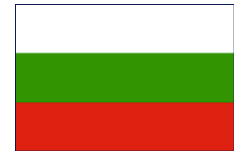
Lecture 10: HW3 Review, Classifying Text

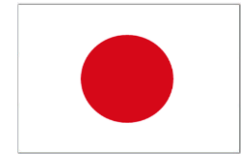
Ling 1330/2330 Computational Linguistics
Na-Rae Han, 9/28/2023

Objectives

- Homework 3 review
- Making assumptions: a cautionary tale
- Text classification; Naïve Bayes classifier
  - Language and Computers: Ch.5 Classifying documents
  - NLTK book: Ch.6 Learning to classify text

HW 3: Two EFL Corpora





Bulgarian Students

It is time, that our society is dominated by industrialization. The prosperity of a country is based on its enormous industrial corporations that are gradually replacing men with machines. Science is highly developed and controls the economy. From the beginning of school life students are expected to master a huge amount of scientific data. Technology is part of our everyday life.

Children nowadays prefer to play with computers rather than with our parents' wooden toys. But I think that in our modern world which worships science and technology there is still a place for dreams and imagination.

There has always been a place for them in man's life. Even in the darkness of the …

Japanese Students

I agree greatly this topic mainly because I think that English becomes an official language in the not too distant. Now, many people can speak English or study it all over the world, and so more people will be able to speak English. Before the Japanese fall behind other people, we should be able to speak English, therefore, we must study English not only junior high school students or over but also pupils. Japanese education system is changing such a program. In this way, Japan tries to internationalize rapidly. However, I think this way won't suffice for becoming international humans. To becoming international humans, we should study English not only school but also daily life. If we can do it, we are able to master English conversation. It is important for us to master English honorific words. …

Assessing writing quality

- Measurable indicators of writing quality

1. **Syntactic complexity**
   - Long, complex sentences vs. short. simple sentences

   ← Average sentence length, types of syntactic clauses used
2. **Lexical diversity**
   - Diverse vocabulary used vs. small set of words repeatedly used

   ← Type-token ratio (with caveat!) or other measures
3. **Vocabulary level**
   - Common, everyday words vs. sophisticated & technical words

   ← Average word length (common words tend to be shorter)

   ←% of word tokens in top 1K, 2K, 3K most common English words (Google Web 1T n-grams!)

Corpus size, essay length

	Bulgarian	Japanese
Corpus size (# of tokens)	17326	16094
Average # words/essay	578	536

Bulgarian students write longer essays.

Sentences

	Bulgarian	Japanese
Corpus size (# of tokens)	17326	16094
Average # words/essay	578	536
# of sentences	745	1027
Average # words/sent	23	16

Bulgarian students write longer sentences.

Types, TTR

	Bulgarian	Japanese
Corpus size (# of tokens)	17326	16094
Average # words/essay	578	536
# of sentences	745	1027
Average # words/sent	23	16
# of types	2584	1496
TTR	0.149	0.093

Bulgarian students have higher TTR

What about the CAVEAT?

Bulgarian student essays are LONGER!

Word length, k-band

	Bulgarian	Japanese
Corpus size (# of tokens)	17326	16094
Average # words/essay	578	536
# of sentences	745	1027
Average # words/sent	23	16
# of types	2584	1496
TTR	0.149	0.093
Average word length (# chars)	4.65	4.45
Average k-band	2.07	1.53
% 11+ k-band	11.69%	5.82%

Bulgarian students use longer words…

and in higher overall k-band.

Vocabulary bands: concept

- 'left' is 448th most common, 'usually' is 847th
  - → both belong to **1K vocabulary band**
- 'worry' is 5,005th most common, 'tired' is 5,678th
  - → both belong to **6K vocabulary band**

1k	2k	3k	5k	10k	15k	20k
we buy loan good green stories kids shopping records forums	move coffee random detailed square parents anyone directly therefore	manner religion factory charges figures proper falls touch brain	skill mutual anytime shorts talks desired aspect filing towns	teaches declare creature earning fuzzy banana switched conversati ons	entertain empires padding problematic harmless validated undergo openly troop	mermaid pancreatic morale marvelous rulings virtues infect regimen plausible

Vocabulary bands

'I am very tired'
→ k-band values 1, 1, 1, 6
→ Average k-band: 2.25

'I am utterly exhausted'
→ k-band values 1, 1, 15, 15
→ Average k-band: 8

- Bulgarian:
  - 17,326 tokens, 2,584 types
  - ← 86.5% of tokens are in 20 bands
  - ← Average k-band value **2.07**
- Japanese:
  - 16,094 tokens, 1,496 types
  - ← 86.0% of tokens are in 20 bands
  - ← Average k-band value **1.53**

An average Bulgarian word is ranked at < 2,000.

