Computing Linear Discriminants for Idiomatic Sentence Detection

Jing Peng[1], Anna Feldman[1,2], and Laura Street[2]

[1] Department of Computer Science
[2] Department of Linguistics
Montclair State University
Montclair, NJ 07043, USA
{pengj,feldmana,streetl1}@mail.montclair.edu

Abstract. In this paper, we describe the binary classification of sentences into idiomatic and non-idiomatic. Our idiom detection algorithm is based on linear discriminant analysis (LDA). To obtain a discriminant subspace, we train our model on a small number of randomly selected idiomatic and non-idiomatic sentences. We then project both the training and the test data on the chosen subspace and use the three nearest neighbor (3NN) classifier to obtain accuracy. The proposed approach is more general than the previous algorithms for idiom detection — neither does it rely on target idiom types, lexicons, or large manually annotated corpora, nor does it limit the search space by a particular linguistic construction.

1 Introduction

Previous work on automatic idiom classification has typically been of two types: those which make use of type-based classification methods (Lin, 1999; Baldwin et al., 2002; Fazly and Stevenson, 2006; Bannard, 2007; Fazly et al., 2009) and those which make use of token-based classification methods (Birke and Sarkar, 2006; Katz and Giesbrecht, 2006; Fazly et al., 2009; Sporleder and Li, 2009). Type-based classification methods recognize idiomatic expressions (=types) to include in a lexicon and typically rely on the notion that many idioms share unique properties with one another. For instance, several idioms are composed of verb-noun constructions (e.g., *break a leg*, *get a grip*, *kick the bucket*) that cannot be altered syntactically or lexically (e.g., *break a skinny leg*, *a grip was got*, *kick the pail*). These unique properties are used to distinguish idiomatic expressions from other types of expressions in a text. Token-based classification methods recognize a particular usage (literal vs. non-literal) of a potentially idiomatic expression. Both of these approaches view idioms as multi-word expressions (MWEs) and rely crucially on preexisting lexicons or manually annotated data. They also tend to limit the search space by a particular type of linguistic construction (e.g., Verb+Noun combinations). The task of automatic idiom classification is extremely important for a variety of NLP applications; e.g., a machine translation system must translate *held fire* differently in *The army held their fire* and *The worshippers held the fire up to the idol* (Fazly et al., 2009).

Special issue: Natural Language Processing and its Applications.
Research in Computing Science 46, 2010, pp. 17-28

Received 23/11/09
Accepted 16/01/10
Final version 10/03/10

2 Our Approach

Unlike previous work on idiom detection, we view the solution to this problem as a two-step process: 1) filtering out sentences containing idioms; 2) extracting idioms from these filtered out sentences. In our current work we only consider step 1, and we frame this task as one of classification. We believe that the result of filtering out idiomatic sentences is already useful for many applications such as machine translation, information retrieval, or foreign/ second language instruction, e.g., for effective demonstrations of contexts in which specific idioms might occur

Our idiom detection algorithm is based on linear discriminant analysis (LDA). To obtain a discriminant subspace, we train our model on a small number of randomly selected idiomatic and non-idiomatic sentences. We then project both the training and the test data on the chosen subspace and use the three nearest neighbor (3NN) classifier to obtain accuracy. The proposed approach is more general than the previous algorithms for idiom detection — neither does it rely on target idiom types, lexicons, or large manually annotated corpora, nor does it limit the search space by a particular type of linguistic construction. The following sections describe the algorithm, the data and the experiments in more detail.

2.1 Idiom Detection based on Discriminant Analysis

The approach we are taking for idiomatic sentence detection is based on linear discriminant analysis (LDA) (Fukunaga, 1990). LDA often significantly simplifies tasks such as regression and classification by computing low-dimensional subspaces having statistically uncorrelated or discriminant variables. In language analysis, statistically uncorrelated or discriminant variables are extracted and utilized for description, detection, and classification. Woods et al. (1986), for example, use statistically uncorrelated variables for language test scores. A group of subjects was scored on a battery of language tests, where the subtests measured different abilities such as vocabulary, grammar or reading comprehension. Horvath (1985) analyzes speech samples of Sydney speakers to determine the relative occurrence of five different variants of each of five vowels sounds. Using this data, the speakers clustered according to such factors as gender, age, ethnicity and socio-economic class.

LDA is a class of methods used in machine learning to find the linear combination of features that best separate two classes of events. LDA is closely related to principal component analysis (PCA), where a linear combination of features that best explains the data. Discriminant analysis explicitly exploits class information in the data, while PCA does not.

Idiom detection based on discriminant analysis has several advantages. First, it does not make any assumption regarding data distributions. Many statistical detection methods assume a Gaussian distribution of normal data, which is far from reality. Second, by using a few discriminants to describe data, discriminant

analysis provides a compact representation of the data, resulting in increased computational efficiency and real time performance.

2.2 Linear Discriminant Analysis

In LDA, within-class, between-class, and mixture scatter matrices are used to formulate the criteria of class separability. Consider a J class problem, where m_0 is the mean vector of all data, and m_j is the mean vector of jth class data. A within-class scatter matrix characterizes the scatter of samples around their respective class mean vector, and it is expressed by

$$S_w = \sum_{j=1}^{J} p_j \sum_{i=1}^{l_j} (x_i^j - m_j)(x_i^j - m_j)^t, \tag{1}$$

where l_j is the size of the data in the jth class, p_j ($\sum_j p_j = 1$) represents the proportion of the jth class contribution, and t denotes the transpose operator. A between-class scatter matrix characterizes the scatter of the class means around the mixture mean m_0. It is expressed by

$$S_b = \sum_{j=1}^{J} p_j (m_j - m_0)(m_j - m_0)^T. \tag{2}$$

The mixture scatter matrix is the covariance matrix of all samples, regardless of their class assignment, and it is given by

$$S_m = \sum_{i=1}^{l} (x_i - m_0)(x_i - m_0)^T = S_w + S_b. \tag{3}$$

The Fisher criterion is used to find a projection matrix $W \in \Re^{q \times d}$ that maximizes

$$J(W) = \frac{|W^t S_b W|}{|W^t S_w W|}. \tag{4}$$

In order to determine the matrix W that maximizes $J(W)$, one can solve the generalized eigenvalue problem: $S_b w_i = \lambda_i S_w w_i$. The eigenvectors corresponding to the largest eigenvalues form the columns of W. For a two class problem, it can be written in a simpler form: $S_w w = m = m_1 - m_2$, where m_1 and m_2 are the means of the two classes. In practice, the small sample size problem is often encountered, when $l < q$. In this case S_w is singular. Therefore, the maximization problem can be difficult to solve.

2.3 Margin Criterion for Linear Dimensionality Reduction

For idiomatic sentence detection, we propose an alternative to the Fisher criterion. Here we first focus on two class problems. We note that the goal of LDA is

to find a direction w that simultaneously places two classes afar and minimizes within class variations. Fisher's criterion 4 achieves this goal. Alternatively, we can achieve this goal by maximizing

$$J(w) = tr(w^t(S_b - S_w)w), \tag{5}$$

where tr denotes the trace operator. Notice that $tr(S_b)$ measures the overall scatter of class means. Therefore, a large $tr(S_b)$ implies that the class means spread out in a transformed space. On the other hand, a small $tr(S_w)$ indicates that in the transformed space the spread of each class is small. Thus, when maximized, J indicates that data points are close to each other within a class, while they are far from each other if they come from different classes.

To see that our proposal (Eq. 5) is margin based, notice that maximizing $tr(S_b - S_w)$ is equivalent to maximizing $J = \frac{1}{2}\sum_i^2 \sum_j^2 p_i p_j d(C_i, C_j)$, where p_i denotes the probability of class C_i. The interclass distance d is defined as $d(C_i, C_j) = d(m_i, m_j) - tr(S_i) - tr(S_j)$, where m_i represents the mean of class C_i, and S_i represents the scatter matrix of class C_i. Here $d(C_i, C_j)$ measures the average margin between two classes. Therefore, maximizing our objective produces large margin linear discriminants. Large margin discriminants often result in better generalization (Vapnik, 1998). In addition, there is no need to calculate the inverse of S_w, thereby avoiding the small sample size problem associated with the Fisher criterion.

3 Computing Linear Discriminants with Semi-Definite Programming

Suppose that w optimizes (5). So does cw for any constant $c \neq 0$. Thus we require that w have unit length. The optimization problem then becomes

$$\begin{aligned} \max_w \ & tr(w^t(S_b - S_w)w) \\ \text{subject to: } & \|w\| = 1. \end{aligned}$$

This is a constraint optimization problem. Since $tr(w^t(S_b - S_w)w) = tr((S_b - S_w)ww^t) = tr((S_b - S_w)X)$, where $X = ww^t$, we can rewrite the above constraint optimization problem as

$$\begin{aligned} \max_X \ & tr((S_b - S_w)X) \\ & I \bullet X = 1 \\ & X \succeq 0 \end{aligned} \tag{6}$$

where I is the identity matrix and the inner product of symmetric matrices is $A \bullet B = \sum_{i,j}^n a_{ij} b_{ij}$, and $X \succeq 0$ means that the symmetric matrix X is positive semi-definite. Indeed, if X is a solution to the above optimization problem, then $X \succeq 0$ and $I \bullet X = 1$ implies $\|w\| = 1$, assuming $\text{rank}(X) = 1$.

The above problem is a semi-definite program (SDP), where the objective is linear with linear matrix inequality and affine equality constraints. Because linear matrix inequality constraints are convex, SDPs are convex optimization problems. The significance of SDP is due to several factors. SDP is an elegant generalization of linear programming, and inherits its duality theory. For a comprehensive overview on SDP, see (Vandenberghe and Boyd, 1996).

SDPs arise in many applications, including sparse PCA, learning kernel matrices, Euclidean embedding, and others. In general, generic methods are rarely used for solving SPDs, because their time grows at the rate of $O(n^3)$ and their memory grows in $O(n^2)$, where n is the number of rows (or columns) of a semidefinite matrix. When n is greater than a few thousands, SDPs are typically not used. However, there are algorithms that have a good theoretical foundation to solve SDPs (Vandenberghe and Boyd, 1996). In addition, semidefinite programming is a very useful technique for solving many problems. For example, SDP relaxