The Role of Word Statistical Features in Automatic Sentiment Word Extraction and Product Review Classification (Postprint)

Authors: Han Tonghui, Yang Dongqiang, Ma Hongwei

Date: 2018-05-18T00:00:00+00:00

Abstract

Statistical features of words have extensive applications in natural language processing. Focusing on the impact of statistical features on the accuracy of keyword extraction and text classification, we analyze eight common statistical features and investigate the role of statistical features in sentiment analysis through sentiment word extraction and product review classification. Experimental results on sentiment word extraction demonstrate that by combining statistical features with part-of-speech information, the accuracy of sentiment word extraction can reach 76.4%, which is significantly higher than algorithms based solely on statistical features or word part-of-speech. Test results on product review classification show that compared with traditional word-based text sentiment classification, the accuracy of statistical feature-based product review classification improves by 10.8%. Utilizing eight statistical features to construct a text vector space model, as an alternative to word-based text vector space model construction methods, can reduce the dimensionality of text vectors and possesses a compression effect similar to Latent Semantic Analysis/Singular Value Decomposition (LSA/SVD), effectively reducing algorithmic complexity while ensuring classification accuracy, thus capable of replacing traditional vector space models.

Full Text

The Role of Lexical Statistical Features in Automatic Sentiment Word Extraction and Product Review Classification

Han Tonghui, Yang Dongqiang, Ma Hongwei
School of Computer Science and Technology, Shandong Jianzhu University, Jinan 250100, China

Abstract: Lexical statistical features are widely used in natural language processing. This paper analyzes eight types of statistical features and investigates their role in sentiment word extraction and product review classification. Experimental results demonstrate that combining statistical features with part-of-speech information achieves a sentiment word extraction accuracy of 76.4%, significantly outperforming methods based solely on statistical features or part-of-speech tags. Product review classification tests show that using statistical features improves classification accuracy by 10.8% compared to traditional word-based sentiment classification. By constructing text vector space models using eight statistical features instead of individual words, the dimensionality of text vectors is reduced, achieving a compression effect similar to latent semantic analysis (LSA/SVD). This approach effectively reduces algorithmic complexity while maintaining classification accuracy, offering a viable alternative to traditional vector space models.

Keywords: statistical features; sentiment word extraction; product review classification

0 Introduction

Text sentiment analysis has emerged as a prominent research area in natural language processing with important applications in public opinion monitoring and product review systems. Sentiment word extraction forms the foundation of sentiment analysis, where extraction precision and coverage directly impact the construction of sentiment lexicons, text sentiment classification, and sentiment intensity computation. Grammar rule-based sentiment word extraction algorithms represent an easily implementable approach to automatic sentiment word identification. For instance, Qiu et al. mined syntactic relationships between sentiment words and topic words based on probabilistic word associations, simultaneously expanding both sentiment and topic word collections. Liu et al. improved algorithmic efficiency by incorporating semantic similarity on top of syntactic rules. However, these methods are limited to adjectives, whereas in real-world language environments, sentiment words are not restricted to this part of speech alone. The use of lexical statistical features can overcome limitations imposed by part-of-speech and domain dependencies while demonstrating good adaptability across different languages.

This paper analyzes eight common statistical features in natural language processing and examines their effectiveness in sentiment word extraction and product review classification. Sentiment word extraction experiments reveal that algorithms combining statistical features with part-of-speech information achieve significantly higher precision than other commonly used methods. Product review classification results demonstrate that low-dimensional vector space models constructed from eight statistical features can improve classifier accuracy while effectively reducing computational and space complexity.

1 Related Research

Lexical statistical features numerically reflect the association between words and text categories, providing a basis for keyword extraction. Pointwise Mutual Information (PMI) represents a typical statistical feature-based sentiment word extraction algorithm that can also determine word sentiment polarity based on PMI values. Aliaksei et al. treated words appearing in corpora as features and employed linear classification algorithms for sentiment classification, using optimized parameter vectors as standards for sentiment word extraction. Yu et al. argued that sentiment words contribute far more to text sentiment polarity than non-sentiment words, proposing to extract sentiment words by calculating word weights within texts given known sentiment polarities. However, these implementations are overly complex, and PMI-based extraction algorithms exhibit limited reliability in certain domains.

