Chen B, He H, Guo J. Constructing maximum entropy language models for movie review subjectivity analysis. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 23(2): 231–239 Mar. 2008

Constructing Maximum Entropy Language Models for Movie Review Subjectivity Analysis

Bo Chen (陈 博), Hui He (何 慧), and Jun Guo (郭 军)

Pattern Recognition and Intelligent System Laboratory, School of Information Engineering
Beijing University of Posts and Telecommunications, Beijing 100876, China

E-mail: {chb615, hh1012}@gmail.com; guojun@bupt.edu.cn

Revised September 23, 2007.

Abstract Document subjectivity analysis has become an important aspect of web text content mining. This problem is similar to traditional text categorization, thus many related classification techniques can be adapted here. However, there is one significant difference that more language or semantic information is required for better estimating the subjectivity of a document. Therefore, in this paper, our focuses are mainly on two aspects. One is how to extract useful and meaningful language features, and the other is how to construct appropriate language models efficiently for this special task. For the first issue, we conduct a Global-Filtering and Local-Weighting strategy to select and evaluate language features in a series of n-grams with different orders and within various distance-windows. For the second issue, we adopt Maximum Entropy (MaxEnt) modeling methods to construct our language model framework. Besides the classical MaxEnt models, we have also constructed two kinds of improved models with Gaussian and exponential priors respectively. Detailed experiments given in this paper show that with well selected and weighted language features, MaxEnt models with exponential priors are significantly more suitable for the text subjectivity analysis task.

Keywords exponential prior, language model, maximum entropy, n-gram, subjectivity analysis

1 Introduction

As the rapid growth of the Internet, billions of web documents are available on-line. People need some better organized information from such a huge web information database. This has motivated the research into techniques of automatic text categorization. In early times, researches have been focused on so-called "topic categorization", i.e., given a document in text format, a computer tells which already known topic (e.g., politics, economics, military) the document should belong to. Recently, as more and more weblog, review and shopping sites appear, people are not only satisfied to know what kinds of things are there, but also eager to know how others think about things (are they good or bad?). How to make such requirements conveniently have brought out a new research field, sentiment analysis. One important aspect of sentiment analysis is document subjectivity analysis, which aims to tell people whether an article about a particular thing is subjective or objective. This technique maybe useful in many areas of applications, for example, stock discussion boards[1], blog sentiment[2], customer feedback[3], product and movie review[4].

Most documents on these websites often tend to be short (just one or two sentences, or even several words), even too "tiny" to be viewed as articles. We would like to view them as short texts or text snippets. These phenomena bring out a first challenge on how to mine efficient language features from such short web texts.

In this paper, the movie review dataset① being studied on is comprised of ten thousand text snippets extracted from websites. To mine as much meaningful language features as possible, under the well-known n-gram model framework, we utilize various n-gram templates to capture language features with different orders and within various distance-windows. Additionally, we then conduct a Global-Filtering and Local-Weighting strategy to select and evaluate these language features.

On the other hand, a main difference between subjectivity classification and traditional top-based categorization is that while topics can often be identified

Regular Paper

Supported by the National Natural Science Foundation of China under Grant Nos. 60475007 and 60675001, the Key Project of Chinese Ministry of Education under Grant No. 02029, and the Foundation of Chinese Ministry of Education for Century Spanning Talent.

①This dataset comes from http://www.cs.cornell.edu/people/pabo/movie-review-data, offered by Pang *et al.* at Cornell.

by some individual literal features, subjectivity is more latent in contexts. Thus, more sophisticated language features may be employed. This problem arises the second challenge on language information fusion, which can be defined as constructing appropriate language models that can efficiently combine all kinds of language features. Fortunately, in recent years, there are a lot of studies on Maximum Entropy (MaxEnt) models, which can combine almost all kinds of information into an individual probabilistic model, especially in natural language processing (NLP) field[5,6]. Thereby, we adopt MaxEnt modeling methods to construct our language model framework. Besides the classical MaxEnt model, we have also tried two improved models with Gaussian and exponential priors respectively to make the models more suitable for the text subjectivity analysis task.

The remainder of this paper is organized as follow. In Section 2, a briefly review of previous work on text categorization and sentiment classification is introduced. How the n-gram based language features are extracted and weighted are described in Section 3. Then, in Section 4, how the Maximum Entropy language models are constructed and how the priors are applied are introduced. Experiments and results are reported in Section 5. At last, conclusions are derived in Section 6.

2 Previous Work

2.1 Topic-Based Text Categorization

The subjectivity analysis problem can be traced back into traditional topic-based text categorization, which is to classify a document into a pre-defined category by computers automatically. In such cases, documents are usually represented as vectors of individual text features that can be readable by computers. Based on such representations, various machine learning approaches, such as probabilistic algorithms, neural networks, regression models, nearest neighbor classifiers, Bayesian probabilistic classifiers, decision trees, inductive rule learning algorithms, profile-construction methods, sample-based classifiers, support vector machines, and classifier committees[7,8], can be applied to construct some classifiers on corresponding training datasets.

The problem of subjectivity analysis can also be viewed as a special categorization task, where categories are "subjective" and "objective" labels.

