CIS630 Project 1 - Text-based analysis on NYT

Yayang Tian, Chen Ma

March 8, 2013

1. **Introduction**

   In this project, we performed text analysis on the New York Times, hoping to find some new words that also express opinions. The key is to extract words with higher co-occurence rate with respect to emotion words described in MPQA, Bing Liu, FrameNet and WordnetAffect.

2. **How Does the Matching Work**

   (a) We extract each xml into separate words without punctuations, each of which is in lower-case. We don't perform pos taggings.

   (b) To find new positive words, we only utilize the 1000 positive files in each lexicon and vice versa. Because from our histogram distributions in the appendix, we see if we use 1000 positive files to find negative new words, most of the files occur about only 10 times, compared to our approach of 100 - 200 times.

3. **Experiment Steps**

   (a) **Extract representative sets.** For each of the emotion lexicon: "mpqa, bingliu, wordnet, and framenet", extracted 1000 highest-ranked positive set {P}, 1000 highest-ranked negative set {N}, and 1000 random set{R}. The ranking is according the # of emotion words in that document.

   (b) **Derive contexts with emotion words in middle.** Extracted 21-grams with the target emotion words in the middle. When calculating PMI, delete the middle words.

   (c) **Calculate PMI of each distinct context word.** For each distinct word that occurs more than 5 times in the 2000 articles,

   (d) **Rank PMI and derived new emotion lexicons.** For the resulting PMI, we rank them and pick those that don't appear in the prior lexicons. We do this in terms of "mpqa, bingliu, wordnet and framenet" respectively.

4. **PMI Calculation**

$$PMI^{+}(word) = log_2 \frac{p(word\&emotion^{+})}{p(word)p(emotion^{+})} = log_2[N \times \frac{hits(word\ \ NEAR\ \ emotion^{+})}{hits(word)hits(emotion^{+})}] \qquad (1)$$

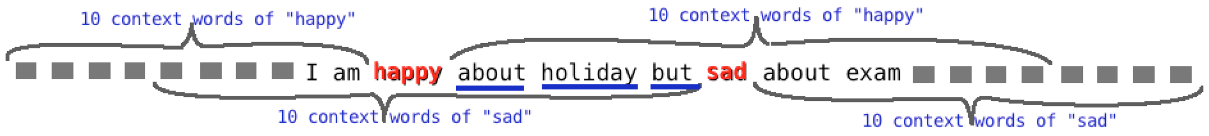
```
  # PMI = log2[ N * #co-occur / (#word * #emotion+) ]
PMI[seedWord] = math.log((numTotalWords * numCooccur) / (numWord * numEmotion), 2)
```

   (a) "+" means positive and "-" means negative. $PMI^{+}(word)$ or $PMI^{-}(word)$ means how likely such a word would co-occur near a positive/negative word.

   (b) When calculating "NEAR", this word occur within the range of 21-grams contexts: [10 words before emotion, word, 10 words after emotion].

   (c) To avoid division by zero, I adopt Laplace Smoothing by adding 0.01 to all hits.

   (d) Some tricks. When calculating the hits, I don't take the current target emotion word into consideration, and pay special care to overlapping. For example, in Figure 1, the sentence contains two contrast emotion classes "positive" and "negative", then the three words "about", "holiday" and "but" overlapping have to be counted twice. Yet, if the two emotion words are belonging to the same class, they should be only considered once.


10 context words of "happy"
10 context words of "happy"
I am happy about holiday but sad about exam
10 context words of "sad"
10 context words of "sad"

   (e) Since framenet is not designed for polarity classification, we treat the whole corpus as a single set that expresses opinions.

5. **Discussion**

Distribution & Overlapping:
We selected about top 10% of each set, which is about 2000 top new words, and overlap them to have a sense results.

Number of New Words in Each Dictionary					
number of new words:	MPQA	Bing Liu	FrameNet	WordNet	#Overlap
# of positive words from positive set	27156	27412	28915	23310	19207
# of negative words from negative set	23790	22921	28915	16524	12779
# of positive words from random set	6535	4171	6797	5464	2749
# of negative words from random set	6563	4092	6797	3537	2083

Top Overlap:

(1). New overlapping positive words with top 2000 PMI values:
’strives’, ’postscript’, ’oddball’, ’chugging’, ’decorous’, ’fizz’, ’infusing’, ’brushings’, ’cockamamie’, ’drip’, ’glazing’, ’mccracken’, ’rec’, ’pudgy’, ’seabird’, ’naturalism’, ’slumber’, ’radiates’, ’superhero’, ’showoff’, ’humored’, ’pfrancing’, ’elevating’, ’puffed’, ’sneaky’, ’genuinely’, ’enigmas’, ’cultivating’, ’barrage’, ’serendipity’, ’patterned’, ’erudition’, ’dearly’, ’primates’, ’oakley’, ’fable’, ’llama’, ’moonlighting’, ’childlike’, ’halpern’, ’squabble’, ’contrives’, ’tillmans’, ’nonchalant’, ’havlish’, ’cattelan’, ’hooker’, ’crinkles’, ’overflows’, ’hyperactive’, ’blush’, ’disarmingly’, ’conveying’, ’beadwork’, ’dashes’, ’gesture’, ’beecroft’, ’eras’, ’brandeis’, ’orchestrations’, ’cliche’, ’carbone’, ’catches’, ’throttle’, ’lusty’, ’blaze’, ’marionettes’, ’misunderstanding’, ’asymmetrical’, ’virtuosity’, ’mathematicians’, ’esoteric’, ’nighttime’, ’resurrection’, ’slurpee’, ’haughty’, ’needlepoint’, ’punks’
(2). New overlapping negative words with top 2000 PMI values:
’refracts’, ’thundering’, ’slumber’, ’ebersole’, ’galumphing’, ’primates’, ’recrimination’, ’inducing’, ’redeems’, ’balances’, ’dissolves’, ’cumpsty’, ’hilariously’, ’overflows’, ’individuality’, ’annihilatingly’, ’cavernous’, ’unleashed’, ’compulsively’, ’conveys’, ’climax’, ’effortlessly’, ’herded’, ’bosnian’, ’nolte’, ’casts’, ’snort’, ’existential’, ’resolves’, ’cannonballs’, ’toothache’, ’triggered’, ’abdominal’, ’appetites’, ’trapeze’, ’lark’, ’hyperactive’, ’perennial’, ’tactic’, ’deranged’, ’hairy’, ’overcoming’, ’passivity’, ’needn’, ’profound’, ’engendered’, ’faintly’, ’aggravated’, ’outpouring’, ’spruce’, ’backwards’, ’doghouse’, ’financier’, ’respite’, ’donovan’

(a) **Do the lists differ?**
For highest PMI words, The lists differ a lot for different lexicons, but there are many overlap words whose results are promising. As the range of PMI gets loose, the differences become smaller.

