CS 224N Final Project: Unsupervised Clustering of People, Places, and Organizations in Wikileaks Cables with NLP Cues

Xuwen Cao, Beyang Liu

March 11, 2011

1 Purpose and Overview

Our goal is to extract the names of key entities from written U.S. diplomatic communications and then to apply natural-language- and sentiment-based clustering to these entities using contextual features extracted from the data. We extract entities using an off-the-shelf statistical NLP package, and then seek to generate meaningful clusters of entities in an unsupervised fashion. To do so, we experiment with different models, feature sets, and clustering algorithms.

For the purposes of this project, we define key entities to be people, nations, and organizations that occur at least k times in the dataset. We determined k through empirical evaluation of the entities extracted from the data, and also the need of different scenarios.

2 Data

Our data is the set of American diplomatic cables published by Wikileaks (`http://mirror.wikileaks.info/`) as of 10 February 2011 as part of its Cablegate release. It is available for public download as a torrent on the Cablegate website (`http://213.251.145.96/cablegate.html`). The data encompasses 3891 cables, mostly sent to and from American embassies abroad by U.S. State Department officials. The cables, which were initially intended only for internal State Department use, encompass 7 levels of classification - "CONFIDENTIAL," "CONFIDENTIAL/NOFORN," "SECRET," "SECRET/NOFORN," "UNCLASSIFIED," "UNCLASSIFIED/FORN," and "OFFICIAL USE ONLY." (The designation "NOFORN" indicates that the cable should not be viewed by foreign eyes). Each cable includes a header indicating the identification number of the cable, the date of its transmission, its classification level, the sender (usually an embassy or U.S. government office in Washington D.C.), the recipient (usually a set of embassies and/or government offices in Washington D.C.), tags indicating subject matter, a subject line, and finally, the body. The contents of the cables include many frank assessments of foreign governments, individuals, and organizations by State Department officials. They also shed light on American diplomatic tactics and present accounts of events previously unreported in the media. An overview of the revelations provided by the release of the cables can be found at: `http://www.nytimes.com/2010/11/29/world/29cables.html`.

3 Stanford NLP Package Components

The Stanford NLP group offers an integrated suite of core Natural Language Processing tools (`http://nlp.stanford.edu/software/corenlp.shtml`), which we used for this project. We used the Named Entity Recognition (NER) and Part of Speech Tagging (POS) components of the package.

3.1 Named Entity Recognition

The NER component utilizes a Conditional Random Field (CRF) classifier [3], and identifies the following categories of amed entities: PERSON, LOCATION, ORGANIZATION, MISC. It uses a combination of

three CRF sequence taggers trained on various corpora. Our program shows that there are (lexically unique) 21584 people, 7754 locations and 24891 organizations in leaked cables.

3.2 Part-of-Speech Tagging

We utilize a maximum entropy POS tagger [5] to assign each word in the data one of the POS labels from the Penn Treebank tag set [4].

4 Model: K-means Clustering in Sentiment/Frequency Space

4.1 Hypothesis

We choose sentiment scores and entity frequency as our two major features in K-means. In our data analysis, we assume two weak correlations - firstly, the sentiment of adjectives reflect US's attitude towards a given entity; secondly, the frequency of entity appearance in the cables correlates to its importance to US. Therefore these two characteristics will allow us to make interesting discovery about US diplomacy. Moreover, the choice will allow us to evaluate clustering result extrinsically.

4.2 Data Preparation

After we process the cables with Stanford NLP Package, we obtain XML files of original cable annotated with named entity labels and POS tags. We compress these files for easier text processing, example:

```
The[the,DT,O]Political[Political,NNP,O]Director[Director,NNP,O]
emphasized[emphasize,VBD,O],[,,,,O]however[however,RB,O],[,,,,O]
that[that,IN,O]Abbas[Abbas,NNP,PERSON]'[',POS,O]remark[remark,NN,O]
was[be,VBD,O]meant[mean,VBN,O]only[only,RB,O]as[as,IN,O]an[a,DT,O]
example[example,NN,O],[,,,,O]and[and,CC,O]not[not,RB,O]as[as,IN,O]
an[a,DT,O]explicit[explicit,JJ,O]suggestion[suggestion,NN,O].[.,.,O]
```