2.2 Sentiment Classification

Pang *et al.* reported their early work on document polarity classification in [9] and their improved work in [10]. In their early experiments, they used 700 positive and 700 negative reviews and evaluated their classifiers by several 3-fold cross validations. They have used unigrams (with term frequency and Boolean value as feature weight respectively), bigrams, unigrams + bigrams, unigrams + POS (Part of Speech), top unigrams (top 2633), as well as unigrams + position as their language features respectively. From their sentiment classification accuracies reported in [9], we have found some unexpected but interesting results.

1) The best result was obtained while using unigram features with Boolean values (present or absence) other than unigram with term frequency values. This is somewhat different from those cases in topic categorization task, in which term frequency is always a kind of most useful feature attributes.

2) Using bigram + unigram features is worse than merely using unigrams. This is against our intuitions, for there are so many phrases or component words consisting of two or more words expressing significant emotion tendencies, while do not their component words. For instance, "how could..." is often of strong emphasis tone, while "how" and "could" are mostly used as neuter words.

3) Using the top 2633 unigram features even does no worse than using those all 32 330 unigram + bigram features, but is slightly worse than using all 16 165 unigram features. How to extract a small amount of features carrying the most useful information will be a practical problem.

Corresponding to these issues, we have carried out some studies on subjectivity classification as reported in [11]. In our previous work, we have tried three feature weighting methods, Boolean, absolute-term-frequency, and normalized-term-frequency. Models based on the former two weighting methods performed similarly, while models using the normalized frequencies obviously outperformed the former two kinds of models. In addition, we have also investigated models with bigram features added in, and the combining of high-order language features within limited distance-windows leads to obvious improvement on basic unigram feature based models. Finally, the best result was achieved by a model using normalized unigram and bigram features within a 2-distance context window. This model will be treated as the baseline in our current work.

3 Language Feature Extraction

Document representation plays an important role in traditional topic-based categorization. In normal documents, there are always enough items to form term-

vectors, while in short text snippets, fewer items are there. Thus, we utilize n-gram language templates to capture more sophisticated language features.

3.1 Language Features

Definition 3.1. *A sentence T consisting of N individual sequential words appears as $\{w_1, w_2, \ldots, w_N\}$, then*

• *Conventional n-gram*

Using a conventional n-order-gram ($n \geqslant 1$) template, e.g., in bigram ($n = 2$), sentence T will have a feature set $\{w_1w_2, w_2w_3, \ldots, w_{N-1}w_N\}$.

• *Long distance n-gram*

Using an n-order-gram template with d-distance back, for instance, in a 3-distance bigram model, the feature set will be $\{w_1w_4, w_2w_5, \ldots, w_{N-3}w_N\}$. While the distance is set to 1, it becomes a conventional n-gram template.

• *N-grams within long distance-window*

N-grams within d-distance-window mean that n-order-gram templates with distances no larger than d are all combined in, e.g., bigram feature set within a 3-distance-window would be:

$$\{w_1w_2, w_1w_3, w_1w_4, \ldots, w_{N-3}w_N, w_{N-2}w_N, w_{N-1}w_N\}.$$

Huang *et al.* estimated a series of d-distance bigram models for $d = 1, \ldots, 1000$[12]. They concluded that the history of the last 5 words contains most of the significant information. As a reference, we tested a series of long distance-window based bigram features with the maximum distance set to 5.

3.2 Feature Selection and Weighting

By using various n-gram templates, the original amount of features will be too huge for an efficient document analysis system. On the other hand, a great percent of these language-grams are nothing but sequential words, which have neither language structure nor semantic meaning. We will view them as spam items, which must be discarded from our language models. Additionally, different language-grams are of different importance regarding to language functions and semantic meanings, thus they could not be viewed identically. Therefore, these original language features must be filtered and evaluated. In other words, there are two procedures need to be perform, one is feature selection, and the other is feature weighting.

In the MaxEnt framework (described in the next section) used in our work, these two aspects are combined together to a certain extent, for that whether a feature is necessary and how important it is are all reflected on its parameter assigned by the MaxEnt model. for example, if a feature is redundant, then theoretically, it will get a zero weight in the MaxEnt model, while a significant feature receives a relatively high score.

However, we still think that the well selected and weighted features may make MaxEnt models more efficient. Thus, we applied some different feature selection and weighting methods as described below.

In our previous work[11], the three feature weighting methods reported are all directly based on the local term frequency within an individual text snippet. Therefore, the corresponding feature selection strategies are simply as a feature elimination procedure that is to discard those features appearing fewer times than a threshold. Experimental results show that models using normalized-term-frequency features work better compared to those using Boolean value features.

Besides these local feature based methods, we intend to utilize some global feature information to improve feature extraction performance. In text categorization and text retrieval, the TFIDF based method is the most popular and basic text feature weighting method[7,13]. A standard definition form of TFIDF is:

$$tfidf(t_k, d_i) = tf(t_k, d_i) \times \log\Big(\frac{|D|}{df(t_k)}\Big).$$

Wherein, t_k stands for a feature term, d_i indicates the i-th document in the whole document set D, $tf(t_k, d_i)$ is the Term-Frequency (TF) defined as the number of times that term t_k occurs in document d_i, while $df(t_k)$ is the Document-Frequency (DF) denoting how many documents in D does term t_k appear in, and $|D|$ is the total number of documents in set D.