(b) **Which ones do you like more?**
Personlly, I love the results from MPQA and Bing Liu’s list. They make more sense compared to other dictionaries.Some of the words are quite interesting. Yet words from FrameNet do not classify the polarity well and words from wordNet don’t have strong polarity.

(c) **Do you find them meaningful in terms of affect and opinions?**
When inspecting the words, I find them somewhat meaningful, they seem to have higher probabilities to co-occurs with emotional words in the four lexicon. For example, in new words which derived from Bing Liu’s Negative List, ”gulp”, ”fauve”, ”droopy”, ”sneaky”, ”pudgy”, ”unsatisfied, ”burglary” and ”nonchalant” really express strong negative affection. Nevertheless, some of the implicit words have not been discovered, and some of the words discovered do not make much sense, the reason of which has been explained in the conclusion part.

(d) **Any future ideas?**
(1) Syntactic Parser: Caculating PMI is not always effective in terminology findings because it misses too many low-frequency terms, terms with variations, and terms with more than one words. To address this, we could use symbolic approaches that rely on syntactic description of terms, namely noun phrases.

(2) Pos Taggings: pos tagging is important because opinions are often ajective and verbs and features are often nouns and noun phrases.

(3) Pre-processing: removal of stopwords, stemming, and fuzzy matching are also essential.

(4) Determine more fine-grained classes: Besides positive and negative polarity, it’s more helpful to determine whether the holders are happy, excited, sad, angry, or ashamed.

(5) Make full use of PMI: We can further calculate top relevance and semantic orientation of each sentence, as well as derive summarization based on the ranked scores.

(6) Besides PMI, maybe we could find alternative ways like Binomial Test, to extract new opinion words.

For more information about new words and discussion, Please see appendix.

6. **Histogram of Emotion Words**

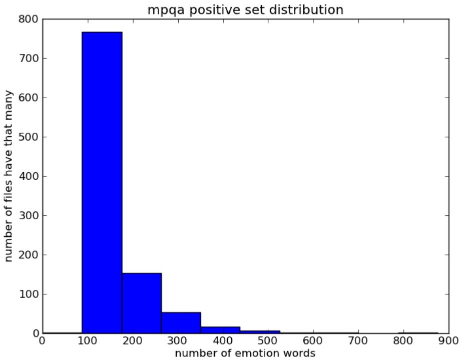


Figure 1: MPQA (Positive)

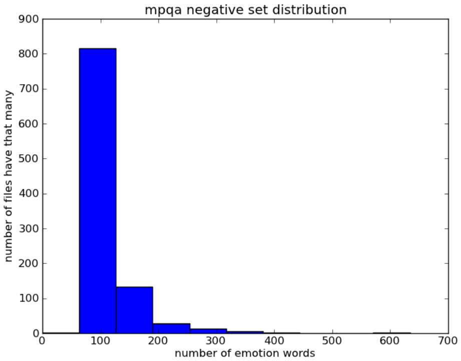


Figure 2: MPQA (Negative)

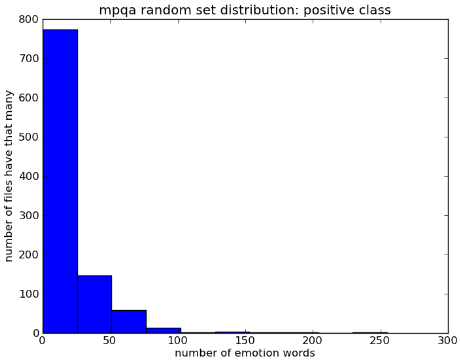


Figure 3: MPQA (Random)

Discussion of Histograms:
The histogram shows for each "number n" of emotions words, how many documents from each group contained that many. The distributions for different lexicons are quite similar. To see others, please refer to appendix.

(a) Just as professor said "The reason to look at this is because in this exercise we are implicitly assuming that if we get the documents that have many affect words, we will find news stories that are rich in affect. "

(b) It turns out that for the 2000 documents of rich affect words, no matter which dictionary we choose, all documents have relatively ample samples of emotion words: almost all have 50-150 emotion words per article.

(c) Yet for the random set, most of articles are lack of emotion words, about 75 % of articles have only 0 - 10, which are not good sources to calculate PMI in order to get reasonable new opinion words.

7. **Final Conclusion**

Based on our experiments, we conclude that using PMI to extract new opinion words is an intuitive method. For one thing, our promising results shows that for words with high PMI, they really express opinions with polarities and they don't appear in any of the lexicons. Some of the words are really interesting: like "aggravated", "redeems", "deranged", "herded" and "lark".

But it might not be a very effective approach. First, it omit many words low frequency terms, terms with variations, because PMI is high if and only if the same word is different contexts. For another, it introduces too many words that we use rarely but rich in one specific document which renders a lot false negative errors. Third, it neither take syntactic dependencies nor domains and negations into consideration.

Appendix 1: Top 100 Positive Words (from 1000 Positive Sets)

