The recommended approach to find the best sentences is to first cluster sentences and then pick the best sentences, in the best order, from the most important clusters.

When developing your solution, be sure to discuss BestSummary and then to document how you take that into consideration. One question is: What sentences should be the input to the clustering?

We first tried to cluster sentences directly from the corpus using ScitKit Learn. We extracted individual sentences and appended them into a single large file (47MB, 403,313 sentences), annotating each with an identifier for the file and the sentence sequence number. See sentences.txt. We eliminated any sentences less than 3 words or longer than 50 words, since empirical observation showed that these lengths usually indicated a problem with the sentence tokenization.

In [2]:

```
import sys
sys.path.append('../src')
import file_utils
import solr
from nltk.tokenize import sent_tokenize
import nltk

def collectSentences():
    file_names = file_utils.getFileNames('/Users/rgruss/CourseWork/CS 4984--
Computational Linguistics/corpora/clean/YourBigger')
    file_out = open('sentences.txt', "w")

    for file_name in file_names:
        file_in =  open(file_name, 'r')
        file_id = (file_name.split('/')[-1]).split('.')[0]
        text = file_in.read()
        sent_tokenize_list = sent_tokenize(text)
        sent_id = 1
        for sent in sent_tokenize_list:
            sent = ' '.join(sent.split())
            sent_len = len(nltk.word_tokenize(sent))
            if sent_len > 3 and sent_len < 50:
                file_out.write("%s-%i:%s\n" % (file_id, sent_id, sent))
                sent_id+=1
```

The code to cluster sentences in below. We tried two different features sets: 1) the TfidfVectorizer that comes with SciKit Learn, and a self-made feature vector consisting of the top words, bigrams, and named entities that occurred in our corpus.

In [3]:

```
from sklearn.cluster import KMeans
import numpy as np
import pandas as pd
import collections
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.base import TransformerMixin
from random import shuffle

MAX_SENTENCES = 200000
key_phrases = []

def readKeyPhrases():
    file_in =  open('best_words_entities.txt', 'r')
    for phrase in file_in.readlines():
        key_phrases.append(phrase.rstrip())

def getSentences():
    sentence_list = []
    file_in =  open('sentences.txt', 'r')
    count = 0
    for sent in file_in.readlines():
        count+=1
        sentence_list.append(sent.split(':')[1])
        if count > MAX_SENTENCES:
            break
    shuffle(sentence_list)
    return sentence_list

def getSentenceFeatureVector(sent):
    feature_vector = []
    for phrase in key_phrases:
        if phrase in sent:
            feature_vector.append(1)
        else:
            feature_vector.append(0)
    return feature_vector


def getSentenceFeatures1(sentences):
    readKeyPhrases()
    sentence_features = []
    for sent in sentences:
        sentence_features.append( getSentenceFeatureVector(sent))

    return np.array(sentence_features)

def getSentenceFeatures2(sentences):
    vectorizer = TfidfVectorizer(min_df = 1, max_df = 0.9)
    return vectorizer.fit_transform(sentences)

def main():

    sentences = getSentences()
    print "Got %i sentences" % len(sentences)

    sentence_features = getSentenceFeatures1(sentences)
    print "Got sentence features: "

    clf = KMeans(init='k-means++', n_clusters=8, random_state=42)
    distances = clf.fit_transform(sentence_features)

    print "Got centroids"

    #print clf.cluster_centers_
    #print clf.labels_
    #print distances

    cluster_sentence_distances = collections.defaultdict(list)

    for i in range(0, sentence_features.shape[0]):
        cluster =  clf.labels_[i]

        cluster_sentence_distances[cluster].append((i,  distances[i, cluster]))


    #for each cluster, print the closest sentences
    for clusterid, distlist in cluster_sentence_distances.items():
        cluster_sentence_distances[clusterid] = sorted(distlist, key=lambda tup: tup[1])
        print "Closest to cluster %i : %s" % (clusterid, sentences[cluster_sentence_distances[clusterid][0][0]])
        print "Second Closest to cluster %i : %s" % (clusterid, sentences[cluster_sentence_distances[clusterid][1][0]])

    #print cluster_sentence_distances

    print "done"


if __name__=='__main__':
    from timeit import Timer
    t = Timer("main()", "from __main__ import main")
    print t.timeit(1)
```

```
Got 200001 sentences
Got sentence features:
Got centroids
Closest to cluster 0 : People wait outside a new Ebola treatment center in M
onrovia on Tuesday, September 23.

Second Closest to cluster 0 : A doctor puts on protective gear at the treatm
ent center in Kailahun on Sunday, July 20.

Closest to cluster 1 : "We must fight Ebola because there is huge anxiety fo
r our populations along with significant social and economic consequences,"
Younoussa Ballo, secretary-general of Guinea's health ministry, told Reuters
 at Thursday's talks.

Second Closest to cluster 1 : It is more likely to be a result of the combin
ation of dysfunctional health systems, international indifference, high popu
lation mobility, local customs, densely populated capitals, and lack of trus
t in authorities after years of armed conflict.


Closest to cluster 2 : Well Virginia, It's the limits of epidemiology.

Second Closest to cluster 2 : "And it doesn’t have to be this way.

Closest to cluster 3 : Special objects, animals and region-specific products
 from infected areas are forbidden to enter China, the NHFPC said.

Second Closest to cluster 3 : Exotic infections are no longer relegated to r
emote parts of the world.

Closest to cluster 4 : The toll in West Africa continued to mount.

Second Closest to cluster 4 : Ebola in West Africa
Closest to cluster 5 : Nigerian Born,UK Based Model and Female Entreprene...

Second Closest to cluster 5 : Nigeria has seen its GDP growth rate increase
from 3.64 percent in the first quarter of 2013 to 6.77 percent in the same q
uarter of 2014, but Ebola -- along with the Boko Haram threat -- could lead
to a slowdown.

Closest to cluster 6 : Chan said the virus was "still running ahead, jumping
 over everything we put in place to slow it down".

Second Closest to cluster 6 : Remember this is a RNA virus.

Closest to cluster 7 : "Liberians are facing their gravest threat since thei
r war," Landgren told the UN Security Council on Tuesday.

Second Closest to cluster 7 : Meet The Men With The Most Dangerous Job In Eb
ola-Ridden LiberiaMeet The Men With The Most Dangerous Job In Ebola-Ridden L
iberia This BuzzFeed homepage is tailored for our readers in the USA.

done
17.7326750755
```

Results were disappointing. We were hoping for clearly delineated topics within the clusters, but wound up with sentences that shared only superficial similarities.

Another question is: How to cluster the sentences?

A second approach that yielded better results involved some pre-selection. First, we tokenzed the entire corpus on paragraph boundaries (resulting in 175,124 paragraphs). Then we selected the 1000 most relevant paragraphs (relevance defined as containing the highest number of the top words, bigrams, and entities), and then clustered the sentences from these paragraphs (total sentences 25422). Results were better, because they were all relevant to our event, but still weren't clustered around distinct latent topics.

In [4]:

```
#treat best_words_entities as a query, and take paragraphs in descending ord
er of relevance
#relevance measure: sum of bits in feature vector

from sklearn.cluster import KMeans
import numpy as np
import pandas as pd
import collections
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.base import TransformerMixin
from random import shuffle

key_phrases = []

def readKeyPhrases():
    file_in =  open('best_words_entities.txt', 'r')
    for phrase in file_in.readlines():
        key_phrases.append(phrase.rstrip())

def getParagraphs():
    paragraph_list = []
    file_in =  open('paragraphs.txt', 'r')
    count = 0
    for sent in file_in.readlines():
        count+=1
        paragraph_list.append(sent.split(':')[1])
    shuffle(paragraph_list)
    return paragraph_list

def getParagraphScore(paragraph):
    score = 0
    for phrase in key_phrases:
       if phrase in paragraph:
           score+=1
    return score

def getParagraphScores(paragraphs):
    readKeyPhrases()
    paragraph_scores = []
    for i in range(0, len(paragraphs)):
```

```python
        paragraph_scores.append( (i, getParagraphScore(paragraphs[i])) )

    return paragraph_scores

def main():

    paragraphs = getParagraphs()
    print "Got %i paragraphs" % len(paragraphs)

    paragraph_scores = getParagraphScores(paragraphs)
    print "Got paragraph scores "

    paragraph_scores.sort(key=lambda tup: tup[1], reverse=True)

    file_out = open('best-paragraphs.txt', "w")

    for i,score in paragraph_scores[:1000]:
        file_out.write(paragraphs[i] + "\n")

    print "done"


if __name__=='__main__':
    from timeit import Timer
    t = Timer("main()", "from __main__ import main")
    print t.timeit(1)
```

```
Got 175124 paragraphs
Got paragraph scores
done
3.30694198608
```

Closest to cluster 0 : I'm grateful to the team that worked tirelessly -- keep me alive in Liberia. Second Closest to cluster 0 : "Quite honestly if you ask 'can we stamp Ebola out of Liberia? Third Closest to cluster 0 : Liberia accounts for 576 of those fatalities.