For simplicity, we focus on the LOCATION entities(countries, regions and cities) We extract every word tagged both as LOCATION and noun (NN, NNS, NNP, NNPS) and record the adjectives (POS tagged 'JJ', 'JJR' or 'JJS') in the sentences containing the LOCATION word. After we set up a dictionary of keys(LOCATION) and values(adjectives), we compute a sentiment score for every adjectives collected based on SentiWordNet (`http://sentiwordnet.isti.cnr.it/`). The SentiWordNet annotates each word with three scores in the intervals [0,1], namely positivity, negativity and neutrality [1]. In our case, we compute the sentiment score as the difference of positivity and negativity.

For each of the dictionary keys (LOCATION), a weighted average is obtained for all adjectives with non-zero sentiment score. Only non-zero sentiment scores are included as the vocabulary size of SentiWordNet is limited and the LOCATION words that appear more frequently will be penalized more heavily if most adjectives are assigned zero scores. Since all the cables are related to US to some extent, we calibrate the mean of dataset by the scores attributed to US (location entity 'u.s.', 'us', 'united states' and 'usa'), which is around -0.02. In the last step, we normalize the scores by the maximum absolute score so that the data points are more sparse in the interval [-1, 1].

For the entity frequency feature, we simply keep counts of how many times a given LOCATION appear across all cables. Due to the power law distribution, the majority of frequencies clutter at the lower end (zero), thus we used the logarithm and normalized the log of frequencies to interval [0, 1]. Moreover, to only include the sentiment scores that are statistically sound, we used a threshold for the appearance frequency, empirically determined as 200.

4.3 K-means

After the data preparation step, a 2D space of sentiment scores and entity frequencies is obtained. We use K-means for clustering the location entities by Euclidean distance. As discussed in our hypothesis, we intended to separate the entities by two criteria - importance and attitude. The importance is related

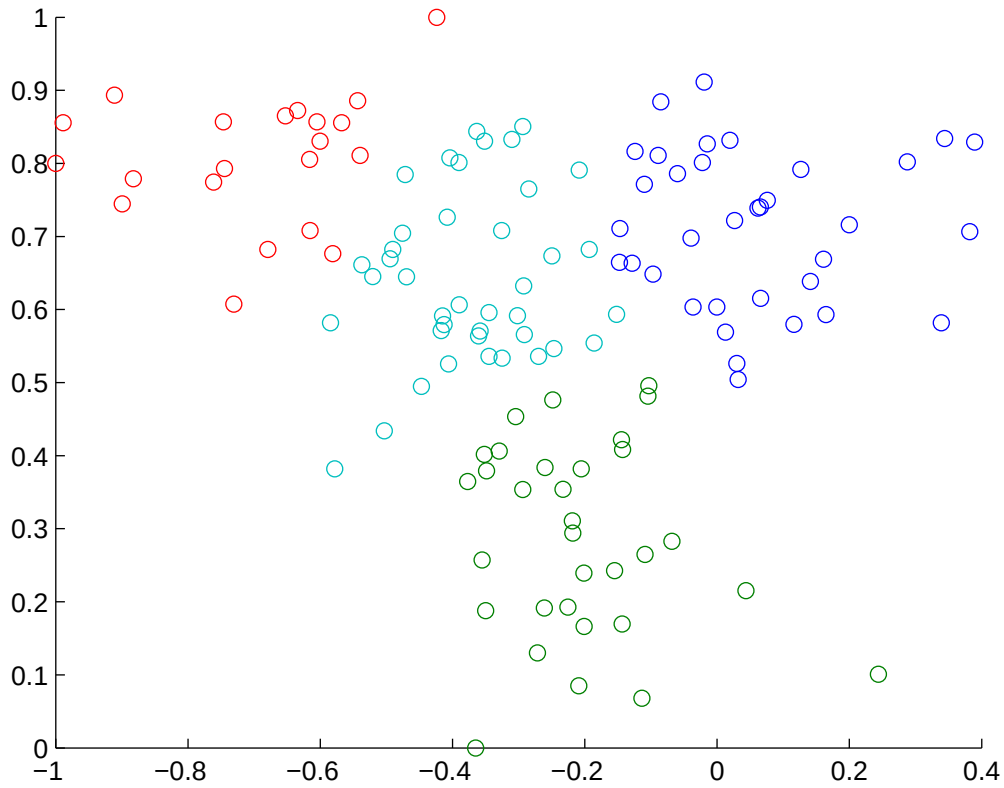


Figure 1: K-Means Clusters

to the frequency while attitude correlates to sentiment score. So ideally we want to have at least four clusters (combination of important/unimportant, friendly/hostile).