Text sentiment classification trains models based on word distribution features, enabling the construction of vector representations using sentiment words as special keywords. Consequently, statistical feature-based keyword extraction algorithms are equally applicable to sentiment word extraction. For example, Rajeswari et al. used information gain for keyword extraction, while Uysal employed information gain, odds ratio, and Gini coefficient. McAuley extracted latent keywords within LDA models, Chen mined keywords through LDA models and classified them based on frequency distribution, Mesleh weighted words using chi-square tests, Mitra used correlation coefficients between words and texts as primary features, and Juola computed associations between texts and words via cross-entropy. Although statistical features intuitively reflect word-text associations, threshold determination remains a challenge.

This paper examines the role of eight common statistical features in sentiment word extraction and text sentiment classification. Machine learning-based methods are used for text sentiment classification to evaluate the optimization capability of vector space models constructed from these eight statistical features.

2 Feature Value Computation and Data Representation

We investigate eight statistical features in sequence: Information Gain (IG), Odds Ratio (OR), Mutual Information (MI), Logarithmic Probability Ratio (LPR), Cross Entropy (CE), Chi-square Test (CHI), Correlation Coefficient (CC), and Differential Distribution (DD), analyzing their effectiveness in sentiment word extraction and product review classification.

2.1 Feature Value Computation

Let C denote text sentiment type, where C {pos, neg} (pos for positive, neg for negative). P(C) represents the probability of sentiment type C, and P(¬C) represents the probability of non-C type, where P(¬C) = 1 - P(C). Let T denote text and w denote word. P(w) represents the probability of w appearing in T, P(¬w) represents the probability of w not appearing, where P(¬w) = 1 - P(w).

To facilitate statistical feature computation, we create a quadruple Q = p, q, n, q' to store word distribution information, where p represents the frequency of positive texts containing w, q represents positive texts not containing w, n represents negative texts containing w, and q' represents negative texts not containing w. Table 1 lists the probability approximation formulas used in feature value calculation.

The computation processes for these feature values are similar. We illustrate using the word 'good' as an example. Based on 'good' s distribution information, we construct quadruple Q_good. Statistics show that positive and negative text frequencies containing ' good' are 9,974 and 5,978 respectively, while those not containing 'good' are 23,400 and 27,464. Thus Q_good = 9974, 23400, 5978, 27464 . Substituting Q_good into Table 1 yields the relevant probabilities needed for computing ' good' s information gain, with results shown in Table 3 . Substituting these probabilities into Equation (1) gives IG(good) = 1.0003.

Standard OR, LPR, and CHI algorithms tend to assign high weights to low-frequency words, making many low-frequency non-sentiment words overly influential and reducing reliability. To improve robustness, we multiply the standard algorithms by word probability P(w).

Taking OR as an example, Table 2 lists OR values for a set of words and their rankings in feature value lists. The table shows noticeable changes in word ranking before and after OR algorithm improvement. Sentiment word extraction experiments based on OR demonstrate that the improved version achieves 17.8% higher accuracy than the standard version.

1) Information Gain (IG)
Word information gain represents the information carried for distinguishing text sentiment types. Higher IG indicates stronger discriminative capability:

$$IG(w) = -\sum_{C\in\{pos,neg\}} P(C)\log P(C) + \sum_{t\in\{w,\neg w\}} P(t) \sum_{C\in\{pos,neg\}} P(C|t)\log P(C|t)$$

2) Improved Odds Ratio (OR)
OR reflects a word's ability to influence text sentiment polarity. Higher absolute OR values indicate stronger influence:

$$OR(w) = \frac{P(w|C)\times(1-P(w|\neg C))}{(1-P(w|C))\times P(w|\neg C)}$$

3) Mutual Information (MI)
MI measures the information word w carries about text sentiment type. Higher MI indicates greater information content:

$$MI(w) = \sum_{C \in \{pos,neg\}} P(w,C) \log \frac{P(w,C)}{P(w)P(C)}$$

The final MI value for w is: $MI(w) = \max_{C \in \{pos,neg\}} MI(w,C)$

4) Improved Logarithmic Probability Ratio (LPR)
LPR uses the logarithmic ratio of word probabilities in positive versus negative texts as an information measure. Larger absolute values indicate stronger discriminative power:

$$LPR(w) = \log \frac{P(w|C)}{P(w|\neg C)}$$

5) Cross Entropy (CE)
CE describes distribution differences of words between positive and negative texts. Higher CE indicates more distinct distributions and higher probability of being a sentiment word:

$$CE(w) = \sum_{C \in \{pos,neg\}} P(C|w) \log \frac{P(C|w)}{P(C)}$$

6) Improved Chi-square Test (CHI)
CHI tests association strength between words and sentiment types, assuming a chi-square distribution with 1 degree of freedom. Higher chi-square values indicate stronger associations:

$$\chi^2(w,C) = \frac{(A \times D - B \times E)^2}{(A+B)(A+E)(B+D)(D+E)}$$

where A = count of texts with sentiment C containing w, B = count of texts with non-C sentiment containing w, E = count of texts with sentiment C not containing w, and D = count of texts with non-C sentiment not containing w. The final CHI value is: $\chi^2(w) = \max_{C \in \{pos,neg\}} \chi^2(w,C)$

7) Correlation Coefficient (CC)
CC measures correlation between words and sentiment polarity. Larger coefficients indicate stronger discriminative ability:

$$CC(w) = \frac{P(w,C) - P(w)P(C)}{\sqrt{P(w)P(C)P(\neg w)P(\neg C)}}$$

8) Differential Distribution (DD)
DD measures distribution differences between positive and negative texts. If w' s frequency in C-type texts is significantly higher (or lower) than in non-C texts, w is likely a sentiment word with polarity same as (or opposite to) the text polarity:

$$DD(w) = \frac{P(C|w) - P(\neg C|w)}{\max(P(pos|w), P(neg|w))} \times P(w)$$

DD(w) ranges in [-1,1]. Multiplying by P(w) reduces noise impact.

2.2 Automatic Sentiment Word Extraction

Sentiment word extraction essentially discretizes continuous features to assign words to sentiment or non-sentiment sets. To partition word sets efficiently and reasonably, we employ the Standard Deviation Reduction (SDR) algorithm to determine thresholds for statistical features. SDR dynamically allocates words to appropriate sets and calculates error reduction after each allocation. When error reduction reaches maximum, the partition is optimal:

$$value = sd(L) - \frac{|L_s|}{|L|} \times sd(L_s) - \frac{|L_n|}{|L|} \times sd(L_n)$$

where L is the list of candidate sentiment word feature values sorted in descending order, L_s is the sentiment word feature value list, L_n is the non-sentiment word list (L = L_s + L_n), $|\cdot|$ denotes set/list size, and sd($\cdot$) is the standard deviation function. When value reaches maximum, the partition between sentiment and non-sentiment words is optimal, and the maximum feature value in L_n becomes the threshold for that statistical feature. Algorithm 1 describes the SDR process.

To demonstrate SDR execution, we use information gain as an example with a sample list L containing 10 IG values, determining the threshold through SDR. Table 4 shows L sorted in descending order. Initially, the sample set S is partitioned into L_s = {IG_disappoint} and L_n = {IG_happy ~ IG_well}, both sorted descending. During execution, when value > V, we update V with value and threshold with the maximum feature value in L_n. After processing, the final threshold for this IG sample is 0.8053.

2.3 Data Representation in Sentiment Analysis

We create feature vectors for words using the statistical features from Section 2.1, enabling vector representation:

$$V_w = \langle f_1, f_2, f_3, f_4, f_5, f_6, f_7, f_8 \rangle$$

where vector elements f_1 through f_8 correspond to the eight statistical features.

Using vector addition from semantic composition, we construct vector space models to represent texts:

$$V_T = \sum_{i=1}^{t} sig(w_i) \times V_{w_i}$$

where w_i is the i-th word in the sentiment dictionary, sig(w_i) is a sign function (sig(w_i) = 1 if w_i appears in T, otherwise 0), and V_{w_i} is the word vector. This vector representation is detailed further in Section 4.3.2.

---

3 Sentiment Word Extraction and Product Review Classification

Although Chinese e-commerce sites provide abundant product reviews, existing Chinese word segmentation tools have limitations, and these reviews contain substantial advertising and fake content, making it difficult to effectively validate algorithm performance. English reviews reduce segmentation error impact, and Amazon' s English reviews are relatively more authentic. Therefore, we adopt English product reviews for our experiments.

3.1 Text Preprocessing

Product reviews collected from websites contain numerous stop words and abbreviations, requiring preprocessing:

a) **Normalization**: Convert uppercase letters to lowercase, filter special symbols (e.g., #, @) and stop words (e.g., this, that), expand abbreviations (e.g., that' s → that is), and unify negative adverbs to "not" (e.g., hardly → not).

b) **Stemming**: Reduce words to base forms for plural nouns, comparative adjectives, past tense verbs, etc. (e.g., issues → issue, better → good).

c) **Phrase Extraction**: Multiple consecutive words with syntactic relationships can express sentiment (e.g., "meet my expectation", "not buy again"). We extract such sentiment phrases based on syntactic connections.

d) **Candidate List Construction**: Build candidate sentiment word list L storing words and phrases from texts without duplicates.

e) **Frequency Filtering**: Remove words with frequency below (set to in our experiments, where {IG, OR, MI, LPR, CE, CHI, CC, DD}). If word w' s feature value f_w > (threshold), w is considered a sentiment word satisfying statistical feature .

Table 5 shows vocabulary size changes and counts for adjectives, verbs, nouns, and adverbs before and after preprocessing.

3.2 Sentiment Word Extraction

We define extracted words as follows: For a single statistical feature with threshold , if word w' s feature value f_w > , w is a sentiment word satisfying . Beyond testing single-feature extraction, we evaluate multi-feature methods using eight extraction standards C_1 through C_8, where C_i (i [1,8]) requires words to satisfy at least i statistical features.

3.2.1 Single Statistical Feature Extraction Create sentiment dictionary D_ for each selected feature . Find the corresponding feature value list and threshold, then traverse the list, comparing each word' s feature value with the threshold. Words with values exceeding the threshold are added to the dictionary.

Using IG-based extraction as an example: Create dictionary D_IG initially empty, with feature list L_f1 and threshold _f1 = 0.1017. Traverse L_f1, comparing each word' s IG with _f1. Words with IG > _f1 are added to D_IG. Table 6 shows that "luck" has IG_luck = 0.1033 and is added, while "refuse" with IG_refuse = 0.0814 is not.

3.2.2 Multi-Statistical Feature Extraction Multi-feature extraction requires words to satisfy at least i features according to standards C_1~C_8 from Section 3.2. Algorithm 2 shows the multi-feature extraction process.

Using "refuse" as an example: The algorithm reads "refuse" from the candidate list, initializes variable I = 0, then traverses feature lists L_f1~L_f8 to find "refuse' s" eight feature values. Comparing each with corresponding thresholds _f1~ _f8, I increments when f_i > _fi. Table 6 shows "refuse" satisfies OR, MI, and CC, resulting in I = 3. Since I > 0, we construct its word vector:

$$V_{refuse} = \langle 0.0164, 0.0069, 0.0032, 0.0067, 9.3314, 0.0019, 4.6780, 0.0060 \rangle$$

We store "refuse" and V_refuse in dictionaries D_1~D_3 and remove it from the candidate list. The process continues until the candidate list is empty.

3.2.3 Combining Statistical Features with Part-of-Speech The combined single-feature and POS algorithm requires words to both exceed feature thresholds and be adjectives. As Table 6 shows, "luck" satisfies IG but is a noun and thus excluded. "awful" is an adjective with IG_awful > _f1, so it is extracted.

The multi-feature with POS algorithm (similar to Algorithm 2) adds a POS check at line 10: candidates must be adjectives to proceed. Table 6 shows

both "cheap" and "refuse" satisfy three features, but only "cheap" (adjective) is extracted while "refuse" (verb) is filtered.

3.2.4 Extraction Results HowNet provides rich resources for sentiment classification, containing 9,142 English evaluation words. We construct a standard dictionary from HowNet words also appearing in our corpus to evaluate extraction efficiency.

Table 7 shows that combining statistical features with POS yields higher accuracy, averaging 36.8% improvement over single-feature methods and up to 21.7% improvement over POS-only methods.

Figure 1 Figure 1: see original paper displays multi-feature extraction results: D_1~D_8 represent dictionaries created under C_1~C_8. Accuracy is lowest when satisfying at least one feature (C_1) and highest when satisfying all eight (C_8). Figure 1(b) shows multi-feature with POS extraction: compared to multi-feature alone, accuracy improves by 41.9% under C_1 and 31.1% under C_8.

3.3 Product Review Classification

We test both single-feature and multi-feature constructed dictionaries for classification, using five algorithms: Naïve Bayes, SVM, Decision Tree, BP Neural Network, and Random Forest. Test texts are Amazon product reviews (997 positive, 999 negative), processed using Weka with 10-fold cross-validation and default parameters. POS-based classification serves as baseline, achieving accuracies of 77.1%, 74.5%, 69.5%, 63.0%, and 76.3% respectively.

3.3.1 Single Statistical Feature Classification Using single-feature dictionaries with traditional word-based vector space models, we represent texts with binary vectors (0/1) as Pang et al. demonstrated this performs better:

$$V_T = \langle sig(w_1), sig(w_2), ..., sig(w_t) \rangle$$

where t is dictionary size and sig(w_i) = 1 if w_i appears in T, otherwise 0.

3.3.2 Multi-Statistical Feature Classification Using dictionaries D_1~D_8, we construct vector space models from the eight statistical features instead of traditional word-based representations. Algorithm 3 describes text vector construction using statistical features.

For example, processing review T: "Very nice, sleek and works great. My grandkids talked me into it, it' s what they use in school. So far I love it." Using dictionary D_1, we find T D_1 = { 'nice' , 'great' , 'love' }. With sig(nice) = sig(great) = sig(love) = 1, we compute V_T = V_nice + V_great + V_love:

$$V_T = \langle 10.0177, 4.4538, 0.196, 3.8376, 55.6348, 5.654, 177.1325, 0.3116 \rangle$$

3.3.3 Combining Statistical Features with POS Single-feature with POS classification uses word-based vectors (same as Section 3.3.1). Multi-feature with POS classification uses eight statistical features for text representation (same method as Section 3.3.2) but differs in extracted words. For the same review, multi-feature with POS only uses adjectives 'nice' and 'great' , yielding:

$$V_T = V_{nice} + V_{great} = \langle 9.6516, 3.6138, 0.1808, 3.0351, 44.1894, 5.5982, 155.279, 0.2858 \rangle$$

3.3.4 Classification Results Table 8(a) shows single-feature classification results: correlation coefficient achieves highest accuracy (87.1%) with Naïve Bayes, while cross-entropy only reaches 55.7% with BP-neural networks. Table 8(b) shows single-feature with POS results, with accuracies concentrated between 66%-72%.

Table 9(a) presents multi-feature classification: all algorithms achieve best performance when words satisfy at least one feature (C_1). Multi-feature with POS results (Table 9(b)) show accuracies between 69%-71%.

---

4 Experimental Results Analysis

Since people tend to express emotions using adjectives, POS-based extraction achieves higher precision than statistical feature-based methods. However, in real language environments, verbs, adverbs, and nouns also express sentiment (e.g., 'love' , 'kindly' , 'issue'), making POS-based classification less accurate than statistical feature-based approaches.

Combining statistical features with POS improves extraction accuracy (Tables 7, Figure 1) but increases constraints, reducing the number of qualifying words and potentially decreasing classification precision. For example, 'love' satisfies statistical features but is filtered as a verb.

Figure 1(a) shows multi-feature extraction achieves highest precision under C_8 (all features satisfied) and lowest under C_1 (at least one feature). Conversely, Table 9(a) shows C_1-based dictionaries achieve highest classification accuracy while C_8-based dictionaries perform worst. This occurs because stricter standards (higher i) reduce the number of qualifying words, limiting dictionary coverage. For instance, 'cheap' expresses price opinions but only appears in D_1~D3, causing classification algorithms to miss relevant reviews.

Comparing Table 8(a) and 9(a), single-feature classification outperforms multi-feature in Naïve Bayes, while multi-feature excels in the other four classifiers.

Multi-feature classification using eight statistical features reduces vector dimensionality, achieving LSA/SVD-like compression effects that decrease time and space complexity while maintaining accuracy.

---

5 Conclusion

This paper selected eight common statistical features in natural language processing and investigated their role in sentiment word extraction and product review classification. Experiments demonstrate that statistical feature-based sentiment classification achieves higher precision. Multi-feature classification using eight statistical features to construct vector space models effectively reduces algorithmic time and space complexity while maintaining accuracy and recall.

Future work will improve sentence segmentation algorithms to mine internet slang, study statistical feature weights across different classifiers, and enable automatic weight assignment to improve classification efficiency.

References

[1] Tang D Y, Wei F R, Qin B, et al. Building large-scale twitter-specific sentiment lexicon: a representation learning approach [C]// Proc of the 25th International Conference on Computational Linguistics. 2014: 23-29.

[2] Ibrahim H S, Sherif M A, Gheith M H. Sentiment analysis for modern standard Arabic and colloquial [J]. International Journal on Natural Language Computing, 2015, 4 (2): 95-109.

