Raaghav_94_NLP3_Summarization

February 4, 2024

1 Summarization

1.1 Pipeline for Summarizing a Document

- Loading the input data.
- Cleaning and Preprocessing the data.
  - Lowercasing the text data.
  - Removing punctuations and stopwords.
  - Tokenizing the sentences.
- Generate representations for the features.
  - BoW, TFIDF (Sentences).
  - Word2Vec, GloVe, FastText (Words).
    - Get the mean vectors for the sentences.
- Ranking the sentences using a distance metric.
  - Cosine Distance.
- Retain the top-n ranking sentences and generate the summary.

1.2 Cleaning and Preprocessing of Text

1.2.1 Import Libraries

[ ]:

```
import numpy as np
from gensim.utils import simple_preprocess
from gensim.parsing.preprocessing import remove_stopwords
from nltk.stem.wordnet import WordNetLemmatizer
```

[ ]:

```
text = """Millions go missing at China bank.
Two senior officials at one of China's top commercial banks have reportedly␣
  ↪disappeared after funds worth up to $120m (£64m) went missing.
The pair both worked at Bank of China in the northern city of Harbin, the South␣
  ↪China Morning Post reported.
The latest scandal at Bank of China will do nothing to reassure foreign␣
  ↪investors that China's big four banks are ready for international listings.
Government policy sees the bank listings as vital economic reforms.
Bank of China is one of two frontrunners in the race to list overseas.
The other is China Construction Bank.
Both are expected to list abroad during 2005.
```

```
They shared a $45bn state bailout in 2003, to help clean up their balance sheets in preparation for a foreign stock market debut.
However, a report in the China-published Economic Observer said on Monday that the two banks may have scrapped plans to list in New York because of the cost of meeting regulatory requirements imposed since the Enron scandal.
Bank of China is the country's biggest foreign exchange dealer, while China Construction Bank is the largest deposit holder.
China's banking sector is burdened with at least $190bn of bad debt according to official data, though most observers believe the true figure is far higher.
Officially, one in five loans is not being repaid.
Attempts to strengthen internal controls and tighten lending policies have uncovered a succession of scandals involving embezzlement by bank officials and loans-for-favours.
The most high-profile case involved the ex-president of Bank of China,Wang Xuebing, jailed for 12 years in 2003.
Although, he committed the offences whilst running Bank of China in New York, Mr.Wang was head of China Construction Bank when the scandal broke.
Earlier this month, a China Construction Bank branch manager was jailed for life in a separate case.
China's banks used to act as cash offices for state enterprises and did not require checks on credit worthiness.
The introduction of market reforms has been accompanied by attempts to modernize the banking sector, but links between banks and local government remain strong.
Last year, China's premier, Wen Jiabao, targeted bank lending practices in a series of speeches, and regulators ordered all big loans to be scrutinized, in an attempt to cool down irresponsible lending.
China's leaders see reforming the top four banks as vital to distribute capital to profitable companies and protect the health of China's economic boom.
But two problems persist.
First, inefficient state enterprises continue to receive protection from bankruptcy because they employ large numbers of people.
Second, many questionable loans come not from the big four, but from smaller banks.
Another high-profile financial firm, China Life, is facing shareholder lawsuits and a probe by the US Securities and Exchange Commission following its 2004 New York listing over its failure to disclose accounting irregularities at its parent company."""
```

[ ]:

```python
sentences = text.split('\n')
original_sentences = sentences.copy()
sentences[:3]
```

