Contextual Latent Semantic Networks used for Document Classification

Ondrej Hava[1], Miroslav Skrbek[2] and Pavel Kordik[2]
[1]*Department of Computers, Faculty of Electrical Engineering, Czech Technical University in Prague, Prague, Czech Republic*
[2]*Department of Computer, Faculty of Information Technology, Czech Technical University in Prague, Prague, Czech Republic*

Keywords: Document Classification, Context Network, Latent Semantic Analysis, Structural Similarity of Networks.

Abstract: Widely used document classifiers are developed over a bag-of-words representation of documents. Latent semantic analysis based on singular value decomposition is often employed to reduce the dimensionality of such representation. This approach overlooks word order in a text that can improve the quality of classifier. We propose language independent method that records the context of particular word into a context network utilizing products of latent semantic analysis. Words' contexts networks are combined to one network that represents a document. A new document is classified based on a similarity between its network and training documents networks. The experiments show that proposed classifier achieves better performance than common classifiers especially when a foregoing reduction of dimensionality is significant.

1 INTRODUCTION

The document classification is common problem in the field of text mining. The classifiers are used to solve many tasks such as spam detection or sentiment analysis. The goal is to develop a supervised machine learning classifier over collection of labeled training documents which will be capable to assign unknown category or categories to any new document.

The quality of any classifier can be significantly influenced by the information extracted from the training collection of documents. Hence the transformation of plain text to structured data is the critical step.

The basic structured representation of text documents is a bag-of-words representation (Weiss et al., 2005) where each input feature corresponds to a given word or concept. Words or concepts constitute a vocabulary. The vocabularies of natural languages can be very rich; they often include thousands or tens-of-thousands items. Such huge number of input features should be reduced before building a classifier. The dimensionality reduction methods either filter out input features (Yang and Pedersen, 1997) or they provide smaller number of new extracted features (Marin, 2011). The latent semantic analysis based on well-known singular value decomposition (Landauer et al., 1998) is often used to extract a set of uncorrelated input variables that can be further reduced based on their importance. (Deerwester et al., 1990)

The classifiers learned over reduced number of features rely on different modeling techniques. The baseline algorithms for document classification include naïve Bayes (Eibe and Remco, 2006), logistic regression (Zhang and Oles, 2000) or k-nearest neighbors (kNN) (Han et al., 2001). Currently Support Vector Machines (SVM) (Vapnik, 1995) is a popular method in this area. Other algorithms include decision trees or neural networks.

An emerging insight into relations hidden in text documents is accessible through social networks. In the text mining field they are used in tasks such as link analysis (Berry et al., 2004) or information extraction (Gaizauskas and Wilks, 1998). Social networks can be also used as an advanced structured representation of documents. For example Kelleher (Kelleher, 2004) used two-mode social network to represent semantic relationship among documents. In our approach we propose to extend the classic bag-of-words representation of documents by a social network representation to encode order of

Hava O., Skrbek M. and Kordik P..
Contextual Latent Semantic Networks used for Document Classification.
DOI: 10.5220/0004109304250430
In *Proceedings of the International Conference on Knowledge Discovery and Information Retrieval* (SSTM-2012), pages 425-430
ISBN: 978-989-8565-29-7

words. Presented context networks also contain dimensionality reduction provided by latent semantics analysis.

We propose and tested the network representation of documents to perform the document classification. In the first section we review singular value decomposition (SVD) used for dimensionality reduction of bag-of-words document representation. In the second section we propose to build term context networks utilizing the products of SVD. The context networks are later aggregated to the document level. In the third section we introduce a similarity measure of networks and propose a kNN classifier. The experiments over a collection of Czech documents are described in the fourth and fifth sections. Conclusions are summarized in the sixth section.

2 SINGULAR VALUE DECOMPOSITION

Let us have a training collection *C* of N documents D_n, n=1..N.

$$C = \{D_1, D_2, \dots, D_N\} \tag{1}$$

Each document D_n is represented by a row vector $\mathbf{d}_n$.

$$\mathbf{d}_n = (d_{n1}, d_{n2}, \dots, d_{nM}) \tag{2}$$

The vector item d_{nm} is proportional to a frequency of a term W_m in document D_n. Terms can be words, phrases, n-grams or some other properties derived from a text. The set of M terms W_m, m=1..M, composes a vocabulary *V*.

$$V = \{W_1, W_2, \dots, W_M\} \tag{3}$$

Vocabulary terms are either known in advance or they are derived from the training collection of documents. The whole training collection can be described by matrix **D** of the rank L.

$$\mathbf{D} = \begin{pmatrix} d_{11} & d_{12} & \dots & d_{1M} \\ w_{21} & w_{22} & \dots & d_{2M} \\ \vdots & \vdots & \vdots & \vdots \\ d_{N1} & d_{N2} & \dots & d_{NM} \end{pmatrix} \tag{4}$$

Let us apply the singular value decomposition (SVD) to matrix **D** as

$$\mathbf{D} = \mathbf{P}\Lambda\mathbf{Q}^T \tag{5}$$

where **P** is NxL orthonormal matrix of column eigenvectors of $\mathbf{DD}^{\mathrm{T}}$, **Q** is MxL orthonormal matrix of column eigenvectors of $\mathbf{D}^{\mathrm{T}}\mathbf{D}$ and **Λ** is LxL diagonal matrix of singular values. Singular values are square roots of eigenvalues of $\mathbf{DD}^{\mathrm{T}}$ or $\mathbf{D}^{\mathrm{T}}\mathbf{D}$. These main properties of **P**, **Q** and **Λ** can be transcribed as

$$\begin{aligned} \mathbf{P}^T\mathbf{P} &= \mathbf{I} \\ \mathbf{Q}^T\mathbf{Q} &= \mathbf{I} \\ \Lambda^T &= \Lambda \end{aligned} \tag{6}$$

where **I** stands for identity matrix LxL. SVD enables to represent both documents and terms in a new orthogonal L-dimensional space. Let us call it the space of topics T_l, l=1..L. Extracted topics create a topic set *U*.

$$U = \{T_1, T_2, \dots, T_L\} \tag{7}$$

Matrix **P** is a projection matrix from document space to topic space while **Q** is a projection matrix from term space to topic space. Hence the projection of training documents to topic space is $\mathbf{DQ} = \mathbf{P}\Lambda$ while the projection of vocabulary terms to topic space is $\mathbf{D}^T\mathbf{P} = \mathbf{Q}\Lambda$.

SVD is often used to estimate original matrix **D** by a smaller number of topics. Topics of smaller importance are discarded from the topic set *U*. The importance of a topic is measured by a magnitude of its corresponding singular value. The deletion of topics means that appropriate columns of **D** and **Q** are deleted as well as rows and columns of **Λ**. Then **D** can be approximated by a smaller rank matrix using the same SVD formula as (5). The deletion of topics can significantly reduce the dimensionality of the problem solved while the variability of original matrix **D** is preserved as much as possible.

Additionally the matrix **PΛ** usually serves as substituent of the original matrix **D**. It can be useful in many tasks because of the ortoghonality of columns of **PΛ**. Even thou **D** and **PΛ** represent documents in different spaces (term space and topic space) they both are examples of bag-of-words coding of training documents where the order of terms or topics in the documents does not influences their representation.

3 CONTEXT WINDOW

Any document D_n is not only the unordered set of terms but it creates a sequence of terms where the order matters.

$$D_n = W_{n(1)} W_{n(2)} W_{n(3)} \cdots \tag{8}$$

Each term in the sequence implies a different distribution of expected following terms. If documents are coded as bags-of-words the valuable information about the expectation of next terms is lost. To utilize a context in a useful document representation where reduction of the dimensionality can be performed we propose to exploit the SVD product matrix **Q**. As above mentioned **QΛ** represents the vocabulary terms in the topic space. Hence each vocabulary term can be assigned by an appropriate row vector from **QΛ**. More precisely let $W_{n(k)}=W_m$ is a k^th term in document D_n and let $\mathbf{t}_{n(k)}$ is a vector of L topics assigned to this term.

$$\begin{aligned} \mathbf{t}_{n(k)} &= \left(t_{n(k)1}, t_{n(k)2}, \ldots, t_{n(k)L}\right) \\ &= \left(q_{m1}\lambda_1, q_{m2}\lambda_2, \ldots, q_{mL}\lambda_L\right) \end{aligned} \tag{9}$$

If a term $W_{n(k)}$ is not included in vocabulary *V*, vector $\mathbf{t}_{n(k)}$ does not exist. Thus let us introduce a binary function *E*(n,k) that confirms if $W_{n(k)}$ is present in the vocabulary *V*.

$$\begin{aligned} E(n,k) = 1 \Leftrightarrow W_{n(k)} \in V \\ E(n,k) = 0 \Leftrightarrow W_{n(k)} \notin V \end{aligned} \tag{10}$$

Walking through a document we can record the modification of topic weights when moving from one vocabulary term to another. The frequent changes of topic weights may represent the knowledge about structure of particular document. We suggest to use a network of topics to save the information about topic weights changes in a sequence. The network edges encode transition matrix and the vertices represent the topics.

