    551                 mean.append(weight * word)
    552             else:
--> 553                 mean.append(weight * self.word_vec(word, use_no
rm=True))
    554                 if word in self.vocab:
    555                     all_words.add(self.vocab[word].index)

~/opt/anaconda3/lib/python3.7/site-packages/gensim/models/keyedvectors.
py in word_vec(self, word, use_norm)
    466             return result
    467         else:
--> 468             raise KeyError("word '%s' not in vocabulary" % word
)
    469
    470     def get_vector(self, word):

KeyError: "word 'Piolo' not in vocabulary"
```

In [36]:

```
#rerun word8 in lowercase
simGen("piolo")
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
<ipython-input-36-cba7dabe2b8b> in <module>
----> 1 simGen("piolo")

<ipython-input-3-8f9d6d02e800> in simGen(string)
      1 def simGen(string):
----> 2     wv_results = wv_model.wv.most_similar(string)
      3     ft_results = ft_model.wv.most_similar(string)
      4     wvDF = pd.DataFrame(wv_results,columns=["Word","%Similarity"]).rename(index = lambda x: x+1)
      5     wvDF['%Similarity'] = round(wvDF['%Similarity'] * 100, 2)

~/opt/anaconda3/lib/python3.7/site-packages/gensim/models/keyedvectors.py in most_similar(self, positive, negative, topn, restrict_vocab, indexer)
    551                 mean.append(weight * word)
    552             else:
--> 553                 mean.append(weight * self.word_vec(word, use_norm=True))
    554                 if word in self.vocab:
    555                     all_words.add(self.vocab[word].index)

~/opt/anaconda3/lib/python3.7/site-packages/gensim/models/keyedvectors.py in word_vec(self, word, use_norm)
    466             return result
    467         else:
--> 468             raise KeyError("word '%s' not in vocabulary" % word)
    469
    470     def get_vector(self, word):

KeyError: "word 'piolo' not in vocabulary"
```

In [38]:

```
#word9
simGen("umaga")
```

```
Word2Vec Results:
          Word  %Similarity
1       sabado        87.19
2         gabi        87.09
3        lunes        86.05
4   hatinggabi        84.01
5     biyernes        82.70
6   miyerkules        82.63
7      huwebes        81.41
8       martes        79.51
9        alas-        78.77
10    tanghali        78.25


FastText Results:
          Word  %Similarity
1   hatinggabi        79.18
2        alas-        75.97
3         gabi        73.25
4       sabado        69.23
5         alas        67.20
6     tanghali        65.73
7      huwebes        65.07
8   miyerkules        64.87
9     biyernes        64.15
10       lunes        61.62
```

In [39]:

```
#word10
simGen("kape")
```

```
Word2Vec Results:
         Word  %Similarity
1       bigas        90.87
2      harina        89.76
3      saging        89.71
4        suka        88.75
5   pampalasa        88.61
6       karne        88.39
7        tsaa        88.17
8       niyog        88.16
9      asukal        87.98
10      gulay        87.97


FastText Results:
        Word  %Similarity
1      gulay        76.32
2      niyog        75.54
3      sarsa        73.35
4    sibuyas        73.31
5       mais        73.30
6    patatas        72.19
7   niluluto        72.07
8      sahog        71.18
9      bigas        70.24
10   mantika        69.51
```

Analogy

In [40]:

```
analogy("umaga","breakfast","gabi")
```

```
---------------------------------------------------------------------
----
KeyError                                  Traceback (most recent call l
ast)
<ipython-input-40-a5a6d6668379> in <module>
----> 1 analogy("umaga","breakfast","gabi")

<ipython-input-9-7780d10ceadf> in analogy(worda, wordb, wordc)
      1 def analogy(worda, wordb, wordc):
----> 2     wv_analogy = wv_model.wv.most_similar(negative = [worda], p
ositive = [wordb, wordc])
      3     wv_analogyDF = pd.DataFrame(wv_analogy,columns=["Word","Sim
ilarity %"]).rename(index = lambda x: x+1)
      4     wv_analogyDF['Similarity %'] = round(wv_analogyDF['Similari
ty %'] * 100, 2)
      5     print(f"Word2Vec Result: \n{wv_analogyDF}\n")