In our work, we used a modified TFIDF form as:

$$tfidf(t_k, d_i) = \sqrt{tf(t_k, d_i)} \times \log\Big(\frac{|D|}{df(t_k)}\Big). \quad (1)$$

In this formula, the square root of $tf(t_k, d_i)$ is adopted instead of the original term frequency. This makes a smoothing to terms with high frequencies, and leads to a more balanced feature space consisting of different n-grams, for that unigram terms obviously have much more occurrence opportunities than high-order grams.

In addition, as it has been done in many previous work[7], to make term weights independent of the sizes of document, a cosine-normalization has been applied to make new weight values in [0, 1] interval, formulated

as

$$w(t_k, d_i) = \frac{tfidf(t_k, d_i)}{\sqrt{\sum_{t_k \text{ occurs in } d_j} (tfidf(t_k, d_i))^2}}. \quad (2)$$

Based on the TFIDF weights of each feature, we conduct a Global-Filtering and Local-Weighting strategy to implement feature filtering and weighting.

• *Global-Filtering*

After having got the original statistics on TF and DF of each term, we view that all texts in set D compose a Big-Document, and compute a global significant indicator $W(t_k, D)$ for each term t_k using (1) and (2), with the factor $tf(t_k, d_i)$ in (2) being substituted by a global TF factor defined as $TF(t_k, D) = \sum_i tf(t_k, d_i)$. Having $W(t_k, D)$ for each term t_k, we can apply feature selection by setting a threshold on global weight value, as reported in our experiments later.

• *Local-Weighting*

To generate the features of each document d_i (individual text snippet here), we first get all the possible language-grams according to the n-gram templates being adopted. These candidates are filtered by using the Global-Filtering strategy just mentioned. After that, those qualified features are weighted by the local value $w(t_k, d_i)$ defined in (2).

4 Language Model Construction

In our subjectivity analysis task, language features are with different orders and various distances. We adopt the MaxEnt method to construct our language model framework.

4.1 Maximum Entropy Model

MaxEnt models are very strong tools handy for statistical estimation and pattern recognition related fields, as introduced in [6, 14]. These models are a kind of the exponential models with some interesting mathematical and philosophical properties. Given an observation sample set X and label set Y, for each sample $x \in X$, its probability of being assigned label $y \in Y$ estimated by a MaxEnt model is:

$$p_\Lambda(y|x) = \frac{1}{Z(x)} \exp\Big(\sum_i \lambda_i f_i(y, x)\Big). \quad (3)$$

Wherein, $Z(x) = \sum_{y \in Y} \exp\Big(\sum_i \lambda_i f_i(y, x)\Big)$ is a normalization factor, which makes $p(y|x, \lambda)$ a conditional probability. The factor $f_i(y, x)$ is a Boolean indication function representing a feature based on (y, x). For example, in the subjectivity classification task, a significant feature function $f_i(y, x)$ is

$$f_i(y, x) = \begin{cases} 1, & y = \text{“subjective”},\ x = \text{“}\ldots, but \ldots\text{”} \\ & \text{(disjunctiv\_clause\_structure);} \\ 0, & \text{else.} \end{cases}$$

$\boldsymbol{\Lambda} = \{\lambda_i\}$ is the parameter vector, and λ_i is feature f_i's weight parameter indicating the importance of f_i in a MaxEnt model.

As mentioned in the last section, the weights of language features we used are positive real numbers, not Boolean values as described in the expression above. Fortunately, in [14], which is a cornerstone of the MaxEnt theory, Della Pietra *et al.* demonstrated that a MaxEnt model would be constructed applicably whenever the feature functions f_i are nonnegative. This characteristic allows us to combine the TFIDF-based feature extraction methods into MaxEnt modeling by simply replacing the Boolean value with a real value.

4.2 Maximum Entropy Modeling

From (3), it can be noticed that as long as the feature set is fixed, a MaxEnt model is entirely determined by the parameter vector $\boldsymbol{\Lambda} = \{\lambda_i\}$. Thus, to construct a MaxEnt model is to estimate the parameter vector $\boldsymbol{\Lambda} = \{\lambda_i\}$, so that the MaxEnt model fits the dataset the best.

In [14], it is introduced that one most important property of MaxEnt models is the constraint satisfaction on observed data. For each indication function $f_i(x, y)$, let $\tilde{p}(x, y)$ be the empirical distribution of sample (y, x), then the empirical expectation of $f_i(x, y)$ is defined as:

$$\tilde{E}(f_i) = \sum_{x \in X, y \in Y} \tilde{p}(x, y) f_i(x, y).$$

the expected value of f_i estimated by the MaxEnt model as:

$$E(f_i) = \sum_{x \in X, y \in Y} \tilde{p}(x) p_{\boldsymbol{\Lambda}}(y|x) f_i(x, y).$$

Wherein, $\tilde{p}(x)$ is the empirical distribution of sample x, and $p_{\boldsymbol{\Lambda}}(y|x)$ is the conditional distribution of (y, x) estimated by the MaxEnt model.