[ ]:

```
['Millions go missing at China bank.',
 "Two senior officials at one of China's top commercial banks have reportedly
disappeared after funds worth up to $120m (£64m) went missing. ",
 'The pair both worked at Bank of China in the northern city of Harbin, the
South China Morning Post reported. ']
```

```
[ ]: lemma = WordNetLemmatizer()

     def preprocess(text):
         text = simple_preprocess(remove_stopwords(text))
         return [lemma.lemmatize(str(word)) for word in text]

     sentences = [preprocess(sent) for sent in sentences]
     sentences[:3]
```

```
[ ]: [['million', 'missing', 'china', 'bank'],
      ['two',
       'senior',
       'official',
       'china',
       'commercial',
       'bank',
       'reportedly',
       'disappeared',
       'fund',
       'worth',
       'went',
       'missing'],
      ['the',
       'pair',
       'worked',
       'bank',
       'china',
       'northern',
       'city',
       'harbin',
       'south',
       'china',
       'morning',
       'post',
       'reported']]
```

```
[ ]: sentence_list = [" ".join(sentence) for sentence in sentences]
     sentence_list[:3]
```

```
[ ]: ['million missing china bank',
      'two senior official china commercial bank reportedly disappeared fund worth
     went missing',
      'the pair worked bank china northern city harbin south china morning post
reported']
```

1.3 Feature Representation and Sentence Embeddings

[ ]:
```
from sklearn.feature_extraction.text import CountVectorizer, TfidfVectorizer
from gensim.models.word2vec import Word2Vec
from gensim.models import FastText
```

1.3.1 Bag-of-Words

[ ]:
```
count = CountVectorizer()
count_matrix = count.fit_transform(sentence_list).toarray()
count_matrix
```

[ ]:
```
array([[0, 0, 0, …, 0, 0, 0],
       [0, 0, 0, …, 0, 0, 0],
       [0, 0, 0, …, 0, 0, 0],
       …,
       [0, 0, 0, …, 0, 0, 0],
       [0, 0, 0, …, 0, 0, 0],
       [0, 0, 0, …, 0, 0, 1]], dtype=int64)
```

1.3.2 TF-IDF

[ ]:
```
tfidf = TfidfVectorizer()
tfidf_matrix = tfidf.fit_transform(sentence_list).toarray()
tfidf_matrix
```

[ ]:
```
array([[0.        , 0.        , 0.        , …, 0.        , 0.        ,
        0.        ],
       [0.        , 0.        , 0.        , …, 0.        , 0.        ,
        0.        ],
       [0.        , 0.        , 0.        , …, 0.        , 0.        ,
        0.        ],
       …,
       [0.        , 0.        , 0.        , …, 0.        , 0.        ,
        0.        ],
       [0.        , 0.        , 0.        , …, 0.        , 0.        ,
        0.        ],
       [0.        , 0.        , 0.        , …, 0.        , 0.        ,
        0.1756473]])
```

1.3.3 Word2Vec - CBoW

[ ]:
```
cbow = Word2Vec(sentences, vector_size=100, window=5, min_count=2, sg=0)
```

```
[ ]: vocab = cbow.wv.index_to_key
     def get_mean_vector(model, sentence):
         words = [word for word in sentence if word in vocab]
         if len(words) >= 1:
             return np.mean(model.wv[words], axis=0)
         return np.zeros((100,))
```

```
[ ]: cbow_array = []
     for sentence in sentences:
         mean_vec = get_mean_vector(cbow, sentence)
         cbow_array.append(mean_vec)
     cbow_array = np.array(cbow_array)
     cbow_array
```

```
[ ]: array([[-0.00123678,  0.00456679,  0.00109194, …, -0.00019224,
             -0.00318768,  0.00339288],
            [ 0.00110641,  0.00231303,  0.00055197, …, -0.00157737,
             -0.00280789,  0.00393946],
            [-0.00237076,  0.00180061,  0.0021589 , …, -0.00398201,
              0.00012388,  0.002603  ],
            …,
            [-0.0018367 , -0.0055438 ,  0.0039342 , …, -0.00477992,
             -0.00099786, -0.00261864],
            [-0.00880274,  0.00729735,  0.00306417, …, -0.00090636,
             -0.00204289, -0.00218901],
            [-0.00066024,  0.00045431,  0.0016595 , …,  0.00169176,
             -0.00251216,  0.00289475]])
```