An average Japanese word is ranked at < 1,500.

Higher-band word types used

302 types. examples:
['crying', 'shallow', 'archaeology', 'freak', 'tendency', 'convince', 'exploits', 'advises', 'sooner', 'swallow']

- Bulgarian:
  - 17,326 tokens, 2,584 types
  - ← 86.5% of tokens are in 20 bands
  - ← Average k-band value **2.07**
  - ← 11.69% of types come from 11+

87 types. examples:
['traveled', 'rude', 'interpreter', 'interviewing', 'unrestricted', 'interpreters', 'admire', 'converse', 'unpleasant', 'touching']

- Japanese:
  - 16,094 tokens, 1,496 types
  - ← 86.0% of tokens are in 20 bands
  - ← Average k-band value **1.53**
  - ← 5.82% of types come from 11+

Bulgarian vs. Japanese students

- Your thoughts?

- Bulgarian students write longer essays.
- Bulgarian students use longer sentences & longer words, show higher TTR (even with larger corpus size!), and higher average vocabulary band.

- Can we conclude that Bulgarian students write at a more advanced level than Japanese students?

- What else can we explore?

Unigram frequency

Bulgarian top 50 words:

('the', 808) (',', 768) ('.', 699) ('to', 510) ('of', 498) ('and', 456) ('in', 354) ('is', 339) ('a', 281) ('that', 250) ('it', 235) ('for', 193) ('not', 177) ('are', 167) ('they', 160) ('be', 143) ('have', 134) ('with', 123) ('-', 115) ('their', 110) ('i', 109) ('our', 105) ('we', 105) ('as', 102) ('one', 101) ('life', 99) ('"', 99) ('this', 96) ('people', 92) ('but', 81) ('"', 79) ('imagination', 77) ('university', 77) ('or', 74) ('you', 72) ('on', 67) ('world', 67) ('do', 66) ('all', 65) ('will', 65) ('has', 64) ('students', 60) ('which', 60) ('there', 60) ('what', 60) ('them', 56) ('so', 55) ('us', 55) ('more', 54) ('real', 51)

Japanese top 50 words:

('.', 997) ('english', 710) (',', 709) ('to', 590) ('i', 452) ('is', 413) ('the', 383) ('and', 323) ('in', 272) ('we', 271) ('a', 240) ('that', 203) ('of', 202) ('it', 189) ('japanese', 181) ('language', 180) ('master', 167) ('think', 165) ('"', 163) ('people', 154) ('can', 151) ('for', 151) ('as', 136) ('if', 126) ('speak', 123) ('t', 119) ('but', 115) ('so', 114) ('not', 112) ('are', 111) ('have', 109) ('many', 103) ('be', 102) ('world', 93) ('need', 91) ('with', 90) ('second', 90) ('more', 79) ('study', 78) ('japan', 74) ('students', 73) ('foreign', 71) ('use', 70) ('they', 70) ('because', 69) ('very', 64) ('or', 62) ('will', 62) ('at', 62) ('by', 59)

Bigram frequency

Bulgarian top 50 bigrams:

(('of', 'the'), 98), (('it', 'is'), 75), (('in', 'the'), 72), (('.', 'the'), 68), (('to', 'be'), 48), (('for', 'the'), 46), (("'", 's'), 44), ((',', 'the'), 42), (('.', 'it'), 40), ((',', 'it'), 40), (('do', 'not'), 38), (('to', 'the'), 37), (('is', 'the'), 35), ((',', 'and'), 33), (('life', '.'), 30), (('they', 'are'), 30), (('there', 'is'), 27), (('.', 'they'), 27), (('with', 'the'), 27), (('.', 'in'), 27), (('in', 'a'), 26), (('is', 'not'), 25), ((',', 'they'), 25), (('on', 'the'), 24), (('.', 'but'), 23), (('in', 'our'), 23), (('university', 'degrees'), 23), ((',', 'to'), 23), (('.', 'i'), 22), (('of', 'our'), 21), (('imagination', '.'), 21), (('real', 'life'), 21), (('to', 'do'), 21), ((',', 'in'), 21), (('it', '.'), 20), (('the', 'university'), 20), (('the', 'real'), 20), (('.', 'we'), 20), (('this', 'is'), 20), (('should', 'be'), 20), (('the', 'fact'), 19), (('fact', 'that'), 19), (('real', 'world'), 19), (('is', 'a'), 18), (('that', 'the'), 18), (('in', 'their'), 18), ((',', 'which'), 18), (('as', 'a'), 18), (('and', 'the'), 18), (('their', 'own'), 18)