MaxEnt models require constrains on f_i that

$$E(f_i) = \tilde{E}(f_i), \quad \text{for all } f_i. \quad (4)$$

On the other hand, another important constrain is that MaxEnt models try to remain as similar to the uniform distribution as possible. With these constrains, a

MaxEnt model is trained to find the most suitable parameter vector $\boldsymbol{\Lambda} = \{\lambda_i\}$ that maximizes the likelihood over the dataset, and the optimization object function will be:

$$\begin{aligned}\arg\max_{\boldsymbol{\Lambda}} \Pr(\tilde{p}|p) &= \arg\max_{\boldsymbol{\Lambda}} \prod_{x\in X, y\in Y} p_{\boldsymbol{\Lambda}}(y|x)^{\tilde{p}(x,y)} \\ &= \arg\max_{\boldsymbol{\Lambda}} \sum_{x\in X, y\in Y} \tilde{p}(x,y)\log(p_{\boldsymbol{\Lambda}}(y|x)).\end{aligned} \tag{5}$$

4.3 Priors on Maximum Entropy Models

Like most models trained under the maximum likelihood principle, MaxEnt models also involve in the overfitting problem[15]. Firstly, this is obviously caused by constrains expressed in (4). This can be solved to some extent by applying some smoothing techniques that are widely used in language modeling areas[16]. Furthermore, from the maximum likelihood estimation principle, it can be noticed that "Maximum likelihood estimation is just a degenerate form of Bayesian modeling where the prior over models p (here refers to $p_{\boldsymbol{\Lambda}}(y|x)$) is uniform" (Berger[17]). This could be illustrated as follows.

Finding the model that matches the empirical distribution the best can be formulated as solving $\arg\max_{\boldsymbol{\Lambda}} \Pr(p_{\boldsymbol{\Lambda}}|\tilde{p})$. By using Bayes' law, we get

$$\begin{aligned}\arg\max_{\boldsymbol{\Lambda}} \Pr(p_{\boldsymbol{\Lambda}}|\tilde{p}) &= \arg\max_{\boldsymbol{\Lambda}} \frac{\Pr(\tilde{p}|p_{\boldsymbol{\Lambda}})\Pr(p_{\boldsymbol{\Lambda}})}{\Pr(\tilde{p})} \\ &= \arg\max_{\boldsymbol{\Lambda}} \Pr(\tilde{p}|p_{\boldsymbol{\Lambda}})\Pr(\tilde{p}|p_{\boldsymbol{\Lambda}}).\end{aligned}$$

Assuming $\Pr(p)$ is uniform, then

$$\arg\max_{\boldsymbol{\Lambda}} \Pr(p_{\boldsymbol{\Lambda}}|\tilde{p}) = \arg\max_{\boldsymbol{\Lambda}} \Pr(\tilde{p}|p_{\boldsymbol{\Lambda}}).$$

From the above two formulations, it can be seen that in the maximum likelihood estimation method, the prior $\Pr(p_{\boldsymbol{\Lambda}})$ of each possible model $p_{\boldsymbol{\Lambda}}(y|x)$ is lost.

Recently, many researchers have paid great attentions to the priors on MaxEnt models. Chen and Rosenfeld[18] have implemented a Gaussian prior with 0 mean for MaxEnt model on language modeling task, and concluded that it was consistently the best compared to previous n-gram smoothing methods. With a Gaussian prior on the parameter λ_i as $p(\lambda_i) = \frac{1}{\sqrt{2\pi\sigma_i^2}} \exp\big(-\frac{\lambda_i^2}{2\sigma_i^2}\big)$, the object function (5) becomes

$$\begin{aligned}&\arg\max_{\boldsymbol{\Lambda}} \Pr(\tilde{p}|p)\Pr(p) \\ &= \arg\max_{\boldsymbol{\Lambda}} \prod_{x\in X, y\in Y} p_{\boldsymbol{\Lambda}}(y|x)^{\tilde{p}(x,y)} \\ &\quad \times \prod_i \frac{1}{\sqrt{2\pi\sigma_i^2}} \exp\Big(-\frac{\lambda_i^2}{2\sigma_i^2}\Big) \\ &= \arg\max_{\boldsymbol{\Lambda}} \sum_{x\in X, y\in Y} \tilde{p}(x,y)\log(p_{\boldsymbol{\Lambda}}(y|x)) - \sum_i \frac{\lambda_i^2}{2\sigma_i^2}.\end{aligned}$$

Such a model is aimed at maximize posteriori instead of maximum likelihood on parameter values. Besides, the Gaussian priors will also lead to changes on the feature function constrains (4) as

$$E(f_i) = \widetilde{E}(f_i) - \frac{\lambda_i}{\sigma_i^2}, \quad \text{for all} \quad f_i.$$

From this expression, it can be found that the Gaussian priors in fact add some discounts on the original constrain of f_i, which result in a typical smoothing approach.