(1). MPQA
’totems’, ’reappraisal’, ’craftsmanlike’, ’didacticism’, ’stringently’, ’apollonian’, ’incompetency’, ’storaro’, ’bergeret’, ’murch’, ’hedonist’, ’polenzani’, ’encapsulates’, ’strives’, ’postscript’, ’stoicism’, ’workmanship’, ’civilizing’, ’oddball’, ’emanates’, ’nytoday’, ’allusions’, ’chugging’, ’crusts’, ’imparts’, ’retirees’, ’schematic’, ’gaines’, ’booster’, ’decorous’, ’restores’, ’powwows’, ’behaved’, ’jolting’, ’donofrio’, ’metalsmiths’, ’collegial’, ’1503’, ’twinkle’, ’theatricality’, ’fizz’, ’feats’, ’huggy’, ’flexing’, ’accessorized’, ’unspeakably’, ’unwavering’, ’infusing’, ’enduringly’, ’sfumato’, ’brushings’, ’cockamamie’, ’rdler’, ’thade’, ’malfunction’, ’spiels’, ’accomplishes’, ’turners’, ’brushwork’, ’philosophizing’, ’futility’, ’determinism’, ’satirist’, ’calculatedly’, ’nabokov’, ’confrontational’, ’abcd’, ’titillation’, ’foreshortening’, ’arcy’, ’lunacy’, ’stamina’, ’incandescence’, ’beginners’, ’photojournalism’, ’dictates’, ’stickball’, ’discography’, ’nuance’, ’cassidy’, ’jolts’, ’recklessly’, ’flattened’, ’bites’, ’drip’, ’immediacy’, ’marxist’, ’poyet’, ’intellect’, ’culminates’, ’musketeers’, ’clowes’, ’screenplays’, ’gorris’, ’zoolander’, ’resurrecting’, ’ideologies’, ’glazing’, ’mccracken’, ’emasculating’
(2) Bing Liu:
’stringently’, ’imparts’, ’totems’, ’didacticism’, ’reappraisal’, ’bristles’, ’flexing’, ’craftsmanlike’, ’reviewer’, ’apollonian’, ’bergeret’, ’oddball’, ’metalsmiths’, ’illuminators’, ’storaro’, ’uggams’, ’murch’, ’malfunction’, ’seabird’, ’calamities’, ’calculatedly’, ’trumping’, ’chocolatier’, ’emanates’, ’philosophizing’, ’postscript’, ’plato’, ’mccracken’, ’nytoday’, ’pixilated’, ’feats’, ’hedonist’, ’jolts’, ’hammons’, ’thalberg’, ’pokes’, ’satirist’, ’sexiness’, ’drip’, ’theatricality’, ’vittorio’, ’slumber’, ’liveliness’, ’conjuring’, ’rereleased’, ’booster’, ’crusts’, ’irrepressibly’, ’ingratiating’, ’recklessly’, ’intellect’, ’ingnue’, ’naturalism’, ’squander’, ’whiplash’, ’chugging’, ’wurlitzer’, ’allusions’, ’accomplishes’, ’erudition’, ’polenzani’, ’fishmonger’, ’liberman’, ’sparingly’, ’antecedents’, ’dispenser’, ’strives’, ’seurat’, ’enigmas’, ’crayon’, ’photojournalism’, ’humored’, ’workmanship’, ’bouche’, ’companionable’, ’exuded’, ’greaseless’, ’patterned’, ’decorous’, ’showoff’, ’insubstantial’, ’cinematography’, ’rifle’, ’vastly’, ’representations’, ’fizz’, ’ensor’, ’chanteuse’, ’macabre’, ’elevating’, ’mumbles’, ’claptrap’, ’germont’, ’cushioned’, ’tanned’, ’sandal’, ’draftsmen’, ’giraffes’, ’saxophones’, ’artifice’
(3) FrameNet:
’huggy’, ’authorship’, ’fizz’, ’philosophizing’, ’contrives’, ’oddball’, ’enigmas’, ’fissure’, ’esbjornson’, ’thundering’, ’gulp’, ’stoner’, ’wildebeest’, ’benevolence’, ’burglary’, ’pudgy’, ’unsatisfied’, ’groundhog’, ’riveted’, ’implode’, ’strippers’, ’futility’, ’shaft’, ’smashes’, ’ovals’, ’bohemians’, ’cradles’, ’sharpness’, ’aurore’, ’cutout’, ’refracts’, ’perforated’, ’droops’, ’fauve’, ’bottlelike’, ’lorrain’, ’bodylike’, ’forearm’, ’clment’, ’nebula’, ’polaroids’, ’seabird’, ’bosnian’, ’nonchalant’, ’compulsively’, ’fable’, ’liar’, ’spectral’, ’crossbones’, ’lullaby’, ’fra’, ’czanne’, ’gouache’, ’magritte’, ’sociologist’, ’lippi’, ’interrupts’, ’eras’, ’sendup’, ’kesey’, ’inept’, ’plaintive’, ’weepie’, ’trois’, ’naturalism’, ’snorting’, ’leda’, ’pathway’, ’droopy’, ’infusing’, ’clouded’, ’catunda’, ’mouthful’, ’sawmill’, ’longue’, ’clunk’, ’sarong’, ’sickly’, ’barrage’, ’unbearably’, ’blob’, ’philosophically’, ’winded’, ’sneaky’, ’sketch’, ’shawl’, ’homophobia’, ’skylight’, ’bong’, ’haircut’, ’wot’, ’hedonism’, ’dissolves’, ’toga’, ’slicked’, ’butternut’, ’intolerance’, ’parable’, ’meanness’, ’marxist’
(4) WordNet
’oddball’, ’beecroft’, ’cattelan’, ’crinkles’, ’3587’, ’civilizing’, ’encapsulates’, ’farces’, ’seabird’, ’allusions’, ’tillmans’, ’rending’, ’jolting’, ’shoeshine’, ’protectors’, ’hairy’, ’maurizio’, ’779’, ’lusty’, ’thundering’, ’krasner’, ’eggy’, ’quin’, ’congestive’, ’reviewer’, ’unending’, ’stoicism’, ’decorous’, ’framing’, ’wittily’, ’manifest’, ’winkle’, ’restiveness’, ’successors’, ’serendipity’, ’asserting’, ’seismic’, ’souffls’, ’cadences’, ’thoughtfulness’, ’overflows’, ’warmest’, ’nebula’, ’fielder’, ’puffed’, ’potency’, ’brownrigg’, ’depreciation’, ’mouthed’, ’whoa’, ’yields’, ’2680’, ’dreamcoat’, ’focaccia’, ’proust’, ’dimwitted’, ’misanthropy’, ’leopoldo’, ’natured’, ’7503’, ’blob’, ’aspiration’, ’suzy’, ’stampede’, ’boundless’, ’swamps’, ’fez’, ’humored’, ’tele’, ’democracies’, ’foreshortening’, ’psychobabbling’, ’squabble’, ’cultivating’, ’poppingly’, ’totems’, ’gimnez’, ’faking’, ’unrequited’, ’metalsmiths’, ’carnivores’, ’beatlemania’, ’rorschachs’, ’oestreich’, ’unfashionable’, ’dorsky’, ’romanowski’, ’lesage’, ’sidewalks’, ’filial’, ’netomat’, ’stirrup’, ’deakins’, ’havlish’, ’tibetans’, ’sitch’, ’theatricality’, ’putts’, ’fizz’, ’grimacing’

Appendix 2: Top 100 Negative Words (from 1000 Negative Sets)

(1). MPQA
[’sheeler’, ’mishmash’, ’accessorized’, ’fiascos’, ’stant’, ’godley’, ’bogs’, ’stagecraft’, ’megalomaniacal’, ’refracts’, ’digested’, ’marauders’, ’tedrow’, ’bowel’, ’lows’, ’1892’, ’wasting’, ’highs’, ’schreck’, ’thundering’, ’donofrio’, ’stirringly’, ’graciela’, ’scam’, ’subaltern’, ’mugatu’, ’defensiveness’, ’hypnotically’, ’slumber’, ’fondly’, ’fated’, ’lulled’, ’swine’, ’encephalopathy’, ’jakob’, ’crazed’, ’spoofs’, ’gestural’, ’creutzfeldt’, ’daniele’, ’neutrality’, ’valiantly’, ’egomaniacal’, ’berthe’, ’spongiform’, ’resurfaced’, ’bemusement’, ’terminally’, ’pixilated’, ’pseudonymous’, ’unsubtle’, ’irrepressibly’, ’malnutrition’, ’ebersole’, ’scrapie’, ’subconscious’, ’comatose’, ’diets’, ’bozo’, ’supermodel’, ’reeling’, ’engulfing’, ’sadism’, ’tantrums’, ’dion’, ’hedonist’, ’counterfeiting’, ’numbing’, ’elk’, ’smashes’, ’besieged’, ’uncontrollable’, ’galumphing’, ’primates’, ’mcclinton’, ’ravenous’, ’masterminds’, ’mongering’, ’sportscasters’, ’mantello’, ’nussbaum’, ’pooled’, ’pileup’, ’mizuki’, ’monumentally’, ’recrimination’, ’mending’, ’formalism’, ’inducing’, ’redeems’, ’gravedigger’, ’claptrap’, ’sustains’, ’shrugs’, ’dictum’, ’pitting’, ’persuasively’, ’dafoe’, ’vomiting’, ’nihilistic’]
(2) Bing Liu:
[’refracts’, ’slumber’, ’defensiveness’, ’bogs’, ’stagecraft’, ’accessorized’, ’pulsing’, ’stant’, ’godley’, ’sexiness’, ’annihilatingly’, ’hypnotically’, ’underdeveloped’, ’fiascos’, ’primates’, ’nausea’, ’bowel’, ’megalomaniacal’, ’masterminds’, ’forthrightness’, ’valiantly’, ’spoofs’, ’gestural’, ’1892’, ’lulled’, ’crazed’, ’successively’, ’schreck’, ’digested’, ’pixilated’, ’nonconformist’, ’gastroenteritis’, ’unsubtle’, ’throwaway’, ’inbred’, ’mishmash’, ’classless’, ’deadliest’, ’mending’, ’inducing’, ’counterfeiting’, ’smashes’, ’galumphing’, ’nihilistic’, ’diets’, ’sportscasters’, ’limbed’, ’respiratory’, ’hedonist’, ’contributory’, ’warchus’, ’noun’, ’criminally’, ’gravedigger’, ’recrimination’, ’sadism’, ’pileup’, ’tedrow’, ’fecal’, ’dismisses’, ’monumentally’, ’lows’, ’thundering’, ’dementia’, ’mystique’, ’undetected’, ’cowardice’, ’humanistic’, ’nussbaum’, ’disturbances’, ’bozo’, ’riskiest’, ’aedes’, ’tantrums’, ’effeminate’, ’believability’, ’squeamish’, ’fending’, ’bruck’, ’007’, ’abatement’, ’unduly’, ’ungodly’, ’abdominal’, ’conveys’, ’smirk’, ’roundelay’, ’frights’, ’curative’, ’skulduggery’, ’sores’, ’pensions’, ’musketeers’, ’slurry’, ’pseudo’, ’ravenous’, ’blooded’, ’gigantically’, ’disproportionately’, ’ripper’]
(3) FrameNet:
[’huggy’, ’authorship’, ’fizz’, ’philosophizing’, ’contrives’, ’oddball’, ’enigmas’, ’fissure’, ’esbjornson’, ’thundering’, ’gulp’, ’stoner’, ’wildebeest’, ’benevolence’, ’burglary’, ’pudgy’, ’unsatisfied’, ’groundhog’, ’riveted’, ’implode’, ’strippers’, ’futility’, ’shaft’, ’smashes’, ’ovals’, ’bohemians’, ’cradles’, ’sharpness’, ’aurore’, ’cutout’, ’refracts’, ’perforated’, ’droops’, ’fauve’, ’bottlelike’, ’lorrain’, ’bodylike’, ’forearm’, ’clment’, ’nebula’, ’polaroids’, ’seabird’, ’bosnian’, ’nonchalant’, ’compulsively’, ’fable’, ’liar’, ’spectral’, ’crossbones’, ’lullaby’, ’fra’, ’czanne’, ’gouache’, ’magritte’, ’sociologist’, ’lippi’, ’interrupts’, ’eras’, ’sendup’, ’kesey’, ’inept’, ’plaintive’, ’weepie’, ’trois’, ’naturalism’, ’snorting’, ’leda’, ’pathway’, ’droopy’, ’infusing’, ’clouded’, ’catunda’, ’mouthful’, ’sawmill’, ’longue’, ’clunk’, ’sarong’, ’sickly’, ’barrage’, ’unbearably’, ’blob’, ’philosophically’, ’winded’, ’sneaky’, ’sketch’, ’shawl’, ’homophobia’, ’skylight’, ’bong’, ’haircut’, ’wot’, ’hedonism’, ’dissolves’, ’toga’, ’slicked’, ’butternut’, ’intolerance’, ’parable’, ’meanness’, ’marxist’]
(4) WordNet
[’refracts’, ’annihilatingly’, ’cumpsty’, ’duckling’, ’dubin’, ’balances’, ’primates’, ’dissolves’, ’ebersole’, ’cavernous’, ’recrimination’, ’vagaries’, ’behaviors’, ’privation’, ’rained’, ’skarsgard’, ’thundering’, ’erupts’, ’priceless’, ’osama’, ’bin’, ’pixilated’, ’coyote’, ’watchers’, ’contaminant’, ’burrowing’, ’yorick’, ’marionettes’, ’persists’, ’7e’, ’playthings’, ’sulley’, ’churn’, ’bemusement’, ’weld’, ’admitting’, ’suppressing’, ’nussbaum’, ’culpability’, ’serbian’, ’hopelessness’, ’cannonballs’, ’individuality’, ’conleth’, ’swallowed’, ’jays’, ’condolence’, ’upside’, ’pear’, ’redeems’, ’pleaded’, ’devils’, ’disturbance’, ’dion’, ’galumphing’, ’creaks’, ’ungodly’, ’connemara’, ’pared’, ’resurfaced’, ’hairy’, ’egomaniacal’, ’flagrant’, ’effortlessly’, ’plead’, ’tenderness’, ’zakaria’, ’implode’, ’menace’, ’forthrightness’, ’grossing’, ’slapping’, ’literate’, ’rider’, ’snort’, ’kris’, ’provinces’, ’rhapsody’, ’hilariously’, ’wistful’, ’didacticism’, ’bogs’, ’nightmarishly’, ’curls’, ’parque’, ’comatose’, ’carpeted’, ’pun’, ’appetites’, ’overflows’, ’drillon’, ’watered’, ’fundamentalism’, ’klemperer’, ’befall’, ’implying’, ’annals’, ’romanticized’, ’shutters’, ’countenance’]

Appendix 3: Top 100 Positive Words (from 1000 Random Sets)

The words calculated from random sets do not make much sense because the emotion words contained are too sparse. So it’s just help us to have a sense for the influence of emotion words density to the final results.