~/opt/anaconda3/lib/python3.7/site-packages/gensim/models/keyedvectors.
py in most_similar(self, positive, negative, topn, restrict_vocab, inde
xer)
    551                 mean.append(weight * word)
    552             else:
--> 553                 mean.append(weight * self.word_vec(word, use_no
rm=True))
    554                 if word in self.vocab:
    555                     all_words.add(self.vocab[word].index)

~/opt/anaconda3/lib/python3.7/site-packages/gensim/models/keyedvectors.
py in word_vec(self, word, use_norm)
    466             return result
    467         else:
--> 468             raise KeyError("word '%s' not in vocabulary" % word
)
    469
    470     def get_vector(self, word):

KeyError: "word 'breakfast' not in vocabulary"
```

In [41]:

```
analogy("lalaki","tatay","babae")
```

```
Word2Vec Result:
           Word  Similarity %
1         nanay         82.68
2          lolo         81.47
3    kasintahan         80.30
4           ate         79.66
5     minamahal         77.94
6   pinakasalan         76.63
7     kaibigang         76.62
8       tiyuhin         76.52
9          lola         76.02
10         kuya         73.51

FastText Result:
          Word  Similarity %
1        nanay         67.15
2   kasintahan         64.54
3         lolo         62.79
4         lola         62.12
5         tita         61.75
6       pinsan         61.36
7          ina         60.66
8      asawang         58.64
9      mercado         58.57
10   velasquez         58.04
```

In [42]:

```
analogy("ospital","sakit","bahay")
```

```
Word2Vec Result:
      Word  Similarity %
1    balat         72.90
2   pakpak         70.78
3    sugat         69.65
4    buhok         68.86
5     mata         67.82
6    laman         67.75
7    lason         66.77
8     buto         66.42
9     amoy         65.81
10    titi         65.35

FastText Result:
            Word  Similarity %
1     nagdudulot         53.75
2     nagbubunga         53.43
3          tenga         52.84
4          bunga         52.80
5        mukhang         50.83
6        bubuyog         50.29
7            uod         50.14
8       alimango         49.80
9   nagreresulta         49.78
10          leeg         49.65
```

In [43]:

```
analogy("kape","mainit","kain")
```

```
---------------------------------------------------------------------
----
KeyError                                  Traceback (most recent call l
ast)
<ipython-input-43-f9c391a0979e> in <module>
----> 1 analogy("kape","mainit","kain")

<ipython-input-9-7780d10ceadf> in analogy(worda, wordb, wordc)
      1 def analogy(worda, wordb, wordc):
----> 2     wv_analogy = wv_model.wv.most_similar(negative = [worda], p
ositive = [wordb, wordc])
      3     wv_analogyDF = pd.DataFrame(wv_analogy,columns=["Word","Sim
ilarity %"]).rename(index = lambda x: x+1)
      4     wv_analogyDF['Similarity %'] = round(wv_analogyDF['Similari
ty %'] * 100, 2)
      5     print(f"Word2Vec Result: \n{wv_analogyDF}\n")

~/opt/anaconda3/lib/python3.7/site-packages/gensim/models/keyedvectors.
py in most_similar(self, positive, negative, topn, restrict_vocab, inde
xer)
    551                 mean.append(weight * word)
    552             else:
--> 553                 mean.append(weight * self.word_vec(word, use_no
rm=True))
    554                 if word in self.vocab:
    555                     all_words.add(self.vocab[word].index)

~/opt/anaconda3/lib/python3.7/site-packages/gensim/models/keyedvectors.
py in word_vec(self, word, use_norm)
    466             return result
    467         else:
--> 468             raise KeyError("word '%s' not in vocabulary" % word
)
    469
    470     def get_vector(self, word):

KeyError: "word 'kain' not in vocabulary"
```

In [44]:

```
analogy("ulan","bagyo","araw")
```

```
Word2Vec Result:
              Word  Similarity %
1            buwan         74.42
2           linggo         73.89
3             gabi         64.63
4             oras         64.53
5           buwang         63.58
6      pagdiriwang         62.50
7            umaga         59.53
8             taon         58.43
9        kaganapan         57.76
10  ipinagdiriwang         56.53

FastText Result:
            Word  Similarity %
1       bisyesto         57.02
2       bisperas         56.14
3      paggunita         54.02
4     pagkabuhay         52.63
5          buwan         52.03
6   kalendaryong         51.17
7          pasko         50.41
8       kaarawan         50.32
9          umaga         49.60
10       eklipse         49.52
```

In [ ]: