STAT 215B Final Project, Spring 2012

Christine Kuang, Siqi Wu, and Angie Zhu

May 2, 2012

1 Introduction

the largest microblogging website in China, Sina Weibo 新浪微博
A post can be text, an image, video, or other multimedia.
related works here

2 Methods

2.1 Data Collection

2.2 Tagging

In all, we tagged a total of 4000 Weibo posts. We made a search for the topic we were looking into and did this over time to get a total of 10,000 Weibo posts, from which we picked out 3000 posts as our training set. We each tagged 1000 posts. The final 1000 posts were for our test set and we tagged these together. We had a total of four different categories that we tagged the posts as - neutral, positive, negative, and irrelevant (spam).

As we each tagged the posts, we encountered and realized a few of the limitations involved. One limitation was in the fact that tagging these posts produced subjective responses. What one of us read as a negative response to the topic we chose, another may have read as a positive response. For example, the English phrase 'that wasn't too bad' could be taken as positive or negative. Positive - the experience was better than expected; negative - the experience wasn't great. Hence, it is difficult sometimes to truly know whether the original author of the post had in mind a negative or positive reaction to the topic he was posting about.

Another limitation was that some posts we were not sure how to tag. Many posts consisted of just a quote by the author we were looking into. These could have been seen as positive posts since the writer may have liked the quote and so posted it. But at the same time, the author could have been neutral and was just merely using the quote to apply to a specific circumstance in his life at the time. We were not entirely sure of how to tag each of the posts that fell into this category, and so again, the subjectiveness of tagging the posts by hand comes into play as a limitation in the forming of our model. Another uncertainty that occurred in trying to manually tag the posts was that some of the posts didn't talk about our chosen topic specifically, but a related topic. With our chosen topic, people who had negative responses towards 韩寒 (Hanhan), were usually on the side of the opposing author who was discrediting him. Some of the posts did not directly mention 韩寒, but would instead show support for the opposing side. With these posts, we typically labeled as a negative response. However, an argument could be made for just throwing out those posts since they do not directly say anything about 韩寒, and they could possibly just be supporting the opposing author in his own literary works and not necessarily in his stance against 韩寒. Another uncertainty in tagging is what to tag posts that have no subject. There were a few posts that had nothing to do with the chosen topic at all, but there were also posts that had no subject but contained phrases such as ”Keep it up!”, ”always a supporter!”, etc. that could very well be taken as positive posts since our search is for our specific topic. But because there is no subject in the posts, this cannot be taken with 100% certainty. In these instances, where there is no subject in the posts, we marked them as irrelevant to be on the conservative side in our predictions.

These limitations in tagging will affect our model's accuracy in predicting whether a post contains a positive or negative response to the topic at hand. These limitations also indicate to us that in general, models for predicting whether or not a post is negative or positive towards a chosen topic is limited greatly by the subjectiveness of the sentence's meaning/interpretation which affects the training set used to form the model. This limitation could present a potential future question to look into on how we can improve tagging - whether we should just remove posts that have too subjective of an interpretation to tag or if by studying these types of ambiguous posts, we can create certain priors to help in the tagging process.

2.3 Processing

UTF-8 encoding, GBK, Unicode, Big5

2.3.1 Characteristics of Chinese Language

No explicit delimiter between words in Chinese texts
OOV
ambiguity
[7]

There has been quite a bit of research done into natural language processing in the English language, but not much on the Chinese language. This is due to the fact that the Chinese language contains unique characteristics that makes it difficult to do natural language processing well.

First, the Chinese language, unlike the English language, has no explicit delimiter between words. In the English language, spaces separate words and so to segment a sentence into its appropriate components is not too difficult since one could just separate based on spaces. The Chinese language however has no such delimiter between words.