Japanese top 50 bigrams:

(('.', 'i'), 152), (('i', 'think'), 137), (('master', 'english'), 128), (('english', 'is'), 125), (('english', '.'), 124), (('to', 'master'), 121), (("'", 't'), 119), (('it', 'is'), 110), ((',', 'i'), 90), (('speak', 'english'), 87), (('english', ','), 82), (('second', 'language'), 81), ((',', 'we'), 78), (('english', 'as'), 78), (('as', 'a'), 77), (('the', 'world'), 76), (('need', 'to'), 74), (('in', 'the'), 71), (('language', '.'), 70), (('a', 'second'), 68), (('.', 'if'), 59), (('think', 'that'), 58), (('if', 'we'), 57), (('japanese', 'students'), 56), (('we', 'can'), 54), (('.', 'so'), 51), (('.', 'and'), 50), (('.', 'but'), 49), (('.', 'it'), 46), ((',', 'and'), 43), (('able', 'to'), 43), (('want', 'to'), 42), (('study', 'english'), 40), (('.', 'in'), 40), (('can', "'"), 40), (('world', '.'), 40), (('use', 'english'), 38), (('is', 'very'), 38), (('to', 'speak'), 37), ((',', 'english'), 36), (('in', 'english'), 36), (('communicate', 'with'), 34), (('have', 'to'), 34), (('there', 'are'), 34), (('students', 'need'), 33), (('in', 'japan'), 33), (('don', "'"), 33), (("'", 's'), 33), (('to', 'study'), 32), ((',', 'it'), 31)

N-grams and writing quality

- Beyond impressionistic observations, can we obtain more principled sort of n-gram based measurements?
- YES! We can, and we should!

From the beginning of school life students are expected to master a huge amount of scientific data.

I agree greatly this topic mainly because I think that English becomes an official language in the not too distant.

- We can estimate the probability of these sentences in native-speaker-written English, through n-gram language modeling.
  - Where to get native-speaker n-gram conditional probability?

    ← Google 1T n-gram, COCA…

More Advanced: Consider Individual essays

- We computed **one number across a whole group**: BU/JA
- We could go instead with **per-essay measurements**:
  - 30 & 30 figures!
  - We can get a sense of distribution, range, average, standard deviation, etc.
  - We could test for statistical significance!
  - Graphs, visualization!

- This is best done in **Jupyter Notebook**, using the **pandas** library.
- A quick review: work by Eva Bacas (past student)
  - On Canvas, "Modules" page!

Worksheet

1. Anything interesting you discovered about the BU/JA corpora?

2. What are the main findings of Eva Bacas's analysis of BU/JA corpora?

3. Comparing Norvig/Google unigrams vs. Enable word list. What is your prediction?

4. (Continued) How many word types do they actually have in common?

Worksheet

1. Anything interesting you discovered about the BU/JA corpora?

2. What are the main findings of Eva Bacas's analysis of BU/JA corpora?

3. Comparing Norvig/Google unigrams vs. Enable word list. What is your prediction?

4. (Continued) How many word types do they actually have in common?

Making assumptions: a CAUTIONARY TALE

- About datasets we downloaded from the web
- About text processing output we just built

- … these assumptions just might bite you.

1-grams/word list: Norvig vs. ENABLE

- count_1w.txt

```
the       23135851162
of        13151942776
and       12997637966
to        12136980858
a         9081174698
in        8469404971
for       5933321709
is        4705743816
on        3750423199
that      3400031103
by        3350048871
this      3228469771
.
.
.
goofer    12711
gooek     12711
gooddg    12711
gooblle   12711
gollgo    12711
golgw     12711
```

goog1w_fd.pkl

Total # of entries:
← 333K
vs.
173K →

Assumption:
Norvig/Google word list is a superset of the ENABLE list

- enable1.txt

```
aa
aah
aahed
aahing
aahs
aal
aalii
aaliis
aals
aardvark
aardvarks
aardwolf
aardwolves
.
.
.
zymotic
zymurgies
zymurgy
zyzzyva
zyzzyvas
```

words.pkl

1-grams/word list: Norvig vs. ENABLE

- count_1w.txt

```
the       23135851162
of        13151942776
and       12997637966
to        12136980858
a         9081174698
in        8469404971
for       5933321709
is        4705743816
on        3750423199
that      3400031103
by        3350048871
this      3228469771
.
.
.
goofer    12711
gooek     12711
gooddg    12711
gooblle   12711
gollgo    12711
golgw     12711
```

Total # of entries:
← 333K
vs.
173K →

Assumption:
Norvig/Google word list is a superset of the ENABLE list

WRONG!