(1). MPQA
’wednesdays’, ’cheever’, ’traditionally’, ’component’, ’wit’, ’religion’, ’forever’, ’styles’, ’dedication’, ’cake’, ’eclipse’, ’kavalier’, ’painterly’, ’cuisine’, ’sentimental’, ’leone’, ’capped’, ’1076’, ’crazy’, ’colors’, ’ambition’, ’comic’, ’belief’, ’missed’, ’concepts’, ’sierra’, ’creator’, ’quest’, ’defending’, ’physician’, ’flavors’, ’sadness’, ’desperately’, ’equally’, ’appoint’, ’soup’, ’parody’, ’aged’, ’dessert’, ’packing’, ’phyllis’, ’retter’, ’fought’, ’carrot’, ’remembered’, ’mourn’, ’looks’, ’strength’, ’damage’, ’helps’, ’confront’, ’sid’, ’legacy’, ’deeply’, ’adventures’, ’emphasize’, ’emerge’, ’encouraged’, ’violations’, ’considers’, ’secular’, ’albanians’, ’wimbledon’, ’pea’, ’ruling’, ’nature’, ’coats’, ’steep’, ’traditions’, ’performers’, ’pursued’, ’proponents’, ’disciplined’, ’wiggers’, ’improvements’, ’gibson’, ’eugene’, ’regulate’, ’animation’, ’primaries’, ’applause’, ’invention’, ’hearts’, ’generations’, ’knowledge’, ’ackerman’, ’fielder’, ’victories’, ’die’, ’pet’, ’recognize’, ’rhythm’, ’aimed’, ’initiative’, ’arguments’, ’dishes’, ’dematteis’, ’arguing’, ’cooking’, ’madonna’
(2) Bing Liu:
’blended’, ’niente’, ’keeps’, ’mere’, ’thai’, ’teachers’, ’relationships’, ’beef’, ’flavors’, ’gesture’, ’dishes’, ’sorely’, ’seafood’, ’caring’, ’yellow’, ’environment’, ’mountain’, ’countless’, ’bowl’, ’ties’, ’involving’, ’tomato’, ’citrus’, ’shank’, ’brings’, ’choral’, ’myself’, ’debut’, ’dummy’, ’differences’, ’creamy’, ’reading’, ’danielle’, ’embraced’, ’risks’, ’relating’, ’derby’, ’vineyards’, ’catherine’, ’britain’, ’esteemed’, ’suits’, ’miami’, ’victories’, ’appetizer’, ’beverley’, ’feels’, ’methods’, ’crab’, ’vegetables’, ’providing’, ’sorrow’, ’spots’, ’ivory’, ’treat’, ’woods’, ’adds’, ’shirley’, ’committed’, ’identity’, ’blend’, ’prince’, ’lemon’, ’flesh’, ’basil’, ’knowles’, ’believes’, ’adam’, ’advice’, ’oak’, ’gimelstob’, ’apple’, ’napa’, ’difference’, ’charlotte’, ’fried’, ’access’, ’grandfather’, ’condolences’, ’food’, ’chicken’, ’swan’, ’updated’, ’chapel’, ’broiled’, ’homeowners’, ’shipbuilders’, ’rosemary’, ’stein’, ’indoor’, ’none’, ’employers’, ’bite’, ’aibo’, ’unusually’, ’shepard’, ’dominique’, ’greatly’, ’concept’, ’icons’
(3) FrameNet:
’atop’, ’vehicle’, ’bunch’, ’reminder’, ’fugitive’, ’monument’, ’displayed’, ’figurines’, ’poet’, ’foro’, ’lap’, ’pile’, ’italico’, ’rehearsal’, ’driver’, ’existence’, ’truck’, ’racing’, ’backdrop’, ’silicon’, ’browns’, ’slipped’, ’understands’, ’pencil’, ’toll’, ’ticket’, ’comic’, ’banana’, ’playground’, ’tuned’, ’kicking’, ’fitting’, ’neat’, ’theatrical’, ’suggestion’, ’knife’, ’sudden’, ’lobster’, ’franz’, ’laugh’, ’sofa’, ’negotiating’, ’sings’, ’successfully’, ’sampling’, ’disc’, ’glimpse’, ’wrestler’, ’hangs’, ’sherry’, ’reflecting’, ’pipe’, ’remembered’, ’beside’, ’vision’, ’bomb’, ’balanced’, ’guest’, ’designated’, ’examination’, ’billy’, ’ending’, ’absolute’, ’graduating’, ’solely’, ’openings’, ’abstract’, ’reader’, ’thick’, ’weapon’, ’bathroom’, ’sacrifice’, ’bit’, ’walks’, ’muslims’, ’oversight’, ’financier’, ’span’, ’nonetheless’, ’subjects’, ’prayer’, ’rational’, ’describe’, ’achievement’, ’verdict’, ’focuses’, ’chunk’, ’emerges’, ’disappear’, ’shoe’, ’developments’, ’stretched’, ’reinforced’, ’chart’, ’christ’, ’juror’, ’switzerland’, ’tasty’, ’stagehands’, ’seller’
(4) WordNet
’quin’, ’lesbian’, ’refrain’, ’workplace’, ’cilantro’, ’linden’, ’debra’, ’judith’, ’dearest’, ’friendship’, ’anna’, ’looks’, ’tears’, ’whenever’, ’artificial’, ’survivor’, ’nee’, ’lore’, ’chartreuse’, ’varieties’, ’commuter’, ’minded’, ’lacks’, ’pfrancing’, ’shock’, ’healing’, ’betrayal’, ’generosity’, ’bulgari’, ’krzyzewski’, ’heartfelt’, ’embraced’, ’kindness’, ’dear’, ’grandmother’, ’advocates’, ’despair’, ’terry’, ’regulators’, ’maurice’, ’sounded’, ’corners’, ’linebacker’, ’champions’, ’platforms’, ’ellen’, ’blizzard’, ’cherished’, ’missouri’, ’grandfather’, ’touches’, ’beemo’, ’perrotta’, ’isaac’, ’clockwork’, ’stephanie’, ’crying’, ’repaid’, ’chat’, ’chairs’, ’catherine’, ’theo’, ’vessels’, ’lowest’, ’pointing’, ’species’, ’bidding’, ’shah’, ’frances’, ’benchmark’, ’keen’, ’wendy’, ’inner’, ’fabulous’, ’floating’, ’brandrup’, ’spano’, ’cleanup’, ’mourning’, ’existence’, ’resolve’, ’asserted’, ’2004’, ’deepest’, ’likelihood’, ’sally’, ’sofa’, ’disturbing’, ’annoying’, ’gregory’, ’creeping’, ’ten’, ’offspring’, ’rid’, ’claudia’, ’gentle’, ’random’, ’castle’, ’acceptance’, ’lush’

Appendix 4: Top 100 Negative Words (from 1000 Random Sets)

The words calculated from random sets do not make much sense because the emotion words contained are too sparse. So it's just help us to have a sense for the influence of emotion words density to the final results.