Our K-means algorithm is standard. We initialize by setting four random labels (location entities) as cluster centers. In each iterative step, we assign all labels to their closet cluster (L2 distance to cluster center); then we compute the mean of all data points in a given cluster. This process continues until the result converges. Due to the low dimentionality and small data size, the iteration steps are very fast. Although k-means algorithm suffers from the local minima problem, we were able to run the algorithm many times to ensure the output is relatively stable.

4.4 Result

Due to the overall triangular shape of the data points, we can't obtain the ideal clustering as discussed in last section. But still we seperate the points into four meaningful categories.

A brief explanation about the graph: the horizontal axis is the sentiment scores, the sentiment is more positive along the right; the vertical axis is the frequency, the more frequent points are at the bottom. Then it becomes clear what each cluster represents:

```
green cluster:
['london', 'paris', 'cuba', 'africa', 'brasilia', 'cairo', 'eu', 'brazil',
 'afghanistan', 'egypt', 'europe', 'iran', 'china', 'iraq', 'libya', 'syria',
 'pakistan', 'washington', 'turkey', 'israel', 'moscow', 'spain', 'uk', 'russia',
 'madrid', 'india', 'tripoli', 'kabul', 'iceland', 'france']
```

The green group appear most frequent in the cables, and has a medium sentiment score. If we assume that the cables are leaked independent of their content, this cluster represents the highest importance to US's interest (at least from a diplomatic sense).

```
red cluster:
['djibouti', 'taiwan', 'tajikistan', 'islam', 'mumbai', 'zimbabwe', 'dubai', 'goa',
'tibet', 'armenia', 'yar', 'ecuador', 'benghazi', 'algiers', 'yemen', 'paraguay',
'caracas', 'south africa', 'ouagadougou', 'xxxxxxxxxxxx', 'guinea']
```

The red group has the lowest sentiment score. Assuming the US embassies across world have relatively similar standard of language (also we try to eliminate some statistical anomalies in our data preparation), then we can infer this is group of countries/regions that US is most disapproval of. Also we note because of the slight defect in NER, things like 'islam' and 'xxxxxxx' are recognized as location entities.

```
teal cluster:
['kosovo', 'north korea', 'damascus', 'argentina', 'latin america', 'netherlands',
'uruzgan', 'switzerland', 'reykjavik', 'lebanon', 'qatar', 'sudan', 'somalia',
'venezuela', 'guantanamo', 'colombia', 'sao paulo', 'saudi arabia', 'america',
'peru', 'gaza', 'bolivia', 'ukraine', 'geneva', 'jordan', 'tehran', 'georgia',
'sweden', 'portugal', 'mexico', 'lula', 'kenya', 'italy', 'ethiopia', 'canada',
'germany', 'havana', 'algeria']
```

The teal cluster has a medium negative sentiment score, and also a medium frequency. We infer that this group has relatively lower importance to US than the green group, although US is generally not happy with this group.

```
blue cluster:
['azerbaijan', 'japan', 'chechnya', 'norway', 'australia', 'ankara', 'baghdad',
'poland', 'haiti', 'kazakhstan', 'honduras', 'belgrade', 'copenhagen', 'kuwait',
'karzai', 'amazon', 'burma', 'tunisia', 'west bank', 'doha', 'west', 'new york',
'nigeria', 'serbia', 'darfur', 'chile', 'morocco', 'vatican', 'uae', 'new delhi',
'middle east', 'brussels']
```

The blue cluster has the highest sentiment score, which means that US is relatively happy with this group. As one may notice, there are a few notable anomalies such as 'burma' and 'sudan'. In the case of 'burma', the positive sentiment is mainly caused by Aung San Suu Kyi's release from house arrest from mutliple cables. In the case of 'sudan', it's also a special case because the darfur cables discuss mostly the international help darfur received, instead of it's dire situation.