- enable1.txt

```
aa
aah
aahed
aahing
aahs
aal
aalii
aaliis
aals
aardvark
aardvarks
aardwolf
aardwolves
.
.
.
zymotic
zymurgies
zymurgy
zyzzyva
zyzzyvas
```

1-grams/word list: Norvig vs. ENABLE

- count_1w.txt

```
the       23135851162
of        13151942776
and       12997637966
to        12136980858
a         9081174698
in        8469404971
for       5933321709
is        4705743816
on        3750423199
that      3400031103
by        3350048871
this      3228469771
...
goofer    12711
gooek     12711
gooddg    12711
gooblle   12711
gollgo    12711
golgw     12711
```

Total # of entries:
← 333K
vs.
173K →

Only 78,835 shared between (45.6% of ENABLE)

No single-char words ("I", "a"), lots of arcane words

TONS of proper nouns, many non-words

- enable1.txt

```
aa
aah
aahed
aahing
aahs
aal
aalii
aaliis
aals
aardvark
aardvarks
aardwolf
aardwolves
...
zymotic
zymurgies
zymurgy
zyzzyva
zyzzyvas
```

Know your data

- When using publicly available resources, you must *evaluate* and *understand* the data.
  - Origin? Purpose?
  - Domain & genre?
  - Size?
  - Traits?
  - Merits and limitations?
  - Fit with your project?

NLTK's corpus reader

```
>>> from nltk.corpus import PlaintextCorpusReader
>>> corpus_root = "C:/Users/narae/Documents/ling1330/MLK"
>>> mlkcor = PlaintextCorpusReader(corpus_root, '.*txt')
>>> type(mlkcor)
    <class 'nltk.corpus.reader.plaintext.PlaintextCorpusReader'>
>>> mlkcor.fileids()
    ['1963-I Have a Dream.txt', '1964-Nobel Peace Prize Acceptance Speech.txt',
    '1967-Beyond Vietnam.txt', "1968-I've been to the Mountain Top.txt"]
>>> len(mlkcor.fileids())
    4
>>> mlkcor.fileids()[0]
    '1963-I Have a Dream.txt'
>>> mlkcor.words()[:50]
    ['I', 'Have', 'A', 'Dream', 'by', 'Dr', '.', 'Martin', 'Luther', 'King',
    'Jr', '.', 'Delivered', 'on', 'the', 'steps', 'at', 'the', 'Lincoln',
    'Memorial', 'in', 'Washington', 'D', '.', 'C', '.', 'on', 'August', '28',
    ',', '1963', 'I', 'am', 'happy', 'to', 'join', 'with', 'you', 'today',
    'in', 'what', 'will', 'go', 'down', 'in', 'history', 'as', 'the',
    'greatest', 'demonstration']
```

Assumption:
`.word()` tokens will be tokenized the same way as `nltk.word_tokenize()`

WRONG!

NLTK's corpus reader

```
>>> mlkcor.words()[-50:]
    ['that', 'we', ',', 'as', 'a', 'people', 'will', 'get', 'to', 'the',
    'promised', 'land', '.', 'And', 'I', "'", 'm', 'happy', ',', 'tonight',
    '.', 'I', "'", 'm', 'not', 'worried', 'about', 'anything', '.', 'I', "'",
    'm', 'not', 'fearing', 'any', 'man', '.', 'Mine', 'eyes', 'have', 'seen',
    'the', 'glory', 'of', 'the', 'coming', 'of', 'the', 'Lord', '.']
>>> mlkcor.words()[-100:-50]
    ['God', "'", 's', 'will', '.', 'And', 'He', "'", 's', 'allowed', 'me',
    'to', 'go', 'up', 'to', 'the', 'mountain', '.', 'And', 'I', "'", 've',
    'looked', 'over', '.', 'And', 'I', "'", 've', 'seen', 'the', 'promised',
    'land', '.', 'I', 'may', 'not', 'get', 'there', 'with', 'you', '.', 'But',
    'I', 'want', 'you', 'to', 'know', 'tonight', ',']
```

By default, PlaintextCorpusReader uses a **regular-expression** based tokenizer, which **splits out all symbols** from alphabetic words.