(1). MPQA
’iraq’, ’attempted’, ’beetles’, ’handling’, ’protecting’, ’psychiatric’, ’survival’, ’weapon’, ’pleaded’, ’somehow’, ’graves’, ’surrounding’, ’fingers’, ’stems’, ’finds’, ’resulting’, ’politically’, ’mentally’, ’suffered’, ’combination’, ’heartfelt’, ’pronounced’, ’destroyed’, ’jokes’, ’synchro’, ’neighbor’, ’b1’, ’sexual’, ’triumph’, ’potentially’, ’attitudes’, ’knee’, ’combat’, ’knocked’, ’kavalier’, ’manage’, ’scrimmage’, ’solve’, ’convey’, ’tesser’, ’moose’, ’goats’, ’doomed’, ’slowing’, ’silent’, ’revive’, ’caused’, ’contributing’, ’downsizing’, ’civilian’, ’massive’, ’surely’, ’hijackers’, ’greenspan’, ’ass’, ’sanctions’, ’courageous’, ’economists’, ’dense’, ’patterson’, ’amid’, ’humanity’, ’reveal’, ’sentimental’, ’risks’, ’causes’, ’citing’, ’immigrants’, ’profound’, ’prisoners’, ’implications’, ’noise’, ’islamic’, ’script’, ’parliament’, ’cells’, ’responses’, ’confederate’, ’speeches’, ’savoy’, ’plagued’, ’b3’, ’kuklin’, ’byline’, ’alternating’, ’compassion’, ’marked’, ’dangers’, ’chevrolet’, ’margins’, ’denise’, ’campus’, ’calm’, ’a13’, ’pleading’, ’nail’, ’shouldn’, ’suffers’, ’tap’, ’tactics’
(2) Bing Liu:
’soldier’, ’prop’, ’trembling’, ’emotions’, ’takeover’, ’views’, ’adopt’, ’sentenced’, ’gays’, ’earthquake’, ’civilians’, ’electrical’, ’prepared’, ’politically’, ’blacks’, ’encounter’, ’discover’, ’divided’, ’a8’, ’robbery’, ’cristina’, ’dickens’, ’colonial’, ’treatment’, ’prisoners’, ’attempted’, ’dealers’, ’likes’, ’providing’, ’fuji’, ’ned’, ’cathedral’, ’medicare’, ’condition’, ’measures’, ’communist’, ’bonds’, ’driven’, ’caused’, ’relationships’, ’crimes’, ’dummy’, ’episode’, ’1979’, ’differences’, ’lungs’, ’diagnosed’, ’zero’, ’occasionally’, ’finances’, ’tribunal’, ’boots’, ’sexual’, ’hernandez’, ’actual’, ’howard’, ’bid’, ’bureaucrats’, ’stick’, ’judges’, ’presumably’, ’suggestion’, ’packed’, ’airport’, ’occasional’, ’districts’, ’expressed’, ’perfect’, ’sergeant’, ’lists’, ’viewed’, ’legacy’, ’smooth’, ’repeat’, ’hidden’, ’oregon’, ’values’, ’flying’, ’notably’, ’weight’, ’allowing’, ’korean’, ’solution’, ’newly’, ’biological’, ’significantly’, ’aspects’, ’briefly’, ’determined’, ’esteemed’, ’disease’, ’snow’, ’wealthy’, ’feelings’, ’tone’, ’resolved’, ’rescue’, ’sees’, ’empty’, ’save’
(3) FrameNet:
’atop’, ’vehicle’, ’bunch’, ’reminder’, ’fugitive’, ’monument’, ’displayed’, ’figurines’, ’poet’, ’foro’, ’lap’, ’pile’, ’italico’, ’rehearsal’, ’driver’, ’existence’, ’truck’, ’racing’, ’backdrop’, ’silicon’, ’browns’, ’slipped’, ’understands’, ’pencil’, ’toll’, ’ticket’, ’comic’, ’banana’, ’playground’, ’tuned’, ’kicking’, ’fitting’, ’neat’, ’theatrical’, ’suggestion’, ’knife’, ’sudden’, ’lobster’, ’franz’, ’laugh’, ’sofa’, ’negotiating’, ’sings’, ’successfully’, ’sampling’, ’disc’, ’glimpse’, ’wrestler’, ’hangs’, ’sherry’, ’reflecting’, ’pipe’, ’remembered’, ’beside’, ’vision’, ’bomb’, ’balanced’, ’guest’, ’designated’, ’examination’, ’billy’, ’ending’, ’absolute’, ’graduating’, ’solely’, ’openings’, ’abstract’, ’reader’, ’thick’, ’weapon’, ’bathroom’, ’sacrifice’, ’bit’, ’walks’, ’muslims’, ’oversight’, ’financier’, ’span’, ’nonetheless’, ’subjects’, ’prayer’, ’rational’, ’describe’, ’achievement’, ’verdict’, ’focuses’, ’chunk’, ’emerges’, ’disappear’, ’shoe’, ’developments’, ’stretched’, ’reinforced’, ’chart’, ’christ’, ’juror’, ’switzerland’, ’tasty’, ’stagehands’, ’seller’
(4) WordNet
’shutting’, ’stripped’, ’bin’, ’osama’, ’qaeda’, ’pleaded’, ’frequency’, ’earthquakes’, ’esteemed’, ’verge’, ’excess’, ’lovers’, ’predictions’, ’mccarrick’, ’burgundy’, ’evoke’, ’softened’, ’cancellations’, ’rankings’, ’portraits’, ’cracked’, ’evil’, ’nuclear’, ’missiles’, ’atkins’, ’apologize’, ’shut’, ’b9’, ’passing’, ’estimate’, ’exceptions’, ’welcomed’, ’murders’, ’steadily’, ’tears’, ’animated’, ’b5’, ’avon’, ’pfrancing’, ’aging’, ’oskar’, ’inspectors’, ’elevator’, ’salvadoran’, ’causing’, ’linked’, ’mouse’, ’somehow’, ’haouari’, ’chip’, ’berzok’, ’surfaced’, ’vertical’, ’convened’, ’strained’, ’lock’, ’conspiring’, ’paterson’, ’hopeless’, ’eccentric’, ’edges’, ’saint’, ’beautifully’, ’rid’, ’ignited’, ’crying’, ’shiny’, ’invented’, ’embarrassed’, ’submitted’, ’creeping’, ’islam’, ’collar’, ’bet’, ’cello’, ’disappointed’, ’worries’, ’buses’, ’tate’, ’index’, ’ill’, ’pain’, ’laid’, ’baking’, ’economists’, ’bitter’, ’solomon’, ’chest’, ’neighborhoods’, ’dj’, ’receives’, ’apart’, ’parade’, ’landing’, ’backward’, ’ron’, ’formula’, ’knee’, ’leather’, ’considers’

Appendix 5: Distributions of Emotion Words for Bingliu, Framenet and Wordnet.