Closest to cluster 1 : The issue created problems for Emory University Hospital in Atlanta, the first institution to care for Ebola patients here. Second Closest to cluster 1 : They also have limited capacity to safely bury bodies, and to provide ambulance services to refer patients. Third Closest to cluster 1 : After two weeks of intervention, the team currently has 137 suspected Ebola patients in its care.

Closest to cluster 2 : Hopefully, the video will also appear as a reassuring gesture to West Africans of the intentions and goals of the local and international health community at a time of urgent need. Second Closest to cluster 2 : " "These West Africa countries are unlucky in the sense they have not invested very much in the health system like Uganda has done. Third Closest to cluster 2 : " "These West Africa countries are unlucky in the sense they have not invested very much in the health system like Uganda has done.

Closest to cluster 3 : "When Kelly returned to San Francisco last week, he almost immediately requested a leave of absence from UCSF. Second Closest to cluster 3 : USAID also announced it would invest an extra $12.45 million to support the fight against Ebola. Third Closest to cluster 3 : " In the three hardest-hit countries there was a "mixed pattern", Dye said.

Closest to cluster 4 : There’s a decidedly Stephen King-ish aura around this virus, no doubt the result of those horrific bleed-out scenes depicted in popular books and movies. Second Closest to cluster 4 : They selected the concoction now called ZMapp and gave it to three groups of six monkeys; all received intramuscular injections of high doses of Ebola virus. Third Closest to cluster 4 : Meanwhile, the virus had slipped out of the village.

Closest to cluster 5 : He added that the man appeared to have a good chance of recovering.The man had been under surveillance by health authorities in Guinea because of his contact with Ebola victims but escaped to Senegal, Seck said.Residents in Dakar reacted with anger and concern. Second Closest to cluster 5 : On January 26, officials at the prefectural health authority held a meeting in Guéckédou. Third Closest to cluster 5 : Laurie Garrett, a senior fellow for global health at the Council on Foreign Relations, said her initial reaction when told that the U.S. military would become involved was "Hallelujah – the cavalry's coming.

Closest to cluster 6 : “Since we inherited this thing from Guinea, people were going with the feeling that there is a motive attached to it,” he said in a phone interview. Second Closest to cluster 6 : Few people are working. Third Closest to cluster 6 : But the current situation is different because the experimental Ebola medicines have not been as thoroughly tested in people.

Closest to cluster 7 : Widows for Peace through Democracy has partner organisations throughout West Africa, and supports them in their mission to help widows who face ostracisation, violence and destitution. Second Closest to cluster 7 : And increased awareness of the situation on the ground in West Africa. Third Closest to cluster 7 : He asked for his Bible and he read Psalm 91 and found comfort and inspiration in the words that surely shall deliver – the Lord shall deliver us “from the snare of the fowler, and from the noisome pestilence.” It was no exaggeration to suggest that tragically, we are facing West Africa’s pestilence now and potentially even broader if we don’t take the right steps.

Another question is: Which of the clusters should be considered when selecting sentences for the summary?

The clusters could easily be ranked by taking the nearest three sentences from each and assigning relevance scores based, once again, on top words, bigrams, and named entities.

Another question is: How should the selected sentences be ordered?

Although we included the top dates that emerged from our named entity recognition, dates didn't seem to show up in our nearest sentences. If we were to order these sentences, we would use the annotations to trace them back to their original document and try to find a date.

Explain what criteria you consider when selecting the best sentences and best order, why you chose those, and how your solution optimizes for those criteria, to find a BestSummary.

The template for our summary, which we made several weeks ago and have kept in mind throughout these last units, look like this: An outbreak of has struck , killing children and adults and hospitalizing . spreads and the government has declared a . Volunteers from the are bringing medical supplies, and victims are being treated at . So far, there have been of cases reported. Authorities are urging residents to , and are . Symptoms of include and typically those infected face . Treatments include . Those traveling in are urged to avoid . Each element of the summary is about a different topic: extent, location, government action, and treatment. Unfortunately, neither LDA nor clustering has enabled us to automatically locate sentences firmly within these topics. Probably the best approach will be to use each sentence of our summary as a prototype and find sentences most similar, using cosine distance.

```
In [ ]:
```