HW#2: Basic corpus stats

The Bible

```
Word token count:   946,812
Word type count:     17,188
```

TTR: 0.018

Jane Austen novels

```
Word token count:   431,079
Word type count:     11,642
```

TTR: 0.027

Assumption:
These word types are all legitimate English words (and punctuation).

WRONG!

HW#2: Basic corpus stats

The Bible

```
Word token count:   946,812
Word type count:     17,188
```

TTR: 0.018

Jane Austen novels

```
Word token count:   431,079
Word type count:     11,642
```

TTR: 0.027

```
>>> b_type_nonalnum = [t for t in b_tokfd if not t.isalnum()]
>>> len(b_type_nonalnum)
    4628
>>> b_type_nonalnum[:30]
    ['[', ']', ':', '1:1', '.', '1:2', ',', ';', '1:3', '1:4', '1:5', '1:6',
    '1:7', '1:8', '1:9', '1:10', '1:11', '1:12', '1:13', '1:14', '1:15',
    '1:16', '1:17', '1:18', '1:19', '1:20', '1:21', '1:22', '1:23', '1:24']
```

Over ¼ (!!!) of Bible word types
are verse numberings,
vastly inflating type count & TTR.

Always validate, verify

- About text processing output we just built
  - Make sure to probe and verify.
  - Watch out for oddities.
  - Pre-built text processing functions are NOT perfect!
    - Sentence tokenization, word tokenization ➔ might include errors
    - They also might operate under different rules than you expect (check default setting).

- Especially important when attaching linguistic interpretation to your numbers.
  - Hidden factors might be affecting the numbers.

Automatic classification

- A ***classifier*** is an algorithm that processes a linguistic input and assigns it a **class** from a user-defined set.
  - It usually denotes a statistical model induced through machine learning.
  - The algorithm works off a set of weighted contextual features.

- What are examples of classifiers?

Example: name and gender

- Are the following first names **male** or **female**? Maybe **either**?
  - *James, William, Elizabeth, Hillary, Dana, McKayla, Taylor*
  - *Joffrey, Tyrion, Arya*
  - *Kimiko, Na-Rae, Tae-hyung, Jimin*
- With a novel name, a speaker of the language can still *guess* the gender, often with high accuracy
  - Guessing based on what?

  ← Based on the presence/absence of certain **observable features**.
- Boy or girl?
  - `J...a` `T...n` `N...e` `L...o`
- We can build a **model** that replicates the above human cognitive process.

Boy or girl? A Naïve-Bayes classifier

```
>>> boyorgirl = nltk.NaiveBayesClassifier.train(train_set)
>>> gender_features('Neo')
    {'first_letter': 'N', 'last_letter': 'o'}
>>> boyorgirl.classify(gender_features('Neo'))
    'male'
>>> boyorgirl.classify(gender_features('Arya'))
    'female'
>>> boyorgirl.classify(gender_features('Na-Rae'))
    'female'
>>> nltk.classify.accuracy(boyorgirl, test_set)
    0.768
```

- Classifier built from NLTK's `names` corpus (`nltk.corpus.names`)
- Uses two features: first letter, last letter
- See the NLTK book section for details:
  - https://www.nltk.org/book/ch06.html#gender-identification

Example: movie reviews

all of this , of course , leads up to the predictable climax . but as foreseeable as the ending is , it's so damn cute and well-done that i doubt any movie in the entire year contains a scene the evokes as much pure joy as this part does . when ryan discovers the true identity of her online love , i was filled with such , for lack of a better word , happiness that for the first time all year , i actually left the theater smiling .

the acting is below average , even from the likes of curtis .. sutherland is wasted and baldwin , well , he's acting like a baldwin , of course . the real star here are stan winston's robot design , some schnazzy cgi , and the occasional good gore shot , like picking into someone's brain . so , if robots and body parts really turn you on , here's your movie . otherwise , it's pretty much a sunken ship of a movie .

- Classify each document as "positive" or "negative"

  ← A type of **sentiment analysis**

- What "features" can we use?
  - Words themselves
  - N-grams, length, ...

Wrapping up

- Next class:
  - Continuing with Naïve Bayes classifier, evaluation
  - Language and Computers: Ch.5 Classifying documents
  - NLTK book: Ch.6 Learning to classify text

- Exercise 6 out
  - Take your time!