We are making a few simplifying assumptions about our data - eg. the cables are leaked at random, the diplomats have relatively similar perception about countries, etc. However, we have constructed a pretty accurate model given existing cables. Apart from a few exceptions pointed out, the clustering generally correspond to people's perception of US's relation with the various parties. We can also verify our findings extrinscally using human reports about cablegate. For example, from New York Times:

```
on Belgium (brussels, blue cluster) - Belgium will take Guantanamo prisoner
on Italy (italy, teal cluster) - leaders' close ties with Russian discovered
on Germany (germany, teal cluster) - tensions with Germany for CIA officers arrest
on Syria (damascus, teal cluster) - arms deliveries from Syria to militants
```

4.4.1 Interesting Finding

Given our model, we made a few interesting discoveries:

1. In general, the US diplomats are critical of other countries, as we observe the majority of the data points is in the negative
2. Surprisingly, US's most important ally is spain (seen lower right quardrant)
3. US is most friendly with Norway (right-most point), although it's relatively unimportant

4. Iran appeared most frequently, with a small negative sentiment (which means the attitude is not always hostile)

5. US is least happy with Zimbabwe and Paraguay, although it doesn't care too much about them either

6. US doesn't actually have good relations with its traditional allies such as France, UK and Germany. Canada, Italy and Germany even scored lower than China.

4.5 Discussion

There are a few defects with our approach - firstly we only mechanically select the adjectives surrounding a sentence regardless of the dependencies (eg. the adjective is not describing our location entity) and semantic structure (eg. 'not' should flip the sentiment score). Our hope was that given enough data, this problem can be minimized and we can somewhat approximate the ideal result. If we can incorporate the Stanford parser and co-reference tool (it was taking way too long to train) in the future, it will easier for us to obtain a better result. Secondly our approach relies heavily on the SentWordNet, which in our opinion is not assigning the appropriate score (at least in the context of these cables). We may need to do a few supervised training to obtain a better semantic analysis program. Thirdly, we clustered all locations, including cities, countries and regions. We can try to merge some of the result to conduct a better comparison of perceptions. Lastly, we simplify the collection of words to a sentiment score, discarding many of the interesting features. For example, different entities can have different combinations of words but still compute to the same sentiment score. If we mine the hidden model using words as features, maybe even more interesting clustering can be discovered. The Multinomial Mixture Model addresses some of these problems.

5 Multinomial Mixture Model

Also known as Mixture of Naive Bayes, the Multinomial Mixture Model has been applied in a variety of classification settings, including bilingual text classification [2]. It models each data instance as a generative process in which the data instance label is selected from a multinomial distribution over the set of labels, and the observed data instance properties (i.e. features) are then selected independently, conditioned upon the selected label.

Formally, we define a data instance to be a set of random variables $(X, Y) \in \mathcal{X} \times \mathcal{Y}$, where the feature vector $X = (X_1, X_2, ..., X_n)$ is a vector of n multinomials and the label Y is a single multinomial. Let K be the number of labels, so $\mathcal{Y} = \{1, 2, ..., K\}$. We model the joint probability distribution as:

$$P(X, Y) = P(Y) \prod_{j=1}^{n} P(X_j|Y)$$

This corresponds to the Naive Bayes graphical model (Figure 2), which assumes that the features are independent of each other conditioned on the label.

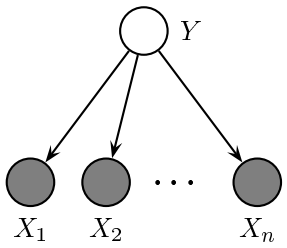


Figure 2: Naive Bayes Model

We then assume we are given a dataset of m examples, $D = \{(x^i, y^i)\}_{i=1}^{m}$, drawn from the distribution $P(X, Y)$. In the sequel, we denote the jth feature of data instance i (i.e. the jth component of vector

x^i) as x^i_j. From D, we would like to learn the parameters of our model (the conditional probability distribution parameters for each of the multinomial variables, i.e. $P(Y)$ and $P(X_j|Y), j = 1, ..., n$). In the sequel, we refer to these parameters as θ.