1.3.4 Word2Vec - Skipgram

```
[ ]: sg = Word2Vec(sentences, vector_size=100, window=5, min_count=2, sg=0)
```

```
[ ]: vocab = sg.wv.index_to_key

     sg_array = []
     for sentence in sentences:
         mean_vec = get_mean_vector(sg, sentence)
         sg_array.append(mean_vec)
     sg_array = np.array(sg_array)

     sg_array
```

```
[ ]: array([[-0.00123678,  0.00456679,  0.00109194, …, -0.00019224,
             -0.00318768,  0.00339288],
            [ 0.00110641,  0.00231303,  0.00055197, …, -0.00157737,
             -0.00280789,  0.00393946],
            [-0.00237076,  0.00180061,  0.0021589 , …, -0.00398201,
              0.00012388,  0.002603  ],
            …,
            [-0.0018367 , -0.0055438 ,  0.0039342 , …, -0.00477992,
             -0.00099786, -0.00261864],
            [-0.00880274,  0.00729735,  0.00306417, …, -0.00090636,
             -0.00204289, -0.00218901],
            [-0.00066024,  0.00045431,  0.0016595 , …,  0.00169176,
             -0.00251216,  0.00289475]])
```

1.3.5 GloVe

```
[ ]: # from gensim.scripts.glove2word2vec import glove2word2vec
     # glove_file = 'glove.6B.100d.txt'
     # word2vec_file = 'glove.6B.100d.txt.word2vec'
     # glove2word2vec(glove_file, word2vec_file)
```

```
[ ]: from gensim.models import KeyedVectors
     file_name = "glove.6B.100d.txt.word2vec"
     model = KeyedVectors.load_word2vec_format(file_name, binary=False)
```

```
[ ]: glove_vocab = model.key_to_index

     def glove_mean_vector(model, words):
         words = [word for word in words if word in glove_vocab]
         if len(words) >= 1:
             return np.mean(model[words], axis=0)
         else:
             return []
```

```
[ ]: glove_array = []
     for sentence in sentences:
         mean_vec = glove_mean_vector(model, sentence)
         glove_array.append(mean_vec)
     glove_array = np.array(glove_array)

     glove_array
```

```
[ ]: array([[ 0.6178975 ,  0.43417996,  0.629341  , …, -0.22204556,
              0.39954   , -0.54206747],
            [ 0.28540468,  0.06498508,  0.24460083, …, -0.13884966,
              0.43221498, -0.06643867],
            [-0.01027839, -0.10825562,  0.39339715, …, -0.18146414,
              0.58340615, -0.01500285],
            …,
            [-0.1004313 ,  0.26319918,  0.00782941, …, -0.20854758,
              0.5661321 , -0.2002395 ],
            [ 0.11928448,  0.230349  ,  0.23061863, …,  0.13888137,
         0.6556625 , -0.13524412],
       [ 0.17414759, -0.03368236,  0.19662355, …, -0.28877276,
         0.5077956 ,  0.02789764]], dtype=float32)
```

1.3.6 FastText

```
[ ]: fasttext = FastText(sentences, sg=1, workers=4, vector_size=100, min_count=2,␣
      ↪window=5)
     # fasttext = Word2Vec.load("fasttext.model")
```

```
[ ]: fasttext_array = []
     for sentence in sentences:
         mean_vec = get_mean_vector(fasttext, sentence)
         fasttext_array.append(mean_vec)
     fasttext_array = np.array(fasttext_array)

     fasttext_array
```

```
[ ]: array([[-5.87673800e-04, -1.03293487e-03, -7.27967999e-04, …,
             -6.96344941e-04, -6.81907462e-04,  7.37620227e-04],
            [-8.43850372e-04, -5.97053498e-04, -4.97252680e-04, …,
              1.65927282e-04, -7.79972528e-04, -6.06873800e-05],
            [-4.90592443e-04, -1.14417751e-03, -9.06919653e-04, …,
             -1.75552233e-03, -1.44209480e-07, -2.42642884e-04],
            …,
            [-6.64076186e-04,  5.73331432e-04, -6.77327625e-04, …,
              1.35922129e-03, -5.75772661e-04, -5.38189546e-04],
            [-2.06752843e-03,  3.42661602e-04,  6.95628056e-04, …,
             -4.53541375e-04, -2.29731595e-04,  1.65456056e-03],
            [-4.05871600e-04,  5.87398477e-04,  6.06821850e-04, …,
              1.42467528e-04,  2.69624521e-04, -6.32533629e-05]])
```

1.4 Semantic Similarity using Cosine Distance

```
[ ]: from sklearn.metrics.pairwise import cosine_similarity
     from sklearn.neighbors import NearestNeighbors
```

```
[ ]: def get_similar_sentences(n, embeddings, sent_index):
         neigh = NearestNeighbors(n_neighbors=n + 1, metric='cosine')
         neigh.fit(embeddings)

         dist, rank = neigh.kneighbors(embeddings[sent_index].reshape(1, -1))
         similar_sentences = [original_sentences[i] for i in rank[0]]
         print(f"Original Sentence: {similar_sentences[0]}\n")
         print(f"Top {n} Similar Sentences:")
         for i in range(len(similar_sentences)):
             if i != 0:
```

```
            print(f"{i}. Dist:{dist[0][i]:.4f} - {similar_sentences[i]}")

    return "\n".join(similar_sentences)
```

```
[ ]: count_summary = get_similar_sentences(10, count_matrix, 0)
```

```
Original Sentence: Millions go missing at China bank.

Top 10 Similar Sentences:
1. Dist:0.5000 - The other is China Construction Bank.
2. Dist:0.5149 - Bank of China is the country's biggest foreign exchange dealer,
while China Construction Bank is the largest deposit holder.
3. Dist:0.5286 - The latest scandal at Bank of China will do nothing to reassure
foreign investors that China's big four banks are ready for international
listings.
4. Dist:0.5636 - Although, he committed the offences whilst running Bank of
China in New York, Mr.Wang was head of China Construction Bank when the scandal
broke.
5. Dist:0.5670 - Two senior officials at one of China's top commercial banks
have reportedly disappeared after funds worth up to $120m (£64m) went missing.
6. Dist:0.5918 - Bank of China is one of two frontrunners in the race to list
overseas.
7. Dist:0.6127 - The pair both worked at Bank of China in the northern city of
Harbin, the South China Morning Post reported.
8. Dist:0.6250 - China's leaders see reforming the top four banks as vital to
distribute capital to profitable companies and protect the health of China's
economic boom.
9. Dist:0.6985 - Earlier this month, a China Construction Bank branch manager
was jailed for life in a separate case.
10. Dist:0.6985 - China's banks used to act as cash offices for state
enterprises and did not require checks on credit worthiness.
```

```
[ ]: tfidf_summary = get_similar_sentences(10, tfidf_matrix, 0)
```

```
Original Sentence: Millions go missing at China bank.

Top 10 Similar Sentences:
1. Dist:0.7598 - Two senior officials at one of China's top commercial banks
have reportedly disappeared after funds worth up to $120m (£64m) went missing.
2. Dist:0.8221 - The other is China Construction Bank.
3. Dist:0.8646 - The latest scandal at Bank of China will do nothing to reassure
foreign investors that China's big four banks are ready for international
listings.
4. Dist:0.8647 - Bank of China is the country's biggest foreign exchange dealer,
while China Construction Bank is the largest deposit holder.
5. Dist:0.8835 - Although, he committed the offences whilst running Bank of
China in New York, Mr.Wang was head of China Construction Bank when the scandal
broke.
```

```
6. Dist:0.8968 - Bank of China is one of two frontrunners in the race to list
overseas.
7. Dist:0.9014 - The pair both worked at Bank of China in the northern city of
Harbin, the South China Morning Post reported.
8. Dist:0.9048 - China's leaders see reforming the top four banks as vital to
distribute capital to profitable companies and protect the health of China's
economic boom.
9. Dist:0.9285 - Earlier this month, a China Construction Bank branch manager
was jailed for life in a separate case.
10. Dist:0.9307 - China's banks used to act as cash offices for state
enterprises and did not require checks on credit worthiness.
```

[ ]:

```
cbow_summary = get_similar_sentences(10, cbow_array, 0)
```

```
Original Sentence: Millions go missing at China bank.

Top 10 Similar Sentences:
1. Dist:0.1553 - Two senior officials at one of China's top commercial banks
have reportedly disappeared after funds worth up to $120m (£64m) went missing.
2. Dist:0.2676 - The pair both worked at Bank of China in the northern city of
Harbin, the South China Morning Post reported.
3. Dist:0.3506 - Bank of China is one of two frontrunners in the race to list
overseas.
4. Dist:0.3676 - The latest scandal at Bank of China will do nothing to reassure
foreign investors that China's big four banks are ready for international
listings.
5. Dist:0.3794 - Although, he committed the offences whilst running Bank of
China in New York, Mr.Wang was head of China Construction Bank when the scandal
broke.
6. Dist:0.3817 - China's leaders see reforming the top four banks as vital to
distribute capital to profitable companies and protect the health of China's
economic boom.
7. Dist:0.3939 - China's banks used to act as cash offices for state enterprises
and did not require checks on credit worthiness.
8. Dist:0.3967 - Bank of China is the country's biggest foreign exchange dealer,
while China Construction Bank is the largest deposit holder.
9. Dist:0.4679 - The other is China Construction Bank.
10. Dist:0.5259 - However, a report in the China-published Economic Observer
said on Monday that the two banks may have scrapped plans to list in New York
because of the cost of meeting regulatory requirements imposed since the Enron
scandal.
```

