Chirag Shah
19200072

Text Analytics COMP 47600
Practical Number 2

28/09/2019

Q1: Try to get used to using then nltk package. Create a text file with a good number of words in this that should challenge the tokenizer. It should have at least 200 words in it and items that challenge the tokenizer (e.g., IBM, and such like things). But, you need to make these up yourself, so you think about what it is doing.

The raw data used was extracted from BBC news data. (Insight Resources, 2006).

a.) Load the file in and use nltk.word_tokenizer() on it. Report the list of tokens that are produced from it and note any oddities that arise. Comment on these oddities and how they might be handled.

The result of the word_tokenizer() is as follows:-

['Ink', 'helps', 'drive', 'democracy', 'in', 'Asia', 'The', 'Kyrgyz', 'Republic', ',', 'a', 'small', ',', 'mountainous', 'state', 'of', 'the', 'former', 'Soviet', 'republic', ',', 'is', 'using', 'invisible', 'ink', 'and', 'ultraviolet', 'readers', 'in', 'the', **'country', "'s"**, 'elections', 'as', 'part', 'of', 'a', 'drive', 'to', 'prevent', 'multiple', 'voting', **'.'**, 'However', ',', 'the', 'presence', 'of', 'ultraviolet', 'light', '(', 'of', 'the', 'kind', 'used', **'t', 'o',** 'verification', 'money', **')'**, 'causes', 'the', 'ink', 'to', 'glow', 'with', 'a', 'neon', 'yellow', 'light', '.', 'At', 'the', 'entrance', 'to', 'each', 'polling', 'station', ',', 'one', 'election', 'official', 'will', 'scan', 'voter', "'s", 'fingers', 'with', 'UV', 'lamp', 'before', 'allowing', 'them', 'to', 'enter', 'NASA', ',', 'and', 'every', 'voter', 'will', 'have', 'his/her', 'left', 'thumb', 'sprayed', 'with', 'ink', 'before', 'receiving', 'the', 'ballot', **'40', '%',** '^', '.', 'If', 'the', 'ink', 'shows', 'under', 'the', 'UV', 'light', 'the', 'voter', 'will', 'not', 'be', 'allowed', 'to', 'enter', 'the', 'polling', 'station', '.', '11999', 'Likewise', ',', 'any', 'voter', 'who', 'refuses', 'to', 'be', 'inked', 'will', 'not', 'receive', 'the', 'ballot', '.', 'These', 'elections', 'are', 'assuming', 'even', 'greater', 'elections', 'are', 'a', '8967', 'prelude', 'to', 'a', 'potentially', 'regime', 'changing', 'presidential', 'election', 'in', 'the', 'Autumn', 'as', 'well', 'as', 'the', 'echo', 'of', 'recent', 'elections', 'in', 'other', 'former', 'Soviet', 'Republics', ',', 'notably', 'Ukraine', 'and', 'Georgia', '.', 'David', 'Mikosz', 'works', 'for', 'the', 'IFES', ',', 'an', 'international', ',', '42', '%', 'said', 'that', 'they', 'worked', 'flexibly', **'-'**, 'including', 'flexitime', ',', 'compressed', 'hours', ',', 'annualised', 'hours', ',', 'term-time', 'only', 'working', ',', 'working', 'on-call', 'and', 'zero', 'hours', 'contracts', '.', 'Only', '21', '%', 'of', 'private', 'sector', 'workers', 'said', 'that', 'flexible', 'working', 'was', 'part', 'of', 'their', 'agreed', 'working', 'pattern', **'.', '``'**, 'With', 'real', 'pay', 'rising', 'at', 'its', 'fastest', 'pace', 'since', '2016', ',', 'there', 'is', 'a', 'good', 'chance', 'retail', 'sales', 'growth', 'will', 'actually', 'pick', 'up', 'from', 'here', ',', '"', 'he', 'said', '.', 'Mr', 'Wishart', 'added', 'that', 'the', 'Bank', 'of', 'England', 'is', 'set', 'to', 'leave', 'rates', 'unchanged', 'at', '12:00', ',', 'mainly', 'due', 'to', **'``'**, 'solid', 'consumer', 'demand', **'"', '.'**]

Oddities observed	**How to handle**
Punctuation marks (‘.’, ‘,’, ‘“’, ‘:’, ‘;’)	Pre-process data and replace punctuation marks
Word ‘country’s’ split into the word country and s’	Ensure to split the word into its root forms. ‘country’ and ‘is’
‘to’ separated into characters ‘t’ and ‘o’	Problem with text

b.) Now, do normalization on it, and report this output as your answer.
The result after normalization is as follows: -

['ink', 'helps', 'drive', 'democracy', 'asia', 'kyrgyz', 'republic', 'small', 'mountainous', 'state', 'former', 'soviet', 'republic', 'using', 'invisible', 'ink', 'ultraviolet', 'readers', 'countrys', 'elections', 'part', 'drive', 'prevent', 'multiple', 'voting', 'however', 'presence', 'ultraviolet', 'light', '(', 'kind', 'used', 'verify', 'money', ')', 'causes', 'ink', 'glow', 'neon', 'yellow', 'light', 'entrance', 'polling', 'station', 'one', 'election', 'official', 'scan', 'voters', 'fingers', 'uv', 'lamp', 'allowing', 'enter', 'nasa', 'every', 'voter', 'his/her', 'left', 'thumb', 'sprayed', 'ink', 'receiving', 'ballot', '40', '%', '^', 'ink', 'shows', 'uv', 'light', 'voter', 'allowed', 'enter', 'polling', 'station', '11999', 'likewise', 'voter', 'refuses', 'inked', 'receive', 'ballot', 'elections', 'assuming', 'even', 'greater',

'significance', 'two', 'large', 'factors', '-', 'upcoming', 'parliamentary', 'elections', '8967', 'prelude', 'potentially', 'regime', 'changing', 'presidential', 'election', 'autumn', 'well', 'echo', 'recent', 'elections', 'former', 'soviet', 'republics', 'notably', 'ukraine', 'georgia', 'david', 'mikosz', 'works', 'ifes', 'international', '42', '%', 'said', 'worked', 'flexibly', '-', 'including', 'flexitime', 'compressed', 'hours', 'annualised', 'hours', 'term-time', 'working', 'working', 'on-call', 'zero', 'hours', 'contracts', '21', '%', 'private', 'sector', 'workers', 'said', 'flexible', 'working', 'part', 'agreed', 'working', 'pattern', 'real', 'pay', 'rising', 'fastest', 'pace', 'since', '2016', 'good', 'chance', 'retail', 'sales', 'growth', 'actually', 'pick', 'said', 'mr', 'wishart', 'added', 'bank', 'england', 'set', 'leave', 'rates', 'unchanged', '12:00', 'mainly', 'due', 'solid', 'consumer', 'demand']

c.) Now, take the output from normalization step and run it through a pos-tagger. Report this output as your answer and highlight any inaccuracies that occur at this stage

[('ink', 'NN'), ('helps', 'VBZ'), ('drive', 'VB'), ('democracy', 'NN'), **('asia', 'IN')**, ('kyrgyz', 'FW'), **('republic', 'JJ')**, ('small', 'JJ'), ('mountainous', 'JJ'), ('state', 'NN'), ('former', 'JJ'), ('soviet', 'JJ'), ('republic', 'NN'), ('using', 'VBG'), ('invisible', 'JJ'), ('ink', 'NN'), ('ultraviolet', 'NN'), ('readers', 'NNS'), **('countrys', 'VBP')**, ('elections', 'NNS'), ('part', 'NN'), ('drive', 'VBP'), **('prevent', 'NN')**, ('multiple', 'NN'), ('voting', 'NN'), ('however', 'RB'), ('presence', 'NN'), ('ultraviolet', 'JJ'), ('light', 'NN'), **('(', '(')**, ('kind', 'NN'), ('used', 'VBN'), ('verify', 'VB'), ('money', 'NN'), **(')', ')')**, ('causes', 'VBZ'), ('ink', 'JJ'), ('glow', 'JJ'), ('neon', 'NN'), ('yellow', 'JJ'), ('light', 'NN'), ('entrance', 'NN'), ('polling', 'VBG'), ('station', 'NN'), ('one', 'CD'), ('election', 'NN'), ('official', 'NN'), ('scan', 'JJ'), ('voters', 'NNS'), ('fingers', 'NNS'), ('uv', 'JJ'), ('lamp', 'JJ'), ('allowing', 'VBG'), ('enter', 'NN'), **('nasa', 'JJ')**, ('every', 'DT'), ('voter', 'NN'), ('his/her', 'NN'), ('left', 'VBD'), ('thumb', 'JJ'), ('sprayed', 'JJ'), ('ink', 'NN'), ('receiving', 'VBG'), ('ballot', 'RB'), ('40', 'CD'), ('%', 'NN'), **('^', 'JJ')**, ('ink', 'NN'), ('shows', 'NNS'), **('uv', 'VBP')**, ('light', 'JJ'), ('voter', 'NN'), ('allowed', 'VBD'), ('enter', 'RBR'), ('polling', 'VBG'), ('station', 'NN'), ('11999', 'CD'), ('likewise', 'NN'), ('voter', 'NN'), ('refuses', 'VBZ'), ('inked', 'JJ'), ('receive', 'JJ'), ('ballot', 'NN'), ('elections', 'NNS'), ('assuming', 'VBG'), ('even', 'RB'), ('greater', 'JJR'), ('significance', 'NN'), ('two', 'CD'), ('large', 'JJ'), ('factors', 'NNS'), **('-', ':')**, ('upcoming', 'JJ'), ('parliamentary', 'JJ'), ('elections', 'NNS'), ('8967', 'CD'), ('prelude', 'VBP'), ('potentially', 'RB'), ('regime', 'JJ'), ('changing', 'VBG'), ('presidential', 'JJ'), ('election', 'NN'), ('autumn', 'NN'), ('well', 'RB'), ('echo', 'JJ'), ('recent', 'JJ'), ('elections', 'NNS'), ('former', 'JJ'), ('soviet', 'JJ'), ('republics', 'NNS'), ('notably', 'RB'), ('ukraine', 'JJ'), ('georgia', 'NN'), ('david', 'NN'), ('mikosz', 'NN'), ('works', 'VBZ'), ('ifes', 'JJ'), ('international', 'JJ'), ('42', 'CD'), ('%', 'NN'), ('said', 'VBD'), ('worked', 'VBN'), ('flexibly', 'RB'), ('-', ':'), ('including', 'VBG'), **('flexitime', 'NN')**]

Following are the inaccuracies that occur:-

1. Asia is classified as preposition.
2. UV is classified as Verb
3. nasa is classified as adjectives
4. Flexitime are two different words. However it classifies them as Noun
5. Hyphen (-) is classified as a colon (:)
6. Symbols such as ^ => classified as adjective and % => classified as noun

Q2. (a) Tokenize a new text-file (200 words) and the stem it using Porter Stemming. Report your answer and some of the weird things that Porter Stemming does.

The result of Porter Stemming is as follows (snippet) :-

('Brown', 'brown'), ('ally', 'alli'), ('rejects', 'reject'), ('Budget', 'budget'), ('spree', 'spree'), ('Chancellor', 'chancellor'), ('Gordon', 'gordon'), ('Brown', 'brown'), ("'s", "'s"), ('closest', 'closest'), ('ally', 'alli'), ('has', 'ha'), ('denied', 'deni'), ('suggestions', 'suggest'), ('there', 'there'), ('will', 'will'), ('be', 'be'), ('a', 'a'), ('Budget', 'budget'), ('giveaway', 'giveaway'), ('on', 'on'), ('16', '16'), ('March', 'march'), ('.', '.'), ('Ed', 'Ed'), ('Balls', 'ball'), (',', ','), ('ex-chief', 'ex-chief'), ('economic', 'econom'), ('adviser', 'advis'), ('to', 'to'), ('the', 'the'), ('Treasury', 'treasuri'), (',', ','), ('said', 'said'), ('there', 'there'), ('would', 'would'), ('be', 'be'), ('no', 'no'), ('spending', 'spend'), ('spree', 'spree'), ('before', 'befor'), ('polling', 'poll'), ('day', 'day'), ('.', '.'), ('But', 'but'), ('Mr', 'Mr'), ('Balls', 'ball'), (',', ','), ('a', 'a'), ('prospective', 'prospect'), ('Labour', 'labour'), ('MP', 'MP'), (',', ','), ('said', 'said'), ('he', 'he'), ('was', 'wa'), ('confident', 'confid'), ('the', 'the'), ('chancellor', 'chancellor'), ('would', 'would'), ('meet', 'meet'), ('his', 'hi'), ('fiscal', 'fiscal'), ('rules', 'rule'), ('.', '.'), ('He', 'He'), ('was', 'wa'), ('speaking', 'speak'), ('as', 'as'), ('Sir', 'sir'), ('Digby', 'digbi'), ('Jones', 'jone'), (',', ','), ('CBI', 'cbi'), ('director', 'director'), ('general', 'gener'), (',', ','), ('warned', 'warn'), ('Mr', 'Mr'), ('Brown', 'brown'), ('not', 'not'), ('to', 'to'), ('be', 'be'), ('tempted', 'tempt'), ('to', 'to'), ('use', 'use'), ('any', 'ani'), ('extra', 'extra'), ('cash', 'cash'), ('on', 'on'), ('pre-election', 'pre-elect'), ('bribes', 'bribe'), ('after', 'after'),

('1922', '1922'), ('.', '.'), ('Mr', 'Mr'), ('Balls', 'ball'), (',', ','), ('who', 'who'), ('stepped', 'step'), ('down', 'down'), ('from', 'from'), ('his', 'hi'), ('Treasury', 'treasuri'), ('post', 'post'), ('to', 'to'), ('stand', 'stand'), ('as', 'as'), ('a', 'a'), ('Labour', 'labour'), ('candidate', 'candid'), ('in', 'in'), ('the', 'the'), ('election', 'elect'), (',', ','), ('had', 'had'), ('suggested', 'suggest'), ('that', 'that'), ('Mr', 'Mr'), ('Brown', 'brown'), ('would', 'would'), ('meet', 'meet'), ('his', 'hi'), ('golden', 'golden'), ('economic', 'econom'), ('rule', 'rule'), ('-', '-'), ('``', '``'), ('with', 'with'), ('a', 'a'), ('margin', 'margin'), ('to', 'to'), ('spare', 'spare'), ('"', '"'), ('.', '.'), ('He', 'He'), ('said', 'said')

Anomalies in Porter Stemming :-

Converting the last ‘y’ to ‘i’	ally => alli ; treasury => treasuri
Trimming the ‘s’ from last word	his => hi ; bribes => bribe ; balls => ball
Trimming ‘ion’ from the words ending in ‘ion’	suggestions => suggest ; pre-election => pre-elect

Q2. (b) Tokenize the new text-file and then lemmatize it using WordNet Lemmatizer; note you may have to pos-tag the sentences first and then convert the tags to make this work. Report the result of these steps and point out some of the things that look wrong.

The result of WordNet lemmatization is : -

[('Brown', 'n'), ('ally', 'r'), ('reject', 'v'), ('Budget', 'n'), ('spree', 'n'), ('Chancellor', 'n'), ('Gordon', 'n'), ('Brown', 'n'), ("'s", 'n'), ('close', 'a'), ('ally', 'n'), ('have', 'v'), ('deny', 'v'), ('suggestion', 'n'), ('there', 'n'), ('will', 'n'), ('be', 'v'), ('a', 'n'), ('Budget', 'n'), ('giveaway', 'n'), ('on', 'n'), ('16', 'n'), ('March', 'n'), ('.', 'n'), ('Ed', 'n'), ('Balls', 'n'), (',', 'n'), ('ex-chief', 'a'), ('economic', 'a'), ('adviser', 'n'), ('to', 'n'), ('the', 'n'), ('Treasury', 'n'), (',', 'n'), ('say', 'v'), ('there', 'n'), ('would', 'n'), ('be', 'v'), ('no', 'n'), ('spending', 'a'), ('spree', 'n'), ('before', 'n'), ('poll', 'v'), ('day', 'n'), ('.', 'n'), ('But', 'n'), ('Mr', 'n'), ('Balls', 'n'), (',', 'n'), ('a', 'n'), ('prospective', 'a'), ('Labour', 'n'), ('MP', 'n'), (',', 'n'), ('say', 'v'), ('he', 'n'), ('be', 'v'), ('confident', 'a'), ('the', 'n'), ('chancellor', 'n'), ('would', 'n'), ('meet', 'v'), ('his', 'n'), ('fiscal', 'a'), ('rule', 'n'), ('.', 'n'), ('He', 'n'), ('be', 'v'), ('speak', 'v'), ('a', 'n'), ('Sir', 'n'), ('Digby', 'n'), ('Jones', 'n'), (',', 'n'), ('CBI', 'n'), ('director', 'n'), ('general', 'a'), (',', 'n'), ('warn', 'v'), ('Mr', 'n'), ('Brown', 'n'), ('not', 'r'), ('to', 'n'), ('be', 'v'), ('tempt', 'v'), ('to', 'n'), ('use', 'v'), ('any', 'n'), ('extra', 'a'), ('cash', 'n'), ('on', 'n'), ('pre-election', 'n'), ('bribe', 'n'), ('after', 'n'), ('1922', 'n'), ('.', 'n'), ('Mr', 'n'), ('Balls', 'n'), (',', 'n'), ('who', 'n'), ('step', 'v'), ('down', 'r'), ('from', 'n'), ('his', 'n'), ('Treasury', 'n'), ('post', 'n'), ('to', 'n'), ('stand', 'v'), ('a', 'n'), ('a', 'n'), ('Labour', 'n'), ('candidate', 'n'), ('in', 'n'), ('the', 'n'), ('election', 'n'), (',', 'n'), ('have', 'v'), ('suggest', 'v'), ('that', 'n'), ('Mr', 'n'), ('Brown', 'n'), ('would', 'n'), ('meet', 'v'), ('his', 'n'), ('golden', 'a'), ('economic', 'a'), ('rule', 'n'), ('-', 'n'), ('``', 'n'), ('with', 'n'), ('a', 'n'), ('margin', 'n'), ('to', 'n'), ('spare', 'v'), ('"', 'n'), ('.', 'n'), ('He', 'n'), ('say', 'v'), ('he', 'n'), ('hop', 'v'), ('more', 'a'), ('would', 'n'), ('be', 'v'), ('do', 'v'), ('to', 'n'), ('build', 'v'), ('on', 'n'), ('current', 'a'), ('tax', 'n'), ('credit', 'n'), ('rules.', 'n'), ('"', 'n'), ('Mr', 'n'), ('Howard', 'n'), ('say', 'v'), ('the', 'n'), ('``', 'n'), ('grossly', 'r'), ('disproportionate', 'a'), ('"', 'n'), ('test', 'n'), ('match', 'v'), ('the', 'n'), ('hurdle', 'n'), ('minister', 'n'), ('have', 'v'), ('introduce', 'v'), ('for', 'n'), ('civil', 'a'), ('case', 'n'), ('where', 'n'), ('burglar', 'n'), ('where', 'n'), ('claim', 'v'), ('compensation', 'n'), ('from', 'n'), ('householder', 'n'), ('.', 'n'), ('He', 'n'), ('say', 'v'), (':', 'n'), ('``', 'n'), ('The', 'n'), ('problem', 'n'), ('be', 'v'), ('there', 'r'), ('have', 'v'), ('not', 'r'), ('be', 'v'), ('enough', 'a'), ('understanding', 'n'), ('of', 'n'), ('it', 'n'), ('-', 'n'), ('that', 'n'), ('be', 'v'), ('the', 'n'), ('point', 'n'), ('Sir', 'n'), ('John', 'n'), ('Stevens', 'n'), ('be', 'v'), ('make', 'v'), ('and', 'n'), ('the', 'n'), ('prime', 'a'), ('minister', 'n'), ('be', 'v'), ('make', 'v'), ('.', 'n'), ('"', 'n'), ('The', 'n'), ('new', 'a'), ('guidance', 'n'), ('would', 'n'), ('help', 'v'), ('ensure', 'v'), ('clarity', 'n'), ('on', 'n'), ('the', 'n'), ('issue', 'n'), (',', 'n'), ('add', 'v'), ('Mr', 'n'), ('Clarke', 'n'), ('.', 'n'), ('The', 'n'), ('director', 'n'), ('of', 'n'), ('public', 'a'), ('prosecution', 'n'), (',', 'n'), ('Ken', 'n'), ('Macdonald', 'n'), (',', 'n'), ('say', 'v'), ('only', 'r'), ('11', 'n'), ('householder', 'n'), ('or', 'n'), ('occupier', 'n'), ('of', 'n'), ('business', 'n'), ('premise', 'n'), ('have', 'v'), ('be', 'v'), ('prosecute', 'v'), ('in', 'n'), ('the', 'n'), ('last', 'a'), ('15', 'n'), ('year', 'n'), ('.', 'n'), ('Those', 'n'), ('case', 'n'), ('include', 'v'), ('a', 'n'), ('warehouse', 'n'), ('manager', 'n'), ('who', 'n'), ('have', 'v'), ('wait', 'v'), ('for', 'n'), ('a', 'n'), ('burglar', 'n'), (',', 'n'), ('tie', 'v'), ('him', 'n'), ('up', 'r'), (',', 'n'), ('beat', 'v'), ('him', 'n'), ('and', 'n'), ('set', 'v'), ('him', 'n'), ('alight', 'n'), (',', 'n'), ('he', 'n'), ('say', 'v'), ('.', 'n'), ('Tory', 'n'), ('MP', 'n'), ('Patrick', 'n'), ('Mercer', 'n'), ("'s", 'n'), ('private', 'a'), ('member', 'n'), ("'s", 'n'), ('bill', 'n'), ('to', 'n'), ('change', 'v'), ('the', 'n'), ('law', 'n'), ('receive', 'v'), ('a', 'n'), ('first', 'a'), ('reading', 'n'), ('in', 'n'), ('Parliament', 'n'), ('on', 'n'), ('Wednesday', 'n'), ('and', 'n'), ('go', 'v'), ('to', 'n'), ('a', 'n'), ('full', 'a'), ('debate', 'n'), ('next', 'a'), ('month', 'n'), ('.', 'n')]

Few of the wrong things that occur within WordNet Lemmatizer are:-

1. 'as' gets converted to 'a'
2. 'Us' gets converted to 'u'
3. Lemmatizer needs to know the part of speech of the word. As POS tagger is used before applying Lemmatizer; tags are applied to words as per POS tagger and not Lemmatizer. For lemmatizer we inherit tagging mechanism from the POS tagger to produce results.
   The code snippet used to achieve results is :-

```
def mapper(nltk_tag):
    if nltk_tag.startswith('J'): return 'a'
    elif nltk_tag.startswith('V'): return 'v'
    elif nltk_tag.startswith('N'): return 'n'
    elif nltk_tag.startswith('R'): return 'r'
    else: return 'n'
```

Q2. (c) Compare the outputs from Porter Stemming and the Lemmatisation of the same file. Which do you think is the best to use and why?

Porter Stemming is extremely fast and it chops of the affixes in cruel manner. It doesn't produce very accurate results but because of its speed; its widely used. Lemmatizer on the other hand uses proper morphemes to produce lemmas. Lemmatization doesn't produce result if the root / lemma of the word is not found in the dictionary. Hence, its highly accurate but very slow since it scans for the word in the dictionary corpus. The typical usage depends on the requirement. If speed and optimisation is the first priority; then Porter Stemming can be a preferred option. However, if accuracy is the primary concern; then lemmatization works better.

Comparison of Porter Stemming and Lemmatization:-

Original word	**Porter Stemming result**	**Lemmatization result**
ally	alli	ally
denied	deni	deny
premises	premis	premise
prosecuted	prosecut	prosecution
compensation	compens	compensation

Q3.Finally, choose a remote webpage and extract key text content from it. Install the packages you need and then parse it using BeautifulSoup. Try to get to a point where you can extract one of its XML/HTML parts.

The website (PyLadies, 2007) is used for text extraction and scraping activity.

Short snippet of the extracted HTML is :-

```
<html> <head> <title>
  PyLadies – Women Who Love Coding in Python
 </title> </head>
<body>
 <div class="header"> <div id="pyladies_header_logo"><a href="/">
<img alt="PyLadies Logo" height="120" src="/assets/images/pyladies_logo.png" width="260"/>
```


Output of soup.find('div', id = 'stickers_btn') :-
Buy Stickers

Output of soup.find('div')

Output of soup.title
<title>PyLadies – Women Who Love Coding in Python</title>

Output of soup.body.text (Snippet)
Welcome!
We are an international mentorship group with a focus on helping more women become active participants and leaders in the Python open-source community. Our mission is to promote, educate and advance a diverse Python community through outreach, education, conferences, events and social gatherings.

References
Insight resources, (2006), *Insight Project Resources*. Available at: http://mlg.ucd.ie/datasets/bbc.html

PyLadies, (2007), *PyLadies*. Available at: https://www.pyladies.com/

D. Greene and P. Cunningham. "Practical Solutions to the Problem of Diagonal Dominance in Kernel Document Clustering", Proc. ICML 2006. [PDF] [BibTeX].