Given a dataset D in which the features and labels of each example are fully observed, we define the likelihood function to be the probability of the data under the model parameters. Accordingly, the likelihood function (which we want to maximize) is:

$$L(D;\theta) = \prod_{i=1}^{m} \left[P(y_i) \prod_{j=1}^{n} P(x^i_j|y_i) \right]$$

In an unsupervised setting, however, the label y is not observed at training time. Optimizing the likelihood function jointly over the possible values of y in the data and θ is a nonconvex problem. Instead of optimizing it directly, we optimize a convex lower bound using the Expectation-Maximization (EM) algorithm. In the E-step, we compute the expected fractional counts of each data instance i being drawn from each of the K labels. These counts are given as:

$$\begin{aligned} P(Y = k|X = x^i) &= \frac{P(X = x^i|Y = k)P(Y = k)}{P(X = x^i)} \\ &= \frac{P(X = x^i|Y = k)P(Y = k)}{\sum_{l=1}^{K} P(X = x^i|Y = l)P(Y = l)} \\ &= \frac{P(X = x^i|Y = k)}{\sum_{l=1}^{K} P(X = x^i|Y = l)} \end{aligned}$$

The last equality above holds if we assume $P(Y = k)$ is uniform.

In the M-step, we treat the y-values of the imputed fractional data instances as observed and find the MLE parameters of the model. Since the model is a Bayesian network, the MLE problem simply requires aggregating the fractional counts for each table CPD entry (i.e. computing the expected sufficient statistics). We note that in our case, we do not recompute $P(Y)$ in the M-step, instead fixing it to be uniform. This is because we would like to enforce a uniform prior over the size of each cluster, since roughly equal-sized clusters are desirable and we want to avoid the degenerate case in which one cluster swallows all examples.

Having defined the E-step and M-step, we iterate EM until convergence of the model parameters θ. We define convergence to be the event that the change in the maximum residual of θ decreases below some threshold ϵ.

5.1 Feature Extraction: NLP-informed word-presence indicators

Running the Stanford NER extractor yields 21664 people, 8182 places, and 25544 organizations. Obviously, some of these are duplicates, but we did not have a robust method of merging duplicates. (We considered a naive string-matching approach, but this would have merged people sharing the same last name and this was undesirable. We hoped instead that the clustering algorithm would cluster together entity names that referred to the same underlying entity).

Due to the limitation of this dataset size, we could not utilize a naive bag-of-words feature set, as this would have resulted in too many features as well as too sparsely occurring features. Accordingly, we considered the following features:

- Indicator features that indicate the label of the entity as predicted by the Stanford NER system. The possible labels are PERSON, PLACE, and ORGANIZATION.
- Features that indicate the presence of adjectives in a window around the given entity's locations in data. We determined which words were adjectives using the POS-tagger.
- Features that indicate the presence of other entities in a window around the given entity's locations in data.

In addition to using standard indicator features, which indicate the event in which a given word co-occurs with the entity at least once in the data, we also used thresholded indicator features. These indicate whether a given word co-occurred with the entity at least t_i times, where t_i is the threshold for feature i. We varied t_i in discrete intervals proportional to the empirical standard deviation of each feature around its empirical mean. This seemed to yield qualitatively better results in practice.

5.2 Initialization

One issue we encountered when testing the EM algorithm for the Multinomial Mixture Model was numerical issues when computing the log-likelihood objective. This led to a decrease the objective in successive iterations of EM (which is theoretically impossible as EM is guaranteed to improve the objective at every iteration). This phenomenon is illustrated by Figure 3(c). Related to this issue was the fact that most of the cluster sizes outputted by the algorithm were zero (we set the number of clusters to be 100 initially and 95 clusters were empty).