[ ]:

```
sg_summary = get_similar_sentences(10, sg_array, 0)
```

```
Original Sentence: Millions go missing at China bank.

Top 10 Similar Sentences:
1. Dist:0.1553 - Two senior officials at one of China's top commercial banks
have reportedly disappeared after funds worth up to $120m (£64m) went missing.
```

```
2. Dist:0.2676 - The pair both worked at Bank of China in the northern city of
Harbin, the South China Morning Post reported.
3. Dist:0.3506 - Bank of China is one of two frontrunners in the race to list
overseas.
4. Dist:0.3676 - The latest scandal at Bank of China will do nothing to reassure
foreign investors that China's big four banks are ready for international
listings.
5. Dist:0.3794 - Although, he committed the offences whilst running Bank of
China in New York, Mr.Wang was head of China Construction Bank when the scandal
broke.
6. Dist:0.3817 - China's leaders see reforming the top four banks as vital to
distribute capital to profitable companies and protect the health of China's
economic boom.
7. Dist:0.3939 - China's banks used to act as cash offices for state enterprises
and did not require checks on credit worthiness.
8. Dist:0.3967 - Bank of China is the country's biggest foreign exchange dealer,
while China Construction Bank is the largest deposit holder.
9. Dist:0.4679 - The other is China Construction Bank.
10. Dist:0.5259 - However, a report in the China-published Economic Observer
said on Monday that the two banks may have scrapped plans to list in New York
because of the cost of meeting regulatory requirements imposed since the Enron
scandal.
```

```
[ ]: glove_summary = get_similar_sentences(10, glove_array, 0)
```

```
Original Sentence: Millions go missing at China bank.

Top 10 Similar Sentences:
1. Dist:0.0828 - Two senior officials at one of China's top commercial banks
have reportedly disappeared after funds worth up to $120m (£64m) went missing.
2. Dist:0.1061 - Bank of China is the country's biggest foreign exchange dealer,
while China Construction Bank is the largest deposit holder.
3. Dist:0.1150 - The other is China Construction Bank.
4. Dist:0.1208 - The latest scandal at Bank of China will do nothing to reassure
foreign investors that China's big four banks are ready for international
listings.
5. Dist:0.1358 - Earlier this month, a China Construction Bank branch manager
was jailed for life in a separate case.
6. Dist:0.1395 - Although, he committed the offences whilst running Bank of
China in New York, Mr.Wang was head of China Construction Bank when the scandal
broke.
7. Dist:0.1459 - The pair both worked at Bank of China in the northern city of
Harbin, the South China Morning Post reported.
8. Dist:0.1653 - Bank of China is one of two frontrunners in the race to list
overseas.
9. Dist:0.1681 - China's banking sector is burdened with at least $190bn of bad
debt according to official data, though most observers believe the true figure
is far higher.
```

```
10. Dist:0.1706 - China's leaders see reforming the top four banks as vital to
distribute capital to profitable companies and protect the health of China's
economic boom.
```

```
[ ]: fasttext_summary = get_similar_sentences(10, fasttext_array, 0)
```

```
Original Sentence: Millions go missing at China bank.

Top 10 Similar Sentences:
1. Dist:0.0995 - Two senior officials at one of China's top commercial banks
have reportedly disappeared after funds worth up to $120m (£64m) went missing.
2. Dist:0.2317 - Bank of China is the country's biggest foreign exchange dealer,
while China Construction Bank is the largest deposit holder.
3. Dist:0.3057 - China's banks used to act as cash offices for state enterprises
and did not require checks on credit worthiness.
4. Dist:0.3426 - China's leaders see reforming the top four banks as vital to
distribute capital to profitable companies and protect the health of China's
economic boom.
5. Dist:0.3772 - The pair both worked at Bank of China in the northern city of
Harbin, the South China Morning Post reported.
6. Dist:0.3959 - Bank of China is one of two frontrunners in the race to list
overseas.
7. Dist:0.3976 - The latest scandal at Bank of China will do nothing to reassure
foreign investors that China's big four banks are ready for international
listings.
8. Dist:0.4051 - Although, he committed the offences whilst running Bank of
China in New York, Mr.Wang was head of China Construction Bank when the scandal
broke.
9. Dist:0.4719 - Last year, China's premier, Wen Jiabao, targeted bank lending
practices in a series of speeches, and regulators ordered all big loans to be
scrutinized, in an attempt to cool down irresponsible lending.
10. Dist:0.4726 - Earlier this month, a China Construction Bank branch manager
was jailed for life in a separate case.
```

1.5 Explain the impact of embedding techniques in identifying the distance between sentences.

- Traditional methods (BoW, TF-IDF) consider two sentences are similar only if there are words co-occuring in both the sentences and doesn't consider the semantic similarity.
- Neural-Network based methods, (Word2Vec, GloVe, FastText) considers two sentences are similar even if there aren't many words co-occuring, yet are semantically similar.

1.6 Which Embedding technique generates contextual representations? Justify.

- **BERT**, **ELMo**, and other **transformer-based** models generate *contextual* representations.
- Although, Word2Vec, GloVe, FastText captures the semantics of the words, they only provides a single, context-independent embedding vector, for each word. Hence **STATIC** in

application. This limits the capacity of capturing the meaning of a word in two different contexts. Eg: ‘river bank ’and ‘bank deposit’.

- Whereas, BERT generates *different* output vectors for a *same* word when used in different context. Representations produced by BERT for ‘bank’ in river bank will be different than ‘bank’ in bank deposit. Word vectors produced by BERT are contextual and depend on the current input sentence. Hence **CONTEXT SENSITIVE.**