An additional difficulty in segmenting a sentence into its appropriate components and words is that the Chinese languages made of separate characters where each character has a meaning of its own, but if you combine two or more characters together into a phrase, the meaning can be completely different. These types of ambiguities are typically easily resolved by humans reading the sentence and realizing what would make most sense in terms of the context of the sentence, but it is not so easy for a computer to preform those types of automatic word/phrase segmentations. For example, 他好吃 - this sentence could be separated in two ways. The first character means 'he'. The other two words can be taken as two different phrases: 'tastes delicious' or 'loves to eat'. It is obvious that the phrase should be taken as ”he loves to eat” because ”he tastes delicious” does not make sense but it is difficult to have a computer automatically recognize which sentence makes more sense. Some words also have more than one meaning. For example, 打 can be used in different ways with different meanings. It can be used in the contexts of playing a sport, hitting a person or object, or playing a game.

Second, the Chinese language has many OOV words (out of vocabulary). These are new phrases that are not in dictionaries. These phrases typically come out of cultural references, current hot topics, acronyms, abbreviations, names/nicknames, or just plain slang words. Specifically to the posts that we looked into, out of vocabulary words typically occurred in the case of cultural references, where there are nicknames created for some hot topic issue/person/event of the week just popular slang or informal words and phrases used to convey an emotion. Especially since social media posts are more popular with the young adult generation, many of the words used in the posts are popular informal phrases that would not normally be found in dictionaries. Knowledge of these types of OOV words comes out of knowing the current trends in Asian countries and being up to date with the typical language used by the younger generation.

Third, the Chinese language has two forms - traditional and simplified and it is not a 1-1 correspondence between the two languages which makes it difficult to convert between the two languages for translation or natural language processing.

2.3.2 Characteristics of Sina Weibo Posts

Our analysis takes not only the characteristics of Chinese language, but also the characteristics of Sina Weibo posts into consideration.

The writing of Weibo posts are generally informal. The users may not use standard punctuation marks for separation of sentences and parts of sentences. The most important features are described as follows:

Reposting A user may repost other post. Reposting does not automatically imply agreement or liking. This type of post usually consists of two parts: the reposting user's comment and the post being reposted. The user's comment may be empty or set as default text "Repost" or "转发微博" ("Repost Weibo"). The reposted post itself may include multiple reposting. The topic of keeping track of reposting and identifying agreement or disagreement can be a project itself. In our analysis, only the reposting user's comment is kept.

Spams There are a fair amount of spams on Weibo. Some spam posts are identical except for the URL. Hence, URLs are removed and then we check for duplication in the pre-tagging processing step.

Mentioning A user may mention other users whose usernames are preceded by the **@** symbol. The mentioned usernames may be an integrated part of the post. We define a set of topic-related usernames and substitute the mentioning of these usernames by the corresponding proper nouns. The other mentioned usernames are removed.

Emotion Symbols and Internet Slangs Sina Weibo provides the users a set of emotion symbols, which are corresponding words surrounded by square brackets in text. The users may use other emotion symbols, such as `:)` for smile and `T_T` for crying. Internet slangs are a large part of Out-of-Vocabulary (OOV). Some substitute the characters in a word with the characters which have similar pronunciation, such as "蜀黍" (Shu3 Shu2) for "叔叔" (Shu1 Shu1, means "uncle"). Some are Internet popular interjections, such as "喵了个咪" (喵: "meow," 了: past tense marker, 个: universal measure word, 咪: "mew") which means "dog my cats."

Topic Topic words are surrounded by the pound signs `#` since Chinese language has no explicit delimiter between words. The topic word can be an integrated part of the post.

2.3.3 Pre-tagging Processing

The data set `Han.txt` contains 22,398 posts.

In order to obtain labeled messages for training and testing purpose, the authors manually provided sentiment tags to a data set containing 3000 messages. The three types of sentiment tags used here are positive, negative, and noninformative.

The data need to be cleaned before manual labeling. A typical post looks like:

```
1165303315 2012-04-16 09:55:40 《韩寒收到网友死亡威胁》（来自 @ 新浪娱乐）http://t.cn/zOprKap
```

1. The user identification number and time stamp are removed.
2. Only the reposting user's comment is kept. The reposted part is removed from further analysis. If the resulting string is empty, it will be eliminated as well.
3. URLs are removed.
4. Duplicates are removed.

The output file `hanhanweibo.txt` consists of 13,070 unique posts. 3000 posts are chosen from this data set and will be manually tagged.