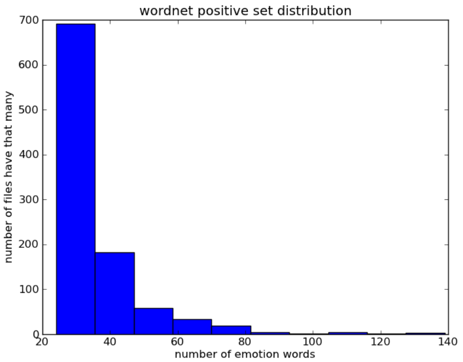


Figure 4: WordNet (Positive)

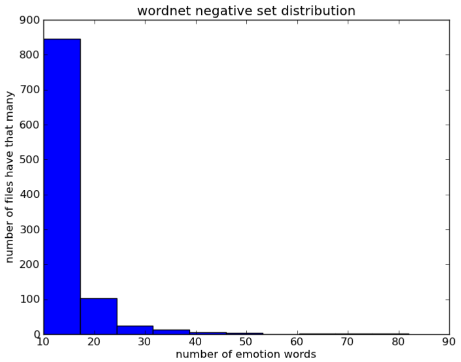


Figure 5: WordNet (Negative)

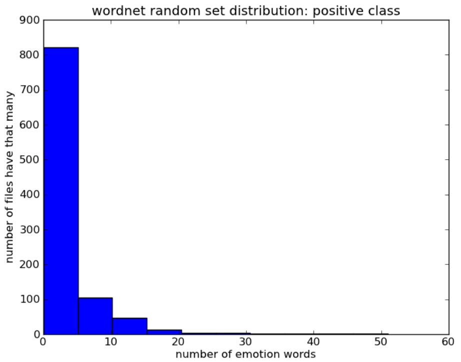


Figure 6: WordNet (Random)

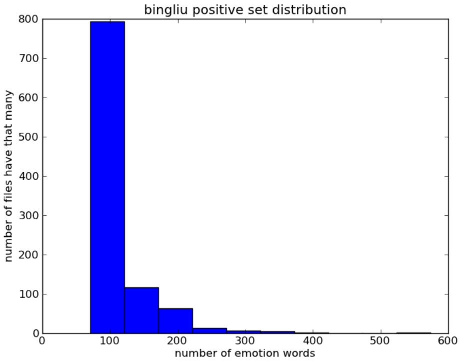


Figure 7: Bing Liu (Positive)

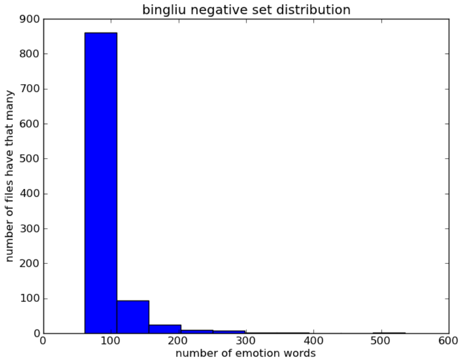


Figure 8: Bing Liu (Negative)

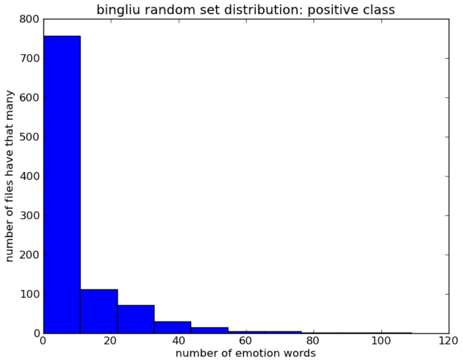


Figure 9: Bing Liu (Random)

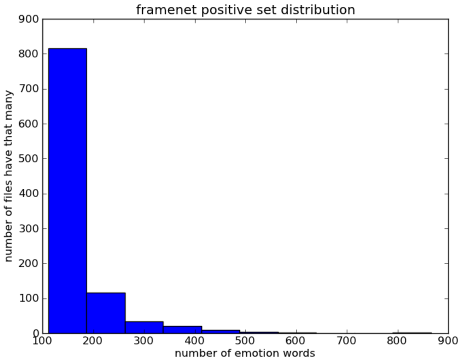


Figure 10: FrameNet (Positive)

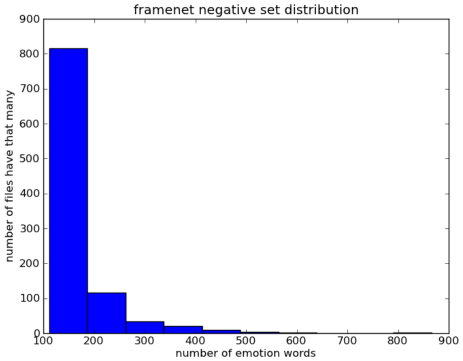


Figure 11: FrameNet (Negative)

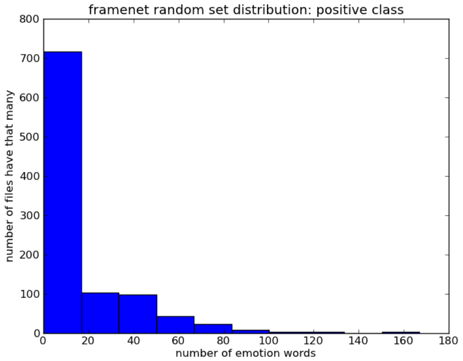


Figure 12: FrameNet (Random)

Appendix 6: soure Code

```
#####################################################
#!/usr/bin/python                                   #
# -*- encoding: utf-8 -*-                           #
#                                                   #
# CIS630 Project1: Spring 2013                      #
# Lexical correlates of affect and opinion in news  #
#                                                   #
# Copyright (c) 2013                                #
# - Yayang Tian  <yaytian@cis.upenn.edu>            #
# - Chen Ma      <machen1@seas.upenn.edu>           #
#####################################################

import re
from xml.etree import ElementTree
import os
import operator
import fnmatch
import math
import random
import matplotlib.pyplot as plt


""" Change it to your rootDir of New York Times"""
# nytDir = "/project/cis/nlp/data/corpora/nytimes/data/2001"
nytDir = "nytimes_2001"
mpqaDir = "lexicon/mpqa.txt"
bingliuDir = "lexicon/bingliu.txt"
# framenetDir ="lexicon/framenet.txt"
wordnetDir = "lexicon/wordnet.txt"

# global variables
allWords = []


def getWords(path):
    """Return all the words for the filepath of NY Times Corpus"""
    tree = ElementTree.parse(path)
    pTag = tree.findall(".//block[2]/p")
    fulltext = []
    for elem in pTag:
        fulltext.append(elem.text)
    pStr = " ".join(fulltext)
     # print pStr
    wordList = re.findall('\w+', pStr.lower())
    return wordList


def getPaths():
    """Return all the paths for the all the news"""
    paths = []
    for currentroot, dirnames, filenames in os.walk(nytDir):
        for filename in fnmatch.filter(filenames, '*.xml'):
            paths.append(os.path.join(currentroot,  filename))
            # print "file:\t" + filename
    return paths


def getArticle(path):
    """Return the article for the filepath of NY Times Corpus"""
    tree = ElementTree.parse(path)
    pTag = tree.findall(".//block[2]/p")
    fulltext = []
    for elem in pTag:
        fulltext.append(elem.text + "\n\n")
    pStr = " ".join(fulltext)
    return pStr


def getEmoWords(corpus, polarity):
    """Return the emotions words list"""

    if polarity == "random":
        polarity = "negative"
    words = []

    if corpus == "mpqa":
        lines = [line.strip().split() for line in open(mpqaDir)]

        for line in lines:
            if len(line) == 6:
                polar = line[5].split("=")[1]
                emoWord = line[2].split("=")[1]

            else:
                polar = line[6].split("=")[1]
                if (polar == "weakneg" or polar == "strongneg"):
                    polar = "negative"
                if line[3].split("=")[0] == "word1":
                    emoWord = line[3].split("=")[1]
                else:
                    emoWord = line[2].split("=")[1]

            if polar == polarity:  # NEED TO CLARIFY #
            # if polar == polarity or polar == "both":
                    words.append(emoWord)

    if corpus == "bingliu":
        lexiconPath = "lexicon/bingliu/" + polarity+"-words.txt"
        words = open(lexiconPath).read().strip().split("\n")

    if corpus == "wordnet":
        if polarity == "positive":
            files = ["joy.txt", "surprise.txt"]
        if polarity == "negative":
            files = ["disgust.txt", "fear.txt", "sadness.txt"]

        for lex in files:
            lexPath = "lexicon/wordnet/" + lex
            words = words + [word for line in open(lexPath) for word in line.strip().split(" ")[1:]]

    if corpus == "framenet":
        words = [word.split(".")[0] for line in open("lexicon/framenet.txt")
                 for word in line.strip().split() if not word.startswith("(")]

    return set(words)


def writeFile(corpus, polarity):
    """
    Write 1000 files containing most emotion words for each of:
    1. positive 2. negative 3. random
    """
    # HashMap: {fileid -> # files containing emotion words}
    freqFile = {}

    if polarity == "random":
        randomSet = random.sample(getPaths(), 1000)
        for path in randomSet:
            writePath = "nytimes_emotion/" + corpus + "/" + polarity + "/" + path.split("/")[-1]
            print writePath
            out = open(writePath, ’w+’)
            out.write(getArticle(path).encode("ascii", "ignore"))

    else:

        print "2. Calculating frequency of emotion words in " + corpus + " with polarity: " +
                                                             polarity + "..."
        emoWords = getEmoWords(corpus, polarity)
        for filename in getPaths():
            allwords = getWords(filename)
            print "extracting file: " + filename
            # print sum(allwords.count(emo) for emo in emoWords)
            freq = sum(allwords.count(emo) for emo in emoWords)

            if freq > 50:
                freqFile[filename] = freq

        # Sort 1000 most frequent files
        sortedFile = sorted(freqFile.items(), key=operator.itemgetter(1), reverse=True)[:1000]
        print sortedFile

        # write to files
        print "3. Writing to files..."
        for elem in sortedFile:
            articlePath = elem[0]
            writePath = "nytimes_emotion/" + corpus + "/" + polarity + "/" + str(elem[1]) + "_" +
                                                                     articlePath.split("/")[-1]
            out = open(writePath, 'w+')
            out.write(getArticle(articlePath).encode("ascii", "ignore"))


def getContexts(corpus, polarity):
    """ This extract all the 20-grams excluding the target emotion word"""
    """ It is similar to the NEAR operator (Turney, 2002) """
    print "Calculating 21grams......"

    contexts = []

    # # read 1000 articles containing abundant emotion words
    freqArticleRoot = os.path.join("nytimes_emotion", corpus, polarity)

    emoWords = getEmoWords(corpus, polarity)
    global allWords

    for filename in os.listdir(freqArticleRoot)[0:1000]:
        print "calculating context of:\t" + filename
        path = os.path.join(freqArticleRoot, filename)
        articleWords = re.findall('\w+', open(path).read().lower())

        allWords = articleWords + allWords
        for wid in range(len(articleWords)):
            if articleWords[wid] in emoWords:
                # extract 20-grams contexts excluding target emotion word
                contexts.append(articleWords[wid - 10: wid] + articleWords[wid + 1: wid + 11])

    return contexts


def calculatePMI(corpus, polarity):
    """ This calculate the PMI from the calculated contexts words"""

    PMI = {}

    contexts = getContexts(corpus, polarity)
    emoWords = getEmoWords(corpus, polarity)
    numEmotion = len(contexts)
    contextWords = set([word for ct in contexts for word in ct])
    global allWords
    numTotalWords = len(allWords)

    for seedWord in contextWords:
        print "Calculating PMI:\t" + seedWord
        numWord = allWords.count(seedWord)

        #Only pick words with freq > = 5
        if numWord >= 5:
            numCooccur = sum([1 for ct in contexts if seedWord in ct])

            ##################################################################
            # PMI = log2[ N * #co-occur / (#word * #emotion+) ]
            value = float(numTotalWords * numCooccur) / (numWord * numEmotion)
            PMI[seedWord] = math.log(value, 2)
            #################################################################

    # Rank PMI
    sortedPMI = sorted(PMI.items(), key=operator.itemgetter(1), reverse=True)

    # Output High PMI words NOT in the lexicon
    writePath = os.path.join("lexicon", "newLexicon", corpus + "_" + polarity)
    out = open(writePath, "w+")
    out.write("New " + polarity + " Word\t\tPMI\n")
    for item in sortedPMI:
        if (item[0]) not in emoWords:
            out.write(item[0] + "\t\t" + str(item[1]) + "\n")
```

```python
def plot(corpus, polarity):
    emoCount = []
    emoWords = getEmoWords(corpus, polarity)
    setRoot = os.path.join("nytimes_emotion", corpus, polarity)

    for filename in os.listdir(setRoot):
        # if filename.endswith(".xml"):
        #     emoCount.append(int(filename.split("_")[0]))
        articleWords = re.findall('\w+', open(setRoot + "/" + filename).read().lower())
        freq = sum(articleWords.count(emo) for emo in emoWords)
        print filename + "\t" + str(freq)
        emoCount.append(freq)

    print emoCount

    plt.hist(emoCount)
    plt.title(corpus + " " + polarity + " set distribution: negative class")
    plt.xlabel("number of emotion words")
    plt.ylabel("number of files have that many")
    plt.show()


def getNewLexWords(filename):
    lexRoot = os.path.join("lexicon/newLexicon", filename)
    # num of emo word
    # print sum([1 for line in open(lexRoot)])

    # num of overlap
    wordslist = [line.strip().split()[0] for line in open(lexRoot)]
    # print lexRoot
    # print a
    return wordslist


def main():
    corpus = "framenet"
    polarity = "random"

    # writeFile(corpus, polarity)
    # calculatePMI(corpus, polarity)
    plot(corpus, polarity)
    # mpqa = getNewLexWords("mpqa_random_negative")[0:100]
    # bingliu = getNewLexWords("bingliu_random_negative")[0:100]
    # framenet = getNewLexWords("framenet_random_negative")[0:100]
    # wordnet = getNewLexWords("wordnet_random_negative")[0:100]

if __name__ == "__main__":
    main()
```