[3] Wang F X, Zhang Z H, Lan M. ECNU at SemEval-2016 task 7: an enhanced supervised learning method for lexicon sentiment intensity ranking [C]// Proc of the International Workshop on Semantic Evaluation. 2016: 491-496.

[4] Mohammad S M, Bravo-Marquez F. Wassa-2017 shared task on emotion intensity [C]// Proc of the Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis. 2017.

[5] Qiu G, Liu B, Bu J J, et al. Opinion word expansion and target extraction through double propagation [J]. Computational Linguistics, 2011, 37 (1): 9-27.

[6] Liu K, Xu L H, Zhao J. Extracting opinion targets and opinion words from online reviews with graph co-ranking [C]// Proc of the 52nd Annual Meeting of the Association for Computational Linguistics. 2014: 314-324.

[7] Chetviorkin I, Loukachevitch N. Domex: extraction of sentiment lexicons for domains and meta-domains [C]// Proc of COLING. 2012: 77-86.

[8] Jovanoski D, Pachovski V, Nakov P. On the impact of seed words on sentiment polarity lexicon induction [C]// Proc of COLING. 2016: 11-17.

[9] Severyn A, Moschitti A. On the automatic learning of sentiment lexicons [C]// Proc of Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. 2015: 1397-1402.

[10] Yu H L, Deng Z H, Li S Y. Identifying sentiment words using an optimization-based model without seedwords [C]// Proc of the 51st Annual Meeting of the Association for Computational Linguistics. 2013: 855-859.

[11] Rajeswari K, Nakil S, Patil N, et al. Text categorization optimization by a hybrid approach using multiple feature selection and feature extraction methods [J]. International Journal of Engineering Research and Applications, 2014, 4 (3): 86-90.

[12] Uysal A K. An improved global feature selection scheme for text classification [J]. Expert Systems With Applications, 2016, 43: 82-92.

[13] McAuley J, Leskovec J. Hidden factors and hidden topics: understanding rating dimensions with review Text [C]// Proc of the 7th ACM Conference on Recommender Systems. 2013: 165-172.

[14] Chen Z Y, Arjun Mukherjee, Liu B. Aspect extraction with automated prior knowledge learning [C]// Proc of the 52nd Annual Meeting of the Association for Computational Linguistics. 2014, 347-358.

[15] Mesleh A A. Chi square feature extraction based svms arabic language text categorization system [J]. Journal of Computer Science, 2007, 3 (6): 430-435.

[16] Mitra P, Murthy C A, Sankar K. P. Unsupervised feature selection using feature similarity [J]. IEEE Trans on Pattern Analysis and Machine Intelligence, 2002, 24 (3): 301-312.

[17] Juola P, Baayen H. A controlled-corpus experiment in authorship identification by cross-entropy [J]. Literary and Linguistic Computing, 2005, 20 (1): 59-67.

[18] Wang Y, Witten I. Inducing model trees for continuous classes [C]// Proc of the 9th European Conference on Machine Learning. 1997: 128-137.

[19] Zhou X J, Wan X J, Xiao J G. Collective opinion target extraction in Chinese microblogs [C]// Proc of Conference on Empirical Methods in Natural Language Processing. 2013: 1840-1850.

[20] Bakliwal A, Arora P, Patil A, et al. Towards enhanced opinion classification using NLP techniques [C]// Proc of Workshop on Sentiment Analysis where AI Meets. 2011: 101-107.

[21] Yoo J Y, Yang D M. Classification scheme of unstructured text document using tf-idf and naïve bayes classifier [J]. Advanced Science and Technology Letters, 2015, 111 (50): 263-266.

[22] Chen H, Zhan Y, Li Y. The application of decision tree in Chinese email classification [C]// Proc of the 9th International Conference on machine Learning and Cybernetics. 2010: 305-308.

[23] Zhang M L, Zhou Z H. Multi-label neural networks with applications to functional genomics and text categorization [J]. IEEE Trans on Knowledge and Data Engineering, 2006, 18 (10): 1338-1351.

[24] Moreira S, Filgueiras J, Martins B, et al. Reaction: a naive machine learning approach for sentiment classification [C]// Proc of the 2nd Joint Conference on Lexical and Computational Semantics. 2013: 490-494.

[25] Pang B, Lee L, Shivakumar Vaithyanathan. Thumbs up? sentiment classification using machine learning techniques [C]// Proc of Conference on Empirical Methods in Natural Language Processing. 2002: 79-86.

Note: Figure translations are in progress. See original paper for figures.