2.3.4 Pre-segmentation Processing

As discussed in Section 2.3.1, sentences in Chinese are normally strings of Chinese characters without spaces between words. Hence, word segmentation is crucial for our word-based analysis. According to the characteristics of Weibo posts described in Setion 2.3.2, the following processing is preformed:

1. A set of topic-related usernames are defined. Then the mentioning of these usernames are substituted by the corresponding proper nouns. The other mentioned usernames are removed.
2. A set of emotional symbols and Internet slangs are defined. Then they are substituted by the corresponding word surrounded by square brackets.

2.3.5 Segmentation

汉语词法分析系统 ICTCLAS (Institute of Computing Technology, Chinese Lexical Analysis System) is a well known Chinese word segmentation system developed by Institute of Computing Technology, Chinese Academy of Sciences [3]. It offers the functionality of Chinese word segmentation, lexical tagging, named entity recognition, unknown words detection, and the user-defined dictionary. The current version is ICTCLAS 2011, which supports GB2312, GBK, UTF8 and several encodings and has precision rate of 98.45%.

The Java version of ICTCLAS 2011 preforms word segmentation and lexical tagging on a Linux 32-bit machine. A user-defined dictionary is provided. The entries in this dictionary contains proper nouns and some common Internet slangs. For instance, 微博 (Weibo, wei1 bo2) can be written as 围脖 (wei2 bo2, means "scarf"). Some users refer Han Han as 韩少 (韩: Han Han's surname, 少: abbreviation of 少爷, which means "young master of the house").

Even with the user-defined dictionary, some appearance of Han Han's name 韩寒 can not be segmented and tagged correctly. This is corrected directly using regular expression.

2.3.6 Conjunction Rules

Lee and Renganathan [4] presented that special consideration should be given to the sentences whose parts are linked by contrasting transitional expressions. In particular, if a sentence contains conjunctions such as “although” and “but,” only the part being emphasized will be kept and used to infer the sentiment polarity of this sentence. There are three cases:

1. Although (part A), (part B).
2. (Part A), but (part B).
3. Although (part A), but (part B).

For each case, only part B will be kept.

The four words for “although” are 虽然, 虽说, 虽, and 尽管. The words for “but” are 但, 但是, 不过, 可是, 然而, 只是, 可, 只, 然, and 却.

2.3.7 Stop Words and Punctuation Elimination

Moreover, stop words, non-text strings, and punctuation marks are eliminated. The detailed process is as follows:

1. Remove prepositions, punctuation marks, English character strings, interjections, modal particles, onomatopoeia, and auxiliary words.
2. Remove pre-defined stop words and number strings.

Note that the pre-defined stop words do not contain the following six negation words: 不, 不是, 没有, 没, 无, and 别. These negation words will be used in sentiment score assignment.

2.3.8 Sentiment Score Assignment

Dictionary-based sentiment score can provide us some intuitive understanding of the sentiment polarity of the posts. The dictionaries are obtained from HowNet [2]. HowNet is an online extralinguistic common-sense knowledge system for the computation of meaning in human language technology.

Each post is examined and the numbers of positive and negative words are recorded. Positive word contributes $+1$ to the sentiment score, whereas negative word contributes -1. If there is a negation words among the three words before the positive/negative word, their combination will be treated as an entity and their updated contribution is -1 times the original contribution. The six negation words used are 不, 不是, 没有, 没, 无, and 别. The sentiment score of the post is the sum of all the contributions.

Also topic-related positive/negative words are added to the dictionaries. For instance, the users who refer Han Han as 韩少 (韩: Han Han’s surname, 少: abbreviation of 少爷, which means “young master of the house”) clearly have positive feelings about him.

Another interesting quantities are the numbers of positive and negative words in a neighborhood of a particular person. The neighborhood used in our analysis is three words before and after the person’s name.

2.4 Feature analysis

It is of interest to study the relation between features, i.e. in our case, the relation between words in tweets. We shall base our study on the frequency matrix $X \in R^{n \times p}$, where the entry of the matrix x_{ij} stores the times of occurence of the j-th word in the i-th post. Analyzing the feature matrix X can help us understand the word usage better and identify possible cluster in feature space.