Kazama *et al.*[19] explored a MaxEnt model with box-type inequality constraints to relax the equality constrains defined by (4) as

$$A_i \geqslant \tilde{E}(f_i) - E(f_i) \geqslant -B_i, \quad A_i, B_i > 0, \quad \text{for all } f_i.$$

This optimization with inequality constraints results in sparse solution, so that features with a zero parameter can be removed from the MaxEnt model. They concluded that the inequality MaxEnt model embedded feature selection in its estimation and the sparseness of the solution improved the robustness of the MaxEnt model as well.

In fact, the inequality MaxEnt model is just like a MaxEnt model with Laplace priors, which is similar to models with exponential priors (i.e., single-side Laplace priors), as introduced by Goodman in [20]. Assuming an exponential prior on parameter λ_i as $p(\lambda_i) = \mu_i \exp(-\mu_i\lambda_i)$, the object function of a MaxEnt model becomes

$$\begin{aligned}&\arg\max_{\boldsymbol{\Lambda}} \Pr(\tilde{p}|p)\Pr(p) \\ &= \arg\max_{\boldsymbol{\Lambda}} \prod_{x\in X, y\in Y} p_{\boldsymbol{\Lambda}}(y|x)^{\tilde{p}(x,y)} \\ &\quad \times \prod_i \mu_i \exp(-\mu_i\lambda_i) \\ &= \arg\max_{\boldsymbol{\Lambda}} \sum_{x\in X, y\in Y} \tilde{p}(x,y)\log(p_{\boldsymbol{\Lambda}}(y|x)) - \sum_i \mu_i\lambda_i.\end{aligned}$$

The priors also make changes on the constrains as

$$\begin{cases} E(f_i) = \tilde{E}(f_i) - \mu_i, & \lambda_i > 0 \\ E(f_i) > \tilde{E}(f_i) - \mu_i, & \lambda_i = 0 \end{cases}, \text{for all } f_i.$$

Such constrains lead to bounded absolute discounting

by constants. Goodman showed that exponential priors could also lead to a simpler learning algorithm.

In our view, which prior is better should be determined by data. From the conditional MaxEnt model defined in (3), it can be found that the prior probabilistic distribution $\Pr(p_{\boldsymbol{\Lambda}})$ of a model $p_{\boldsymbol{\Lambda}}$ is the function of parameter vector $\boldsymbol{\Lambda} = \{\lambda_i\}$, i.e., $\Pr(p_{\boldsymbol{\Lambda}}) = \Pr(\boldsymbol{\Lambda})$. While each λ_i corresponds to a unique feature function f_i, we may assume that the distribution of model $p_{\boldsymbol{\Lambda}}$ is similar to the distribution of feature f_is over the dataset. Taking this viewpoint, we have tried MaxEnt models with the two kinds of priors compared to the normal MaxEnt model, which is in fact with uniform priors, to find out which priors are more suitable to the models on subjectivity classification task. Detailed experiments and analysis are given in the following section.

5 Experiments

5.1 Dataset and Toolkit

The subjective/objective corpus we used here was obtained from Pang's webpage (see footnote ① in Section 1), which was first used by them in [10]. This dataset consists of 10 000 movie review sentences or snippets, 5000 labeled objective and 5000 labeled subjective. The objective ones were taken from plot summaries of the Internet Movie Database (IMDb)②, and the subjective ones were retrieved from ROTTEN TOMATOES website③. We measured the performances of different methods mentioned above using 3-fold cross validations.

All the implementations of MaxEnt models reported in this paper are modified on the basis of SS MaxEnt toolkit④.

5.2 Evaluation Measure

Subjectivity analysis task being studied in this paper can be viewed as a special binary (subjective/objective) classification. Referring to the TREC spam track⑤, which is a similar binary (ham/spam filtering) classification task, we adopted the Logistic Average Misclassification Percentage (LAMP)[21] as our evaluation measure. The LAMP to our task is defined as:

$$lam\% = \text{logit}^{-1}(\text{logit}(subm\%) + \text{logit}(objm\%)).$$

Wherein, $\text{logit}(x) = \log \frac{x}{1-x}$, and its inverse function $\text{logit}^{-1}(x) = \frac{e^x}{1+e^x}$. The two factors $subm\%$ and $objm\%$ are the misclassification percentages of subjective and objective samples respectively. The lower $labm\%$ value, the better classifier.

5.3 TFIDF-Based Feature Extraction Methods

The result reported in our previous work[11] are listed in Table 1, the best result is LAMP 0.10845, which was achieved while using normalized unigram and bigram features within a 2-distance context window. This value will be treated as the baseline to the following experiments.

Table 1. Results Reported by Chen *et al.* in [11]

LAMP	Average	Minimum	Maximum
UNI	0.12315	0.11116	0.13563
LB1	0.12255	0.11318	0.13089
LB2	**0.12216**	**0.10845**	**0.13106**
LB3	0.12449	0.10948	0.13178
LB4	0.12518	0.11122	0.13974
LB5	0.12846	0.11830	0.14286

Using the TFIDF-based Global-Filtering and Local-Weighting strategy, we constructed three series of normal MaxEnt models according to three different kinds of feature sets respectively. The first feature set only contains unigram language features, denoted by "UNI" in the following description. The second set are with bigram features added in, denoted by "UB" hereafter. The third one consists of unigram, bigram and trigram language features, denoted as "UBT". On each feature set, we implemented a series of feature extractions according to a sequence of incremental threshold values on the global TFIDF weights. The performances of those three series of MaxEnt models are shown in Fig.1, compared to the baseline value.

