HU Extension --- Final Project --- S89A DL for NLP

Michael Lee & Micah Nickerson

PART 2B - ADVERSARIAL ATTACK GENERATOR

This is a notebook used to create the different adversarial attack **word perturbations**.

In [25]:
```
adversarial_dir = "Data Sets/adversarial_asap"
test_set_file = adversarial_dir+"/valid_set_plus_ADVERSARIAL_ESSAYS-ML.xls"

#verify data paths
print(test_set_file)
```

```
/content/drive/Shared drives/CSCI S-89A - Group Project/Data Sets/adversarial_a
sap/valid_set_plus_ADVERSARIAL_ESSAYS-ML.xls
```

In [0]:
```
# Attack 1: Shuffling Words

#load excel into dataframe
test_set_shuffle = pd.read_excel(test_set_file, sheet_name='valid_set')

#remove empty n/a cells
test_set_shuffle = test_set_shuffle.drop(['domain2_predictionid'], axis=1)


for i in test_set_shuffle.index:
  words= test_set_shuffle.at[i, 'essay'].split()
  random.shuffle(words)
  new_sentence = ' '.join(words)
  test_set_shuffle.at[i,'essay'] = new_sentence

test_set_shuffle.to_excel(adversarial_dir+"/valid_set_plus_ADVERSARIAL_ESSAYS-SHU
```

Anchor: Library

In [0]:
```
# Attack 2a: Appending - "Library"

#load excel into dataframe
test_set_append = pd.read_excel(test_set_file, sheet_name='valid_set')

#remove empty n/a cells
test_set_append = test_set_append.drop(['domain2_predictionid'], axis=1)


for i in test_set_append.index:
  words= test_set_append.at[i, 'essay'].split()
  words.append("library")
  new_sentence = ' '.join(words)
  test_set_append.at[i,'essay'] = new_sentence

test_set_append.to_excel(adversarial_dir+"/valid_set_plus_ADVERSARIAL_ESSAYS-APPI
```

In [0]:
```
# Attack 3a: Progressive Overload - "Library"

#load excel into dataframe
test_set_progressive = pd.read_excel(test_set_file, sheet_name='valid_set')

#remove empty n/a cells
test_set_progressive = test_set_progressive.drop(['domain2_predictionid'], axis=1


for i in test_set_progressive.index:
  words= test_set_progressive.at[i, 'essay'].split()
  if i < 591:
    continue
  if i < 641:
    for x in range(0,i-590):
      words[x] = "library"
  new_sentence = ' '.join(words)
  test_set_progressive.at[i,'essay'] = new_sentence

test_set_progressive.to_excel(adversarial_dir+"/valid_set_plus_ADVERSARIAL_ESSAYS
```

In [0]:
```
# Attack 4a: Single Substitution - "Library"

#load excel into dataframe
test_set_single = pd.read_excel(test_set_file, sheet_name='valid_set')

#remove empty n/a cells
test_set_single = test_set_single.drop(['domain2_predictionid'], axis=1)


for i in test_set_single.index:
  words= test_set_single.at[i, 'essay'].split()
  if i < 591:
    continue
  if i < 641:
    words[i-591] = "library"
  new_sentence = ' '.join(words)
  test_set_single.at[i,'essay'] = new_sentence

test_set_single.to_excel(adversarial_dir+"/valid_set_plus_ADVERSARIAL_ESSAYS-SIN
```

In [0]:
```
# Attack 5a: Insertion of anchor in random locations

#load excel into dataframe
test_set_insertion = pd.read_excel(test_set_file, sheet_name='valid_set')

#remove empty n/a cells
test_set_insertion = test_set_insertion.drop(['domain2_predictionid'], axis=1)


for i in test_set_insertion.index:
  words= test_set_insertion.at[i, 'essay'].split()
  x = random.randint(0,len(words))
  words.insert(x, 'library')
  new_sentence = ' '.join(words)
  test_set_insertion.at[i,'essay'] = new_sentence

test_set_insertion.to_excel(adversarial_dir+"/valid_set_plus_ADVERSARIAL_ESSAYS-
```

Anchor: Censorship

In [0]:
```
# Attack 2b: Appending - "Censorship"

#load excel into dataframe
test_set_append = pd.read_excel(test_set_file, sheet_name='valid_set')

#remove empty n/a cells
test_set_append = test_set_append.drop(['domain2_predictionid'], axis=1)


for i in test_set_append.index:
  words= test_set_append.at[i, 'essay'].split()
  words.append("censorship")
  new_sentence = ' '.join(words)
  test_set_append.at[i,'essay'] = new_sentence

test_set_append.to_excel(adversarial_dir+"/valid_set_plus_ADVERSARIAL_ESSAYS-APPI
```

In [0]:
```
# Attack 3b: Progressive Overload - "Censorship"

#load excel into dataframe
test_set_progressive = pd.read_excel(test_set_file, sheet_name='valid_set')

#remove empty n/a cells
test_set_progressive = test_set_progressive.drop(['domain2_predictionid'], axis=1

for i in test_set_progressive.index:
  words= test_set_progressive.at[i, 'essay'].split()
  if i < 591:
    continue
  if i < 641:
    for x in range(0,i-590):
      words[x] = "censorship"
  new_sentence = ' '.join(words)
  test_set_progressive.at[i,'essay'] = new_sentence

test_set_progressive.to_excel(adversarial_dir+"/valid_set_plus_ADVERSARIAL_ESSAYS
```

In [0]:
```
# Attack 4b: Single Substitution - "Censorship"

#load excel into dataframe
test_set_single = pd.read_excel(test_set_file, sheet_name='valid_set')

#remove empty n/a cells
test_set_single = test_set_single.drop(['domain2_predictionid'], axis=1)


for i in test_set_single.index:
  words= test_set_single.at[i, 'essay'].split()
  if i < 591:
    continue
  if i < 641:
    words[i-591] = "censorship"
  new_sentence = ' '.join(words)
  test_set_single.at[i,'essay'] = new_sentence

test_set_single.to_excel(adversarial_dir+"/valid_set_plus_ADVERSARIAL_ESSAYS-SIN
```

In [0]:
```
# Attack 5b: Insertion of "censorship" in random locations

#load excel into dataframe
test_set_insertion = pd.read_excel(test_set_file, sheet_name='valid_set')

#remove empty n/a cells
test_set_insertion = test_set_insertion.drop(['domain2_predictionid'], axis=1)


for i in test_set_insertion.index:
  words= test_set_insertion.at[i, 'essay'].split()
  x = random.randint(0,len(words))
  words.insert(x, 'censorship')
  new_sentence = ' '.join(words)
  test_set_insertion.at[i,'essay'] = new_sentence

test_set_insertion.to_excel(adversarial_dir+"/valid_set_plus_ADVERSARIAL_ESSAYS-
```

Anchor: The

In [0]:
```
# Attack 2c: Appending - "The"

#load excel into dataframe
test_set_append = pd.read_excel(test_set_file, sheet_name='valid_set')

#remove empty n/a cells
test_set_append = test_set_append.drop(['domain2_predictionid'], axis=1)


for i in test_set_append.index:
  words= test_set_append.at[i, 'essay'].split()
  words.append("the")
  new_sentence = ' '.join(words)
  test_set_append.at[i,'essay'] = new_sentence

test_set_append.to_excel(adversarial_dir+"/valid_set_plus_ADVERSARIAL_ESSAYS-APPI
```

In [0]:
```
# Attack 3c: Progressive Overload - "The"

#load excel into dataframe
test_set_progressive = pd.read_excel(test_set_file, sheet_name='valid_set')

#remove empty n/a cells
test_set_progressive = test_set_progressive.drop(['domain2_predictionid'], axis=1

for i in test_set_progressive.index:
  words= test_set_progressive.at[i, 'essay'].split()
  if i < 591:
    continue
  if i < 641:
    for x in range(0,i-590):
      words[x] = "the"
  new_sentence = ' '.join(words)
  test_set_progressive.at[i,'essay'] = new_sentence

test_set_progressive.to_excel(adversarial_dir+"/valid_set_plus_ADVERSARIAL_ESSAYS
```

In [0]:
```
# Attack 4c: Single Substitution - "The"

#load excel into dataframe
test_set_single = pd.read_excel(test_set_file, sheet_name='valid_set')

#remove empty n/a cells
test_set_single = test_set_single.drop(['domain2_predictionid'], axis=1)


for i in test_set_single.index:
  words= test_set_single.at[i, 'essay'].split()
  if i < 591:
    continue
  if i < 641:
    words[i-591] = "censorship"
  new_sentence = ' '.join(words)
  test_set_single.at[i,'essay'] = new_sentence

test_set_single.to_excel(adversarial_dir+"/valid_set_plus_ADVERSARIAL_ESSAYS-SIN
```

In [0]:
```
# Attack 5c: Insertion of "the" in random locations

#load excel into dataframe
test_set_insertion = pd.read_excel(test_set_file, sheet_name='valid_set')

#remove empty n/a cells
test_set_insertion = test_set_insertion.drop(['domain2_predictionid'], axis=1)


for i in test_set_insertion.index:
  words= test_set_insertion.at[i, 'essay'].split()
  x = random.randint(0,len(words))
  words.insert(x, 'the')
  new_sentence = ' '.join(words)
  test_set_insertion.at[i,'essay'] = new_sentence

test_set_insertion.to_excel(adversarial_dir+"/valid_set_plus_ADVERSARIAL_ESSAYS-
```