Our observation matrix is the frequency matrix of dimension 756x3000. The columns represent each post and the rows represent 756 phrases. These 756 phrases have total occurance of at least 10 times over all 3000 posts and do not include stop words and negation words. Each element of the observation matrix represents the total number of times that particular word (indicated by row) occurrs in that post (indicated by column). Using 10 fold cross validation, we searched for the optimal regularization parameter.

plot the frequency matrix here.

2.4.1 Exploratory data analysis

converting frequency matrix to cooccurence matrix.

include plots of the co matrix, both matrixplot and the network plot

2.4.2 Sparse principal component analysis (SPCA)

Following Zou et al (2006):

$$(A, B) = \arg\min_{A,B} \left\{ \sum_{i=1}^{n} ||x_i - AB^T x_i||_2^2 + \lambda \sum_{j=1}^{k} ||\beta_j||^2 + \sum_{j=1}^{k} \lambda_{1,j} ||\beta_j||_1 \right\} \tag{1}$$

subject to $A^T A = I_k$.

table here. Some explanations here as well.

2.4.3 Sparse gaussian graphical model

Some review on a Gaussian graphical model. maximum likelihood estimate of the inverse covariance matrix. Suppose the random vector $x \in R^p$ has a multivariate normal distribution with mean μ and covariance matrix Σ. The density function of x is

$$f(x|\mu, \Sigma) = \frac{1}{\sqrt{2\pi \det(\Sigma)}} \exp\left\{ -\frac{1}{2}(x-\mu)^T \Sigma^{-1} (x-\mu) \right\}. \tag{2}$$

Suppose we have IID $N(\nu, \Sigma)$ data $x_1, x_2, ...x_n$ and we want to estimate the inverse covariance matrix Σ^{-1}. The joint likelihood of the data is

$$f(x_1, \ldots, x_n|\mu, \Sigma) = \frac{1}{(2\pi \det(\Sigma))^{n/2}} \exp\left\{ -\frac{1}{2} \sum_{i=1}^{n} (x_i-\mu)^T \Sigma^{-1} (x_i-\mu) \right\}. \tag{3}$$

Taking logarithm to get the log-likelihood (and ignore constant terms):

$$l(\mu, \Sigma^{-1}) = \frac{1}{2 \log \det(\Sigma)} - \frac{1}{2} \sum_{i=1}^{n} (x_i-\mu)^T \Sigma^{-1} (x_i-\mu). \tag{4}$$

Then we can do a maximum likelihood estimation (optimize over μ and $S = \Sigma^{-1}$; easy to see that the MLE for μ is $\bar{X}$ exponential family, MOM = MLE, so just plug in $\bar{X}$ for μ):

$$\max_S \left\{ \frac{1}{2} \log\det(S) - \frac{1}{2} \sum_{i=1}^{n} (x_i - \mu)^T S (x_i - \mu) \right\}. \tag{5}$$

Here comes the trace trick $\sum_{i=1}^{n}(x_i - \mu)^T S(x_i - \mu) = \mathbf{Tr}(\sum_{i=1}^{n}(x_i - \mu)(x_i - \mu)^T S) = \mathbf{Tr}(\hat{\Sigma} S)$. We end up with the following optimization problem

$$\max_S \left\{ \log\det(S) - \mathbf{Tr}\left(\hat{\Sigma} S\right) \right\} \tag{6}$$

if the number of features p is large, want to do model selection... So Banerjee et al. (2007) propose the following optimization problem to recover the sparse structure in a gaussian graphical model

$$\max_S \left\{ \log\det S - \mathbf{Tr}\left(\hat{\Sigma} S\right) - \lambda ||S||_1 \right\} \tag{7}$$

where Σ is the covariance matrix of the data/design matrix X and $||S||_1 = \sum_{i=1}\sum_{j=1}|s_{ij}|$. In that paper they study the senate voting data to recover the party membership. l-1 norm promotes sparsity.