From the results shown in Fig.1, it is obvious that the TFIDF-based feature weighting method is much better than the normalized-frequency-based methods, which only make use of local information of language features within individual text snippets. On the other hand, while feature filtering is applied according to some threshold values, the performances of models on both UB and UBT sets get improved. This illustrates the effectiveness of our Global-Filtering and Local-Weighting strategy on language feature extraction. As

②http://www.imdb.com/

③http://www.rottentomatoes.com

④From Tsujii laboratory, at University of Tokyo, on http://www-tsujii.is.s.u-tokyo.ac.jp/∼tsuruoka/maxent/

⑤http://trec.nist.gov/tracks.html

the threshold value goes beyond 0.005, performances on each set drop down sharply. At this point, there are nearly 66% of the overall language features remained in UNI set, while there are merely 18% and 8.7% in UB and UBT sets respectively. This may indicates that most useful language features are going to be deleted while a stricter filtering is applied. It may also be concluded from the results that many redundant features get global TFIDF weights less than 0.003.

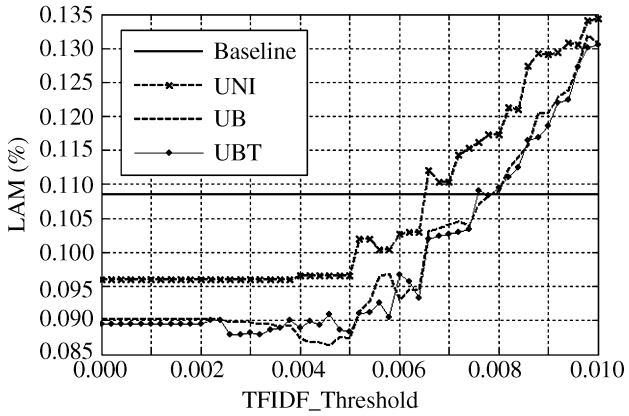

Fig.1. TFIDF-based feature extractions.

Another important point is that the combination of high-order language features, like bigrams and trigrams, is certain to improve the MaxEnt models. This is consistent with the results reported in [11].

5.3 Maximum Entropy Models with Priors

Besides the language feature extraction, we are also interested in the language modeling. As mentioned in Subsection 4.3, we will investigate MaxEnt models with three kinds of priors to find out with which kinds of priors on, the MaxEnt models satisfy the subjectivity classification task the best. For this purpose, firstly, we will analysis the distribution of the language features over the dataset to find out the optimal prior. We made a simple statistics on the distribution of term frequencies in the UNI feature set, as displayed in Fig.2.

It is obvious that the distribution of the unigram language features f_i is a typical exponential distribution. For that the importance of language features, which are reflected by the parameter λ_is assigned by a MaxEnt model, are mainly represented by their occurrence frequencies. Therefore, we may assume that the prior distribution of the parameters is also exponential.

We have constructed MaxEnt models with uniform priors, Gaussian priors and exponential priors respectively, and applied them to the three feature sets to make complete comparisons, as shown in Figs. 3∼5, wherein "UNF" stands for the uniform prior, "GAU" for Gaussian prior, and "EXP" for exponential prior.

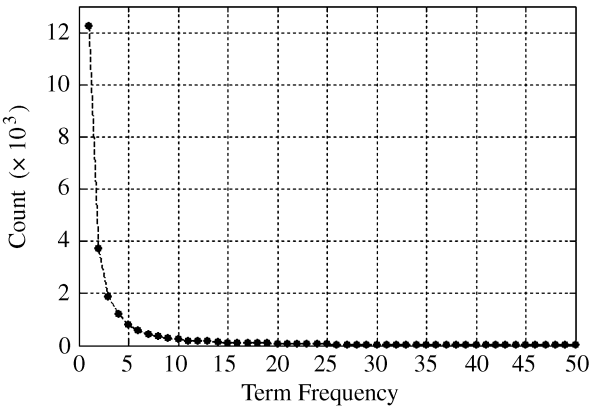

Fig.2. Distribution of term frequency in UNI set.

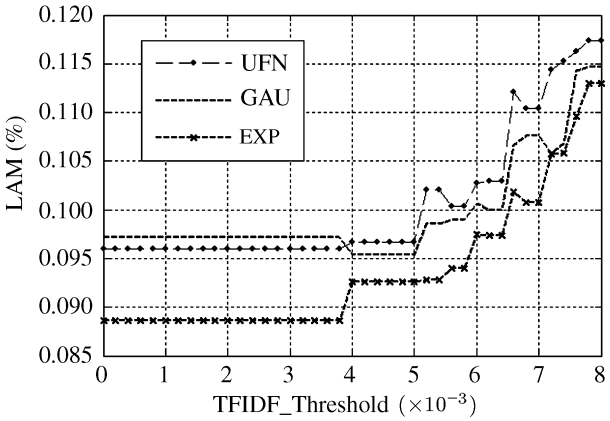

Fig.3. Models with different priors on UNI set.