In natural languages terms are usually related to other terms that appear in small distance. Hence we propose to take into account the changes among particular term and several next terms. We refer to these next terms as context window. We do not assume we have precise information about the relationship among terms that can be obtained from parsing thus we have to investigate all terms in the context window of a particular term .

For k^th term $W_{n(k)}$, k=1..K(n)-S, in document D_n that is present in vocabulary *V* we can define its context window $R_{n(k)}$ that consists of S subsequent terms. K(n) stands for length of document D_n.

$$R_{n(k)} = W_{n(k+1)} W_{n(k+2)} \ldots W_{n(k+S)} \tag{11}$$

Let us call $W_{n(k)}$ the pivot term and $R_{n(k)}$ a context window of $W_{n(k)}$. Like each pivot term is represented by a topic vector $\mathbf{t}_{n(k)}$ its context window can be also represented by a vector of topics $\mathbf{u}_{n(k)}$. To do so we have to combine topic vectors of all terms in the context window $R_{n(k)}$. If no term from the context window is included in vocabulary *V*, vector $\mathbf{u}_{n(k)}$ does not exist. Thus we can introduce the second binary function *F*(n,k) that confirms that at least one term from $R_{n(k)}$ is present in the vocabulary *V*.

$$\begin{aligned} F(n,k) = 1 \Leftrightarrow \exists i \in \{k+1, \ldots, k+S\} : W_{n(i)} \in V \\ F(n,k) = 0 \Leftrightarrow \forall i \in \{k+1, \ldots, k+S\} : W_{n(i)} \notin V \end{aligned} \tag{12}$$

The confirmation function *F*(n,k) can be also directly derived from confirmation function *E*(n,k).

$$\begin{aligned} F(n,k) = 1 \Leftrightarrow \sum_{i=k+1}^{k+S} E(n,i) \geq 1 \\ F(n,k) = 0 \Leftrightarrow \sum_{i=k+1}^{k+S} E(n,i) = 0 \end{aligned} \tag{13}$$

We propose two methods to derive the representation $\mathbf{u}_{n(k)}$ (if exists) for the context window $R_{n(k)}$. The first one is just simple averaging of the items of window topic vectors.

$$\mathbf{u}_{n(k)} = \frac{\sum_{i=k+1}^{k+S} \mathbf{t}_{n(i)} E(n,i)}{\sum_{i=k+1}^{k+S} E(n,i)} \tag{14}$$

The second extreme method selects the value with largest absolute value for each topic in the context window.

$$\begin{aligned} u_{n(k)l} = \max_{i \in \{k+1,k+2,\ldots,k+S\}} (t_{n(i)l}) \Leftrightarrow \left| \max_{i \in \{k+1,k+2,\ldots,k+S\}} (t_{n(i)l}) \right| \geq \left| \min_{i \in \{k+1,k+2,\ldots,k+S\}} (t_{n(i)l}) \right| \\ u_{n(k)l} = \min_{i \in \{k+1,k+2,\ldots,k+S\}} (t_{n(i)l}) \Leftrightarrow \left| \max_{i \in \{k+1,k+2,\ldots,k+S\}} (t_{n(i)l}) \right| < \left| \min_{i \in \{k+1,k+2,\ldots,k+S\}} (t_{n(i)l}) \right| \end{aligned} \tag{15}$$

4 NETWORK DOCUMENT REPRESENTATION

The relationship between each pivot term and its context window can be described by a network. The k^th pivot term $W_{n(k)}$ in document D_n and its context window $R_{n(k)}$ are represented by real vectors of topics $\mathbf{t}_{n(k)}$ and $\mathbf{u}_{n(k)}$. We propose to construct a social network $G_{n(k)}=(U, \mathbf{A}_{n(k)})$ to code a relationship between term $W_{n(k)}$ and context window $R_{n(k)}$. Let us call $G_{n(k)}$ context network. Then the document D_n

will be represented as a sequence of context networks.