Some computation aspects of estimating a sparse graphical model: Banerjee et al (2007) propose a block coordinate ascent method (COVSEL) (updating one row and one column of S at one time). Their approach is exact but takes forever to run. Meinshausen and Buhlmann (2006) uses an approximation approach that is substantially faster. In our study, we adopt the fast and accurate graphical LASSO procedure (R package glasso) due to Friedman et al (2007).

2.5 Classification

We are also interested in classifying the posts into different categories. Let $x_i \in R^p$ be the i-th row of the frequency matrix X and y_i the corresponding category. For simplicity, let us assume that $y_i \in \{-1, +1\}$ is binary, where the "+1" can be used to label the following four categories: 1) positive opinion towards Han Han; 2) negative opinion towards Han Han; 3) netural or unidentifiable opinion; 4) spam and "-1" labels the complement of the inividual category (e.g. if "+1" means positive, "-1" would mean anything but positive. Note that the complement of positive is not negative, but rather the union of negative, neutral and spam). For each of the above four cases, we apply LASSO and l_1-norm support vector machine to classify the data points into "-1" and "+1".

For each of the four models, we changed all responses that were not the category we were modeling to -1 and responses that fell into the category we were looking into as 1.

2.5.1 Sparse regression with the LASSO

The LASSO (Tibshirani, 1996) is a popular sparse regression method which adds a l_1-norm penalty to the linear least squares problem to promote sparsity in the regression coefficients:

$$\hat{\beta}(\lambda) = \arg\min_\beta \frac{1}{2} ||y - (\beta_0 + X\beta)||_2^2 + \lambda ||\beta||_1 \tag{8}$$

In our study, we regress the class label vector y onto the word frequency matrix X, yielding the intercept $\hat{\beta}_0$ and the sparse regression vector $\hat{\beta}(\lambda)$. We can then define the classifier to be $f(x) = \mathbf{sign}(\hat{\beta}_0 + X\hat{\beta}(\lambda)) \in \{-1, +1\}$, where the resulting coefficient regression coefficient $\hat{\beta}$ has the following explanation: for each feature/word j, given all other feature/word variable fixed, the increase of the j-th word frequency by one lead to increase in regression function $\beta_0 + X\beta$ by an amount of β_i (if β_i turns out to be positive, this means the chance of classifying the data point into the $+1$ category is increased).

To look at which words were most relevant to each category we modeled (positive, negative, neutral and spam), we looked at three sets of 20 words based on, respectively, the highest absolute beta values, most positive coefficient values and most negative coefficient values. The top 30 words based on the top 50 highest absolute beta values tells us in general which words were most relevant to predicting that particular category. The top 50 words based on the top 50 highest positive beta values tells us which words typically were common in posts that fell into the category observed while the top 50 words based on the top 50 highest negative beta values tells us which words typically were not in posts that fell into the category observed but were rather more common in the categories outside of the one we were modeling.

Since the estimated β depend on the regularization parameter, we are left with the issue of choose the “best” λ. A commonly used approach is to do a grid search for λ: for each value of λ, do a 10-fold cross validation; then choose the λ that yields the smallest cross validation testint sample error. For this purpose, we used the approach least angle regression by Efron and Hastie (2007) to do the model selection. Their R package lars efficiently fits an entire lasso sequence with the least squares loss function.

include the one of the CV plot here; and include the others in the appendix.

For the positive responses, the top 20 most relevant words (based on taking the absolute values of the betas) returned by our model are shown in the table. As can be seen, the top 20 relevant words have an assortment of fairly neutral or positive words. The top 20 most relevant words for just the positive betas resulted in words that were very positive in nature. For example, words such as ’mature’, ’support’, and ’keep going’ are very positive in nature and would certainly indicate a positive reaction to 韩寒. The top 20 most relevant words for just the negative betas resulted in words that showed a great dislike for 韩寒. Words such as ’liar’ indicate a negative response to the author.

For the negative responses, the top 20 most relevant words (based on taking the absolute values of the betas) returned by our model are shown in the table. As can be seen, the words are typically negative or neutral. The top 20 most relevant words for just the positive betas are very telling in the posts sentiment towards 韩寒. Words such as ’disgusting, ’hate’, ’liar’ and ’annoying’ demonstrates easily that the post has a negative sentiment towards the author. The top 20 most relevant words for just the negative betas are hence typically more positive towards the author. With words such as ’support’, ’good’, and ’like’, the posts would not have a negative sentiment. What is interesting to note with these betas is the fact that these betas are all close to zero except the highest phrase.

For the neutral responses, the top 20 most relevant words (based on taking the absolute values of the betas) as well as the top 20 most relevant words based on the positive betas are all neutral words. The top 20 most relevant words based on the negative betas consist mainly of phrases that have some clear emotion attached to them, such as ”support”, ”hate”, ”agree”, and ”ghostwriter”. For the spam posts, the top 20 most relevant words (based on taking the absolute values of the betas) as well as the top most relevant words based on the positive betas are words/phrases that have no relation whatsoever with the author we picked to look into. As expected, the top 20 most relevant words based on the negative betas are words/phrases that do have to do with the topic, such as the author’s name.

Table 1: Positive category

Word	Absolute Coef.	Word	Positive Coef.	Word	Negative Coef.
加油 (keep going)	0.820	加油 (keep going)	0.820	样子 (manner)	-0.396
韩少 (Master Han)	0.644	韩少 (Master Han)	0.644	恋 (love)	-0.344
成熟 (mature)	0.546	成熟 (mature)	0.546	发表 (announce)	-0.336
顶 (support)	0.533	顶 (support)	0.533	道理 (rational)	-0.336
宽容 (tolerant)	0.518	宽容 (tolerant)	0.518	利益 (benefit)	-0.335

Table 2: Negative category

Word	Absolute Coef.	Word	Positive Coef.	Word	Negative Coef.
讨厌 (hate)	0.481	讨厌 (hate)	0.481	支持 (support)	-0.008
无耻 (shameless)	0.412	无耻 (shameless)	0.412	不 (no)	0.000
恶心 (disgusting)	0.395	恶心 (disgusting)	0.395	人 (people/person)	0.000
骗子 (liar)	0.380	骗子 (liar)	0.380	说 (say)	0.000
扁 (beat up)	0.353	扁 (beat up)	0.353	方舟子 (FangZhouZi)	0.000

for the spam responses... change 20 to 5, and say the complete lists of words can be found in the appendix

2.5.2 l_1-norm support vector machine

Support vector machine (Vapnik 1996) is another commonly-used machine learning method to classify data points into two categories. Consider again the linear decision function $f(x) = \beta_0 + \beta x$ and the classifier $Class(x) = \textbf{sign}(f(x))$. To "learn" the parameters, we want the training misclassification rate to be small and the margin of the decision boundary (which can be shown to be $1/||\beta||_2$) to be wide. Hence we consider the following optimization problem (give a 2d illustration here...):

$$\min_{\beta_0,\beta} \sum_{i=1}^{n} (1 - y_i(\beta_0 + \beta^T x_i))_+ + \frac{\lambda}{2}||\beta||_2, \tag{9}$$

Table 3: Neutral category

Word	Absolute Coef.	Word	Positive Coef.	Word	Negative Coef.
上调 (increase)	0.586	上调 (increase)	0.586	加油 (keep going)	-0.491
道理 (rational)	0.566	道理 (rational)	0.566	韩少 (Master Han)	-0.358
账号 (account)	0.534	账号 (account)	0.534	苦肉计 (the ruse of self-injury to win somebody's confidence)	-0.327
加油 (keep going)	0.491	铁证 (clear evidence)	0.459	支持 (support)	-0.290
铁证 (clear evidence)	0.459	称 (refer)	0.453	善良 (kind)	-0.268