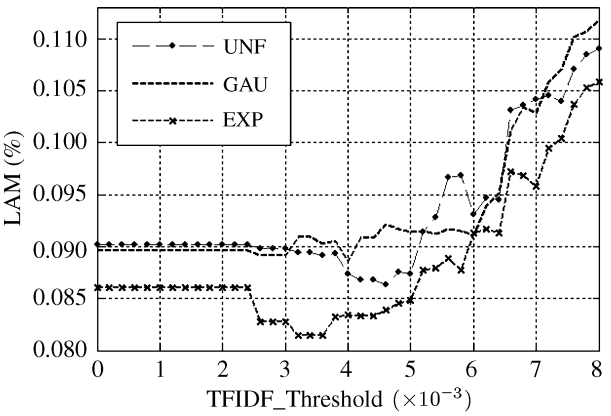

Fig.4. Models with different priors on UB set.

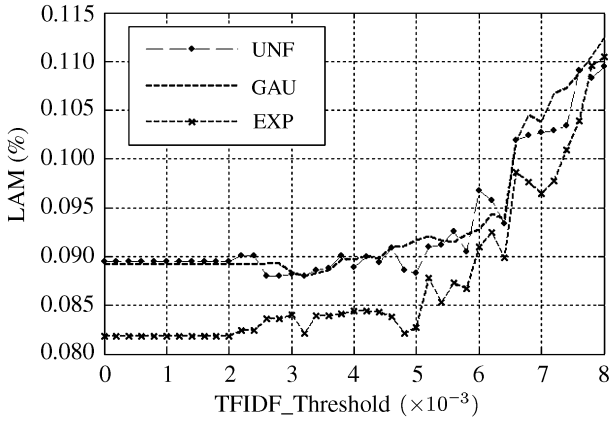

Fig.5. Models with different priors on UBT set.

In the comparisons on three feature sets, the MaxEnt models with exponential priors significantly surpass the models with other priors. Meanwhile, models with Gaussian priors even do no better than the normal MaxEnt models. This illustrates that the exponential prior fits the parameter distribution well. To make a further confirmation, we make another statistics on the distribution of parameter λ_is' values of the MaxEnt model with uniform priors trained on the UNI feature set, for that such a MaxEnt model is without particular priors, its parameter values would reveal the real distribution. To make the statistics realizable, we first perform a quantizing on λ_is' values with an interval step of 0.1. As shown in Fig.6, regardless of the polarity of values, the distribution of the parameter values is obviously exponential. This strongly backs up the conclusion that MaxEnt models with exponential priors are more suitable for the subjectivity analysis task.

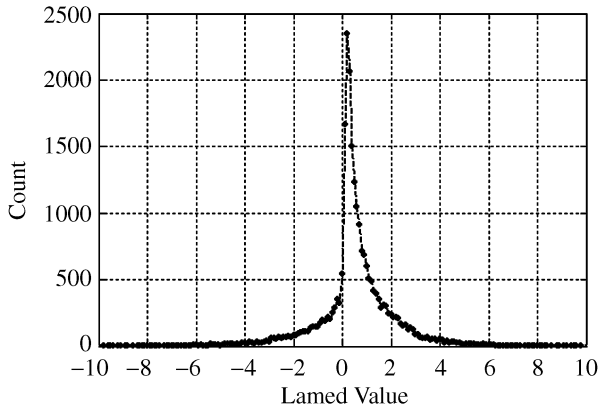

Fig.6. Distribution of Parameter values of the MaxEnt model with uniform prior trained on UNI set.

5.4 Comparison of Different Language Features

Since the superiority of the MaxEnt models with exponential priors has been illustrated, the following experiments are all based on them. As mentioned in Subsection 2.2, there are some interesting issues reported in previous works. In this section, we will investigate more kinds of features, mainly on bigrams within long-distance-windows. Experimental results are listed in Table 2.

Table 2. Comparison of Different Language Features

	Best LAMP	Improvement
Baseline	0.108 45	–
UNI	0.088 67	18.24%
UB1	**0.081 46**	**24.89%**
UB2	0.083 31	23.18%
UB3	0.083 26	23.23%
UB4	0.084 91	21.71%
UB5	0.085 11	21.52%
UBT	0.081 88	24.50%

In this table, "UBd" indicates that the feature set is comprised by unigram and bigram language features within a d-distance-window. Results listed in this table are the best ones on each feature set with a series of thresholds. Once again, the results show that high-order language features can improve the performances of models. However, in our previous work[11], the best models was trained on the feature set corresponding to "UB2" here, while results here show no improvement are achieved as long-distance language features being added in. This is a little confused, and we will try to find the reason.

6 Conclusions and Future Work

In this paper, we make a case study on constructing language models for document subjectivity analysis. Emphasizes are put on language feature extraction and language modeling. On the basis of TFIDF weighting scheme, we conduct a Global-Filtering and Local-Weighting strategy to improve language feature extraction. While constructing language models, we utilize the Maximum Entropy framework and put some priors on to make models more suitable for the task. Analyzing the experiments, we think that using high-order language grams within context windows and applying exponential priors on MaxEnt models will be quite helpful.