We hypothesized that this issue arose from the way in which parameters were being initialized. Our first approach was to initialize the parameters on the uniform distribution between 0 and 1. That is, we would initialize $P(X = 1|Y = k) \sim \text{Unif}(0, 1)$. However, closer inspection of the feature values in the dataset reveals that this method of initialization yields probabilities that are far too high. This is due to the fact that feature activation is very sparse. That is, not only does a given data instance contain relatively few active features (i.e. sparse feature vector), each feature is activated in relatively few data instances (i.e. sparse feature activation). Initializing our activation probabilities to be uniform on the interval [0, 1] thus resulted in gross initial overestimates of these parameters. This resulted in very low probabilities being assigned to data instances for most clusters. The few clusters that assigned higher probabilities to the data (because by random chance, more of their initial probabilities happened to be low) thus grew and continued to grow, as all remaining clusters were starved (Figure 3(a)).

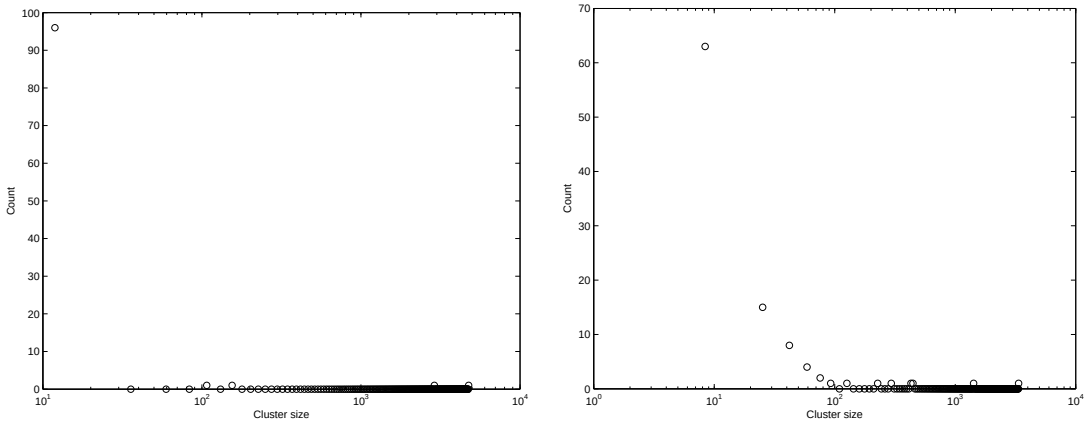
To rectify this problem, we changed the initialization scheme to initialize feature activation probabilities to be close to the empirical frequency of the feature in the dataset. Concretely, we initialized each feature activation probability $P(X_i = 1|Y = k) = \theta_i^k$ to be $\theta_i^k := \mathcal{N}(\mu_i, \sigma_i^2)$, where μ_i, σ_i^2 are the empirical mean and variance of feature i in the data D. Changing to this initialization scheme yielded much better results, as illustrated in Figures 3(b) and 3(d).

5.3 Results

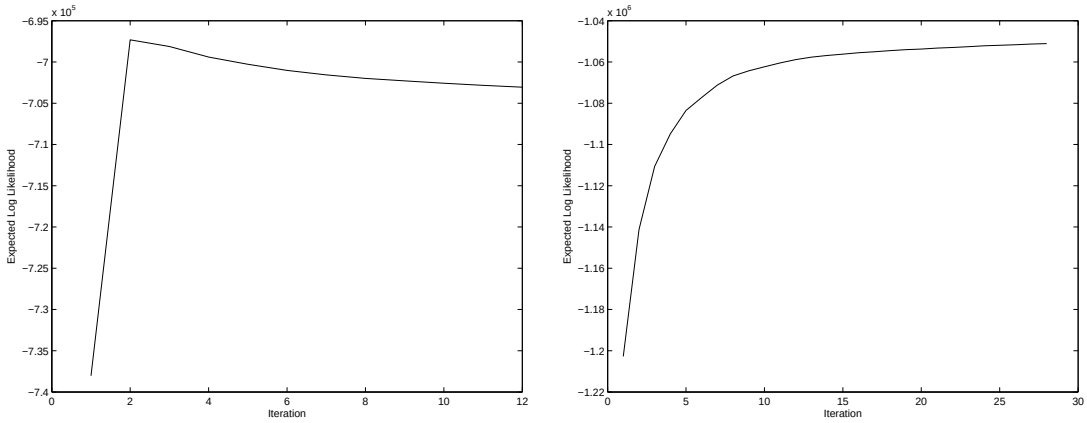
Full results are available in the directory cablegate/cable/output. The files are in the format output.<entity-threshold>.<feature-type>.<feature-activation-threshold>. The entity-threshold indicates the frequency below which we discarded entities from our our data. I.e., if a given entity detected did not occur more than <entity-threshold> times in the data, we ignored it and did not include it in the clustering algorithm. The feature-type indicates the types of features used in producing the output and feature-activation-threshold indicates the number of standard deviations above the mean rate of a word's occurrence required for its corresponding feature to be active (i.e. set to 1). As mentioned in the Feature Extraction subsection, we varied this value.

We present some illustrative examples of clusters found by our algorithm below. The parameters used to generate these clusters were entity-threshold = 100, feature-type = surrounding named entities, feature-activation-threshold = 0.

- Cluster 5 : helmand, karzai, seoul, brown, karzai, williams, tadic
- Cluster 59 tripoli dutch france abuja muammar al-qadhafi icc
- Cluster 70 tajikistan somalia khartoum ukraine colombia spain georgia yar uruguay al-qadhafi kouchner abdullah moratinos chirac leon sarkozy labor gos ftaa ipr ministry of defense ustr
- Cluster 78 libyan guatemala switzerland serbia saif al-islam el-jahmi bashir megrahi ruehtro de ruehtro gol qdf mre



(a) Histogram of Cluster Sizes after Naive Initialization (b) Histogram of Cluster Sizes after Smart Initialization



(c) Objective Over EM Run with Naive Initialization (d) Objective Over EM Run with Smart Initialization

Figure 3: The effect of the initialization scheme on EM performance

- Cluster 92 cairo iran saudi arabia west bank palestinian authority qatar middle east karachi maliki atmar ben ali saleh european union gbp eu icrc wto ahmadinejad
- Cluster 94 haiti goa asia cuba amorim biato lula gob zapatero farc

As one can see, the results are mixed. Some entities grouped together seem to make logical sense. For example, Cairo, Iran, the West Bank, and the Palestinian Authority seem tightly correlated. Furthermore, Ahmadinejad is the president of Iran. The coupling of Maliki (prime minister of Iraq) seems to reflect the importance of the relationship of Iran and Iraq to U.S. interests in the area and the coupling of Saif al-Islam (son of Muammar Qadhafi) with Libya makes sense. Other choices, however, seem less coherent. For example, it's unclear why Spain would be grouped together with Sarkozy, Uruguay, and Qadhafi (Cluster 70).

In addition, the results include numerous smaller clusters that seem to have no coherent internal logic. These seem to be "garbage collector" clusters, which pick up the scraps that don't fit neatly into any of the other clusters. By increasing the number of clusters, we could potentially increase the number of these garbage collectors and perhaps improve the coherence of the remaining clusters. Alternatively, the existence of these clusters could indicate a shortcoming of the Multinomial Mixture model, which we seek to address below.

6 Future Direction

6.1 Mixture of Log-Linear Clustering

There are two issues with the Multinomial Mixture Model that we would like to have addressed. The first is the fact that the model restricts features to be multinomial random variables. While this is fine for indicator features, we might want to incorporate real-valued features. In particular, word frequency (i.e. the percentage of times a word co-occurs with the given entity) might be a better feature by which to cluster than word presence (i.e. whether or not a word ever co-occurs with the given entity). One way to handle such features within the Multinomial Mixture Model is to bucket the real values into discrete levels. However, this introduces many arbitrary parameters (i.e. the number and ranges of the buckets) into the model. Picking good buckets requires studying the distribution of the feature values, and even then, the choice of buckets remains purely heuristic.

Secondly, the Multinomial Mixture Model makes conditional independence assumptions that may not hold in the data. In particular, it assumes that each feature is independent of every other features given the label. This is probably not true. For example, co-occurrence with the word "human" is likely highly correlated with the word "rights" even if the cluster label is known.

To address these 2 issues, we'd like to have explored the use of a Log-Linear Mixture Model. Concretely, we would define $P(Y = k|X) = \text{sigmoid}(\sum_{i=1}^{n} \theta_n^k X_n)$. Graphically, the model would be that of Figure 4. Note that utilizing this model would allow the use of arbitrary-valued features as well as eliminating the independence assumptions between features. The one caveat with this approach, however, would be that the M-step would be significantly more expensive. Concretely, the M-step requires inferring the MLE parameters for the model given the expected data. In the case of Naive Bayes, this was only as difficult as aggregating the expected sufficient statistics for each feature conditioned on the label. In the case of the logistic model, however, this learning scheme is not valid. To compute the MLE parameters, we would have to use some sort of iterative approach (e.g. gradient descent), which would take significantly longer and be sensitive to certain parameters (e.g. the learning rate).

Due to the lack of a readily available weighted logistic regression module to conduct learning using expected data, we would need to implement this on our own, and unfortunately time constraints prevented us from getting such a system up and running, much less optimized. Given more time, however, pursuing this model would be one of the primary next steps, for the 2 reasons listed above.

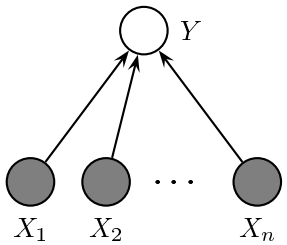


Figure 4: Log Linear Model

6.2 N-gram Feature Selection

Another extension we would like to pursue is the use of n-gram-based features. Multi-word phrases such as "human rights," "presidential election," and "foreign investors" are essentially single semantical units that have greater significance than the sum of their parts. One approach to extract these n-gram tokens would be to train an n-gram language model and collect all n-grams that are assigned high enough probability by the model. However, this approach is susceptible to grouping together words that simply occur together frequently, even if they are not a single semantic unit (e.g. "and then," "there were," etc.). Another approach would be to use the output of the Part-of-Speech tagger and chunk together nouns with preceding adjectives. A more sophisticated chunking scheme would utilize a lexicalized parser

to produce a parse tree and then group together words using the predicted parse tree (similar to the pre-chunking scheme in PA3).

7 Appendix: Running Our Code

The raw wikileaks cables are available in the directory cablegate/cable/all. The post-processed data, after having been annotated with the Stanford NLP tools is in the directory cablegate/cable/result.

Our code is in the directory cablegate/cable. To run the full pipeline for evaluating the Multinomial Mixture Model, use the script nbPipeline.sh, which will output results for a variety of different parameters. The discovered clusters will be dumped to the cablegate/cable/output directory (overwriting the outputted files that are currently there) in the format "output.<additional-meta-info>". Other intermediate files will also be dumped into the output directory. Because the script runs the full pipeline with different parameters, it will take a long time to complete. However, it is designed such that it can be parallelized. For example, if you run ./nbPipeline.sh from multiple unix screens or on multiple corns, it will detect runs that are already going and skip over them to runs that haven't started yet.

To run the K-Means clustering pipeline, just run kmean.py in cablegate/cable. To see the full pipeline from feature extraction, run kmPipeline.sh. To do so, you will need to change the absolute paths in the file finalproject/src/Sentiment.java

References

[1] Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Bente Maegaard, Joseph Mariani, Jan Odijk, Stelios Piperidis, Mike Rosner, and Daniel Tapias, editors, *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10)*, Valletta, Malta, may 2010. European Language Resources Association (ELRA).

[2] Jorge Civera, Alfons Juan, and Departament De Sistemes Inform tics. Multinomial mixture modelling for bilingual text classification. Technical report, 2005.

[3] Jenny Rose Finkel, Trond Grenager, and Christopher Manning. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL '05, pages 363–370, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics.

[4] Mitchell P. Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. Building a large annotated corpus of english: the penn treebank. *Comput. Linguist.*, 19:313–330, June 1993.

[5] Kristina Toutanova and Christopher D. Manning. Enriching the knowledge sources used in a maximum entropy part-of-speech tagger. In *Proceedings of the 2000 Joint SIGDAT conference on Empirical methods in natural language processing and very large corpora: held in conjunction with the 38th Annual Meeting of the Association for Computational Linguistics - Volume 13*, EMNLP '00, pages 63–70, Stroudsburg, PA, USA, 2000. Association for Computational Linguistics.