Table 4: Spam category

Word	Absolute Coef.	Word	Positive Coef.	Word	Negative Coef.
查看 (examine)	3.777	查看 (examine)	3.777	韩少 (Master Hanhan)	-1.033
抽 (win)	1.998	抽 (win)	1.998	韩寒 (Master Hanhan)	-0.716
每天 (everyday)	1.251	每天 (everyday)	1.251	别 (don't)	-0.232
往往 (often)	1.208	往往 (often)	1.208	支持 (support)	-0.217
外 (outside)	1.043	外 (outside)	1.043	这种 (this kind)	-0.202

where $z_+ = \max(0, z)$ (the function $h(z) = (1 - z)_+$ is also known as the hinge loss function). Similarly, the sparse version of SVM simply replaces the l_2-norm by l_1-norm (which is just another measure of the wideness of the margin):

$$\min_{\beta_0,\beta} \left\{ \sum_{i=1}^{n} (1 - y_i(\beta_0 + \beta^T x_i))_+ + \lambda ||\beta||_1 \right\}. \tag{10}$$

We repeat the same data analysis as we did for the LASSO method. The results are summaried in table XX. For efficiently fitting the sparse svm, we use the matlab package by Fung and Mangasarian (2004). Again, 10-fold cross validations are performed in order to select the "best" λ.

include the four tables here(each of top five words)... and rest in the appendix

3 Discussion

ROC curve precision and recall curve

3.1 Limitations

sampling [1] [5] [6]

reposting

other language, such as English

simplified Chinese and traditional Chinese: no simple one-to-one correspondence; word segmentation and then substitute words

4 Conclusion

References

[1] Boyd, S., Diaconis, P., and Xiao, L. Fastest mixing markov chain on a graph. *SIAM review* (2004), 667–689.

[2] Dong, Z., and Dong, Q. 知网 HowNet. http://www.keenage.com/.

[3] Institute of Computing Technology, Chinese Academy of Sciences. 汉语词法分析系统 ICTCLAS (Institute of Computing Technology, Chinese Lexical Analysis System). http://ictclas.org/, 2011.

[4] Lee, H., and Renganathan, H. Chinese sentiment analysis using maximum entropy. *Sentiment Analysis where AI meets Psychology (SAAIP)* (2011), 89.

[5] Leskovec, J., and Faloutsos, C. Sampling from large graphs. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining* (2006), ACM, pp. 631–636.

[6] Wang, T., Chen, Y., Zhang, Z., Xu, T., Jin, L., Hui, P., Deng, B., and Li, X. Understanding graph sampling algorithms for social network analysis. In *Distributed Computing Systems Workshops (ICDCSW), 2011 31st International Conference on* (2011), IEEE, pp. 123–128.

[7] Wong, K., Li, W., Xu, R., and Zhang, Z. Introduction to chinese natural language processing. *Synthesis Lectures on Human Language Technologies 2*, 1 (2009), 1–148.

Table 5: Positive category

Word	Absolute Coef.	Word	Positive Coef.	Word	Negative Coef.
加油 (keep going)	0.820	加油 (keep going)	0.820	样子 (manner)	-0.396
韩少 (Master Han)	0.644	韩少 (Master Han)	0.644	恋 (love)	-0.344
成熟 (mature)	0.546	成熟 (mature)	0.546	发表 (announce)	-0.336
顶 (support)	0.533	顶 (support)	0.533	道理 (rational)	-0.336
宽容 (tolerant)	0.518	宽容 (tolerant)	0.518	利益 (benefit)	-0.335
支持	0.477	支持	0.477	称	-0.323
家人	0.467	家人	0.467	遭受	-0.323
样子	0.396	尤其	0.395	媒体	-0.319
尤其	0.395	欣赏	0.383	翻	-0.314
欣赏	0.383	感动	0.381	铁证	-0.289
感动	0.381	影响力	0.370	骗子	-0.248
影响力	0.370	新书	0.327	上调	-0.248
恋	0.344	铁	0.316	投票	-0.234
发表	0.336	不错	0.309	女	-0.230
道理	0.336	终于	0.274	四娘	-0.226
利益	0.335	每个	0.274	关系	-0.215
新书	0.327	咬	0.261	广告	-0.210
称	0.323	文字	0.260	接受	-0.208
遭受	0.323	蛋	0.244	网	-0.204
媒体	0.319	纠缠	0.244	底	-0.196