$$D_n = G_{n(1)} G_{n(2)} G_{n(3)} \cdots \tag{16}$$

The set of vertices U in all $G_{n(k)}$ is set of topics derived by SVD (7). The set of weighted oriented edges represented by asymmetric LxL matrix $\mathbf{A}_{n(k)}$ describes the relationship between pairs of topics of particular pivot term $W_{n(k)}$ and its context window $R_{n(k)}$. We propose to express these relationships as products of appropriate items of column topic vectors $\mathbf{t}_{n(k)}$ and $\mathbf{u}_{n(k)}$.

$$\mathbf{A}_{n(k)} = \mathbf{t}_{n(k)} \mathbf{u}_{n(k)}^T \tag{17}$$

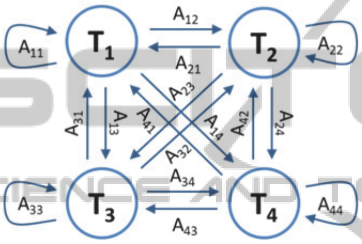


Figure 1: Context network of four topics.

The context network $G_{n(k)}$ cannot be constructed if $\mathbf{t}_{n(k)}$ or $\mathbf{u}_{n(k)}$ does not exist. To efficiently represent any document D_n we have to combine all its context networks together. We propose straightforward averaging of the context networks. Hence the document D_n will be described by a single network $H_n=(U, \mathbf{B}_n)$. The vertices are again the topics from U and the edge matrix $\mathbf{B}_n$ is the mean of all matrices $\mathbf{A}_{n(k)}$ in document D_n. Let K_n is a number of terms in document D_n. Only positions represented by a context network in document D_n contribute to the averaging.

$$\mathbf{B}_n = \frac{\sum_{k=1}^{K_n - S} \mathbf{A}_{n(k)} E(n,k) F(n,k)}{\sum_{k=1}^{K_n - S} E(n,k) F(n,k)} \tag{18}$$

5 CLASSIFICATION

Each training document D_n is labeled by at least one class Z_j, j=1..J, from the set Y of J classes.

$$Y = \{Z_1, Z_2, \ldots, Z_J\} \tag{19}$$

The task is to assign one or more classes from Y to a new document D represented by its network H=(U, **B**). The matrix **B** is constructed by the same method as for training documents utilizing the SVD of training documents and vocabulary V. We propose to compute a dissimilarity measure between network H and all training networks H_n and assign a class or classes to new document D by k-nearest neighbors method (kNN). Different classifiers could be used as well if the weighted edges of networks were considered as input features.

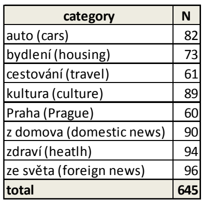
The dissimilarity of two networks of the same vertices can be measured as sum of dissimilarities of their corresponding pairs of vertices. Two vertices are similar if all their edges have similar weights. The dissimilarity of two vertices can be measured by square Euclidian distance between the vectors of edge weights (Burt, 1978). The square Euclidian distance between networks $H_1=(U, \mathbf{B}_1)$ and $H_2=(U, \mathbf{B}_2)$ is then proportional to Frobenius distance of matrices $\mathbf{B}_1$ and $\mathbf{B}_2$.

$$\begin{aligned} d^2(H_1, H_2) &= \sum_{l=1}^{L} \sum_{i=1}^{L} \left[(b_{1li} - b_{2li})^2 + (b_{1il} - b_{2il})^2 \right] \\ &= 2 \sum_{l=1}^{L} \sum_{i=1}^{L} \left[(b_{1li} - b_{2li})^2 \right] = 2 d_{Frob}(\mathbf{B}_1, \mathbf{B}_2) \end{aligned} \tag{20}$$

The first sum of above formula is sum over all vertices (topics) from V. The second sum is sum over incoming edges (the first square) and outcoming edges (the second square). Frobenius dissimilarity of two matrices of same dimensions is defined as the sum of square differences of all their items.