Our work is still worth further studying. The n-gram based feature representation is useful but not delicate enough, more sophisticated language features may be

useful to particular kind of documents, and more natural language processing techniques are needed. There is still some room to improve MaxEnt modeling. And also, some other machine learning algorithms can be applied to construct language models.

References

[1] Das S R, Chen M Y. Yahoo! for Amazon: Sentiment extraction from small talk on the web. Working paper, Santa Clara University, Available at http://scumis.scu.edu/srdas/chat.pdf.

[2] Chesley P, Vincent B, Xu L, Srihari R. Using verbs and adjectives to automatically classify blog sentiment. In *Proc. Computational Approaches to Analyzing Weblogs*: Papers from the 2006 Spring Symposium, Nicolov N, Salvetti F, Liberman M, Maartin J H (eds.), AAAI Press, Menlo Park, CA, Technical Report SS-06-03, 2006, pp.27–29.

[3] Gamon M. Sentiment classification on customer feedback data: Noisy data, large feature vectors, and the role of language analysis. In *Proc. 20th Int. Conf. Computational Languages*, Geneva, CH, 2004, pp.841–847.

[4] Kennedy A, Inkpen D. Sentiment classification of movie and product reviews using contextual valence shifters. *Computational Intelligence*, 2006, 22(2): 110–125.

[5] Berger A L, Della Pietra S A, Della Pietra V J. A maximum entropy approach to natural language processing. *Computational Languages*, 1996, 22(1): 39–71.

[6] Rosenfeld R. A maximum entropy approach to adaptive statistical language modeling. *Computer, Speech and Language*, 1996, 10: 187–228.

[7] Sebastiani F. Machine learning in automated text categorization: A survey. *Tech. Rep. IEI-B4-31-1999*, Istituto di Elaborazione dell'Informazione, Consiglio Nazionale delle Ricerche, Pisa, IT, 1999.

[8] Yang Y. An evaluation of statistical approaches to text categorization. *Journal of Information Retrieval*, 1999, 1: 69–90.

[9] Pang B, Lee L, Vaithyanathan S. Thumbs up? Sentiment classification using machine learning techniques. In *Proc. Conf. Empirical Methods in Natural Language Processing*, Philadelphia, US, 2002, pp.79–86.

[10] Pang B, Lee L. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proc. 42nd Meeting of the Association for Computational Languages*, Barcelona, ES, 2004, pp.271–278.

[11] Chen B, He H, Guo J. Language feature mining for document subjectivity analysis. In *Proc. 1st Int. Symp. Data, Privacy, & E-Commerce*, Chengdu, China, November 1–3, 2007, pp.62–67.

[12] Huang X D, Alleva F, Hon H W, Hwang M Y, Lee K F, Rosenfeld R. The SPHINX-II speech recognition system: An overview. *Computer, Speech and Language*, 1993, 2: 137–148.

[13] Salton G, Buckley C. Term-weighting approaches in automatic text retrieval. *Information Processing and Management*, 1988, 24(5): 513–523.

[14] Della Pietra S A, Della Pietra V J, Lafferty J. Inducing features of random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1997, 19(4): 380–393.

[15] Bahl L, Jelinek F, Mercer R. A maximum likelihood approach to continuous speech recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1983, 5(2): 179–190.

[16] Chen S F, Goodman J. An empirical study of smoothing techniques for language modeling. Tech. Rep. TR-10-98, Harvard University, 1998.

[17] Berger A. Convexity, maximum likelihood and all that, 1996. http://www.cs.cmu.edu/afs/cs/user/aberger/www/ps/convex.ps.

[18] Chen S F, Rosenfeld R. A Gaussian prior for smoothing maximum entropy models. *Tech. Rep. CMUCS-99-108*, Carnegie Mellon University, 1999.

[19] Kazama J, Tsujii J. Evaluation and extension of maximum entropy models with inequality constraints. In *Proc. EMNLP 2003*, 2003, pp.137–144.

[20] Goodman J. Exponential priors for maximum entropy models. *Microsoft Research Tech. Rep.*, 2003.

[21] Cormack G. TREC 2006 spam track overview. In *Proc. TREC 2006*, Gaithersburg, MD, 2006.

Bo Chen received his B.E. degree from North China Electric Power University, China in 2003, and entered Beijing University of Posts and Telecommunications (BUPT) for his M.E. degree in 2003. At present, he is a Ph.D. candidate in School of Information Engineering, BUPT, China. His research interests include pattern recognition, machine learning, information retrieval, information extraction, and natural language processing.

Hui He received her B.E. and M.E. degrees from North China Electric Power University, in 2003 and 2006 respectively. At present, she is a Ph.D. candidate in School of Information Engineering, Beijing University of Posts and Telecommunications, China. Her research interests include pattern recognition, text clustering, web information extraction and data mining.

Jun Guo received the B.E. and M.E. degrees from Beijing University of Posts and Telecommunications, China in 1982 and 1985, respectively, the Ph.D. degree from the Tohuku-Gakuin University, Japan in 1993. At present he is a professor and the dean of School of Information Engineering, BUPT. His research interests include pattern recognition theory and application, information retrieval, content based information security, and network management.