5 Appendix

Table 6: Negative category

Word	Absolute Coef.	Word	Positive Coef.	Word	Negative Coef.
讨厌 (hate)	0.481	讨厌 (hate)	0.481	支持 (support)	-0.008
无耻 (shameless)	0.412	无耻 (shameless)	0.412	不 (no)	0.000
恶心 (disgusting)	0.395	恶心 (disgusting)	0.395	人 (people/person)	0.000
骗子 (liar)	0.380	骗子 (liar)	0.380	说 (say)	0.000
扁 (beat up)	0.353	扁 (beat up)	0.353	方舟子 (FangZhouZi)	0.000
装	0.321	装	0.321	韩少	0.000
选项	0.292	选项	0.292	真	0.000
苦肉计	0.290	苦肉计	0.290	好	0.000
利益	0.283	利益	0.283	没	0.000
全	0.261	全	0.261	一个	0.000
国家	0.247	国家	0.247	微博	0.000
智商	0.216	智商	0.216	写	0.000
告	0.198	告	0.198	喜欢	0.000
虚伪	0.192	虚伪	0.192	想	0.000
演	0.191	演	0.191	威胁	0.000
语	0.186	语	0.186	只	0.000
烦	0.178	烦	0.178	太	0.000
掉	0.142	掉	0.142	事	0.000
下去	0.141	下去	0.141	没有	0.000
公开	0.141	公开	0.141	看到	0.000

Table 7: Neutral category

Word	Absolute Coef.	Word	Positive Coef.	Word	Negative Coef.
上调 (increase)	0.586	上调 (increase)	0.586	加油 (keep going)	-0.491
道理 (rational)	0.566	道理 (rational)	0.566	韩少 (Master Han)	-0.358
账号 (account)	0.534	账号 (account)	0.534	苦肉计 (the ruse of self-injury to win somebody’s confidence)	-0.327
加油 (keep going)	0.491	铁证 (clear evidence)	0.459	支持 (support)	-0.290
铁证 (clear evidence)	0.459	称 (refer)	0.453	善良 (kind)	-0.268
称	0.453	想起	0.353	成熟	-0.263
韩少	0.358	杀	0.331	终于	-0.239
想起	0.353	最终	0.329	家人	-0.233
杀	0.331	意思	0.323	同意	-0.228
最终	0.329	遭遇	0.319	越来越	-0.220
苦肉计	0.327	金	0.308	欢乐	-0.217
意思	0.323	片	0.287	崇拜	-0.213
遭遇	0.319	应	0.278	讨厌	-0.202
金	0.308	变成	0.265	顶	-0.197
支持	0.290	有点	0.262	代笔	-0.194
片	0.287	之间	0.254	跳	-0.191
应	0.278	右边	0.250	真善美	-0.186
善良	0.268	民主	0.238	真正	-0.185
变成	0.265	郭敬明	0.237	欣赏	-0.181
成熟	0.263	久	0.226	无耻	-0.180

Table 8: Spam category

Word	Absolute Coef.	Word	Positive Coef.	Word	Negative Coef.
查看 (examine)	3.777	查看 (examine)	3.777	韩少 (Master Hanhan)	-1.033
抽 (win)	1.998	抽 (win)	1.998	韩寒 (Master Hanhan)	-0.716
每天 (everyday)	1.251	每天 (everyday)	1.251	别 (don't)	-0.232
往往 (often)	1.208	往往 (often)	1.208	支持 (support)	-0.217
外 (outside)	1.043	外 (outside)	1.043	这种 (this kind)	-0.202
韩少	1.033	征集	0.948	感	-0.196
征集	0.948	容	0.849	韓	-0.191
容	0.849	风	0.649	没有	-0.179
韩寒	0.716	票子	0.570	上调	-0.174
风	0.649	考	0.540	方舟子	-0.160
票子	0.570	主	0.438	光明	-0.141
考	0.540	性	0.430	一定	-0.137
主	0.438	总是	0.416	照妖镜	-0.132
性	0.430	儿	0.416	写	-0.132
总是	0.416	结论	0.405	觉得	-0.119
儿	0.416	后面	0.397	甚	-0.110
结论	0.405	法律	0.388	韩	-0.092
后面	0.397	机会	0.376	真相	-0.084
法律	0.388	公知	0.357	挺	-0.072
机会	0.376	中	0.357	不	-0.068