6 EXPERIMENTAL SETUP

We tested the proposed classification method on a collection of 645 Czech press releases. The press releases were published by Czech News Agency (ČTK) and Czech publishing company Grand Prince (GP) in July 2007. The press releases are assigned to one of eight categories: cars, housing, travel, culture, Prague, domestic news, health, foreign news. All categories are roughly equally occupied. The typical length of a document converted to plain text format is about 5kB.



category	N
auto (cars)	82
bydlení (housing)	73
cestování (travel)	61
kultura (culture)	89
Praha (Prague)	60
z domova (domestic news)	90
zdraví (heatlh)	94
ze světa (foreign news)	96
total	**645**

Figure 2: Frequencies of categories in the experimental collection.

The documents were partitioned to training and test sets randomly by ratio 70:30. Each document was parsed to words that served as terms. No advanced NLP modification of the extracted words was performed. Only words of low frequency, non-linguistic entities and words from stop-word list were filtered out. The resultant vocabulary included 5108 terms. The popular tf-idf weighting scheme (Salton & Buckley, 1988) was used to produce training document-term matrix **D**.

$$tfidf = tf \log(N / df) \tag{21}$$

The tfidf weight is proportional to term frequency tf in the document, df stands for the number of training documents where the term is present and N is the number of the training documents in the collection. The singular value decomposition was performed over training matrix **D** to obtain matrices **P**, **Q** and **Λ**. Each document from both sets initially represented in term space as row vector **d** was projected to topic space as $\mathbf{p\Lambda} = \mathbf{dQ}$. These topics weights were later used as input features for several standard classifiers that served as baseline models for comparisons.

Similarly each vocabulary term originally represented in training document space by column vector $\mathbf{d}^{\mathrm{T}}$ of matrix **D** was projected to topic space as $\mathbf{q\Lambda} = \mathbf{d}^T\mathbf{P}$. Then documents were expressed as averaged context nets of topics. Before averaging we choose proposed extreme method (15) to represented context windows. To investigate the impact of the length of context window we experimented with four context window sizes.

The SVD projection to topic space was accomplished to perform dimensionality reduction. We compare proposed kNN network classification with standard classifiers in several reduced spaces of topics. The topics with small singular values are discarded. This reduction influences the number of vertices for kNN network classification and the number of input features for standard baseline classifiers as well.

7 EXPERIMENTAL RESULTS

In the following graphs we depict the quality of classifiers measured by the absolute accuracy of recognition of all eight categories in our collection on test set. The proposed classifier is tested for four different lengths of context windows: 2, 3, 5 and 10 terms including the pivot term. Hence the first variant exploits the relations between adjacent terms only while the others look for wider contexts.

Four variants of the proposed classifier are compared with three standard algorithms that rely on topics derived by SVD. The important parameter for all comparisons is the number of topic used. We present results for 2, 3, 5, 7, 10, 15, 20, 30 and 50 topics. The number of topics implies the number of dimensions for standard methods and also the number of vertices in the network representation.

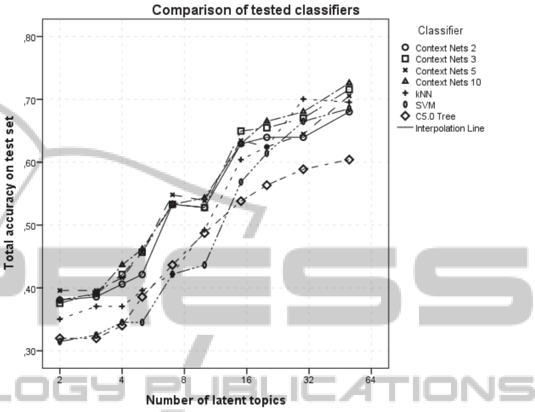


Figure 3: Comparison of proposed and standard classifiers for different number of extracted topics.

From the picture above we can conclude that dimensionality reduction influences quality of all classifiers. Note that x axis is on logarithmic scale. The quality of proposed classifiers is better than quality of other standard classifiers. The difference among variants of different context window is not so significant but we can conclude that wider context slightly improves the classification. The differences will be better recognized if we depict a relative accuracy. We choose C5.0 as a baseline classifier because of its rather smooth dependency on the number of topics.

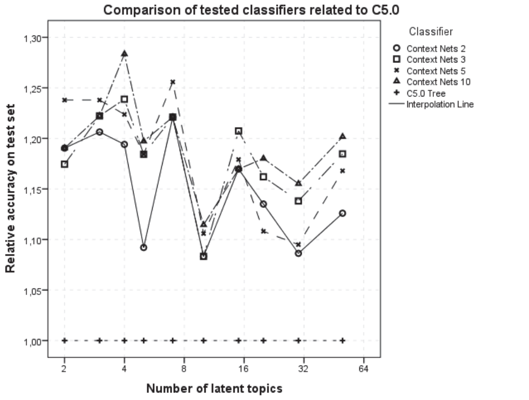


Figure 4: Normalized comparison of proposed classifiers with different lengths of context windows. The accuracy is related to accuracy of C5.0 classifier.

The graph emphases that all tested variants of proposed classifier outperformed C5.0 tree by 10%-30%. Other tested standard classifiers were outperformed as well especially when small number of topics was used. If the reduction of dimensionality is not so significant the information about the word order in documents does not improve the classification much.

8 CONCLUSIONS

We proposed network representation of text documents that contains information about sequences of tokens and enables to exploit extracted features produced by latent semantic analysis. Then we illustrated how the network representation helps to improve the accuracy of classification.

If information about context was present in input features classifiers performed considerably better especially when the dimensionality reduction was significant. We achieved improvement 10-30% in comparison with standard representation combined with kNN or C5.0 algorithms.

The size of context window does not influence the classification accuracy so considerably. We observed that larger context implies slightly better classifier. The largest context of ten tokens outperformed the shortest context of two tokens by 2% in average.

The possible modifications of proposed method include:

- tokenization of documents to n-grams instead of words before SVD and context networks are applied,
- application of different methods of construction of context topic vector **u**.

Our future work will focus to improvements of the algorithm to speed up the construction and comparison of larger context networks.

REFERENCES

Berry, P. M., Harrison, I., Lowrance, J. D., Rodriguez, A. C., & Ruspini, E. H. (2004). *Link Analysis Workbench.* Air Force Research Laboratory.

Burt, R. S. (1978). Cohesion Versus Structural Equivalence as a Basis for Network Subgroups. *Sociological Methods and Research, 7*, pp. 189-212.

Deerwester, S., Dumais, S., Furnas, G., Landauer, T., & Harshman, R. (1990). Indexing by Latent Semantic Analysis. *Journal of the American Society for Information Science, 41*, pp. 391-407.

Eibe, F., & Remco, B. (2006). Naive Bayes for Text Classification with Unbalanced Classes. *Proceedings of 10th European Conference on Principles and Practice of Knowledge Discovery in Databases* (pp. 503-510). Berlin: Springer.

Gaizauskas, R., & Wilks, Y. (1998). Information extraction: beyond document retrieval. *Journal of Documentation, 54*(1), pp. 70-105.

Han, E., Karypis, G., & Kumar, V. (2001). Text Categorization Using Weight Adjusted k-Nearest Neighbor Classification. *Proceedings of 5th Pacific-Asia Conference on Knowledge Discovery and Data Mining* (pp. 53-65). Springer-Verlag.

Kelleher, D. (2004). *Spam Filtering using Contextual Network Graphs.*

Landauer, T., Foltz, P., & Laham, D. (1998). An Introduction to Latent Semantic Analysis. *Discourse Processes, 25*, pp. 259-284.

Marin, A. (2011). Comparison of Automatic Classifiers' Performances using Word-based Feature Extraction Techniques in an E-government setting. Kungliga Tekniska Högskolan.

Salton, G., & Buckley, C. (1988). Term-weighting Approaches in Automatic Text Retrieval. *Information Processing & Management, 24*(5), pp. 513-523.

Vapnik, V. N. (1995). *The Nature of Statistical Learning Theory.* Springer-Verlag.

Wasserman, S., & Faust, K. (1994). *Social Network Analysis: Methods and Applications.* Cambridge University Press.

Weiss, S., Indurkhya, N., Zhang, T., & Damerau, F. (2005). *Text Mining.* Springer.

Yang, Y., & Pedersen, J. O. (1997). A Comparative Study on Feature Selection in Text Categorization. *Proceedings of the Fourteenth International Conference on Machine Learning* (pp. 412--420). Morgan Kaufmann Publishers.

Zhang, T., & Oles, F. J. (2000). Text Categorization Based on Regularized Linear Classification Methods. *Information Retrieval, 4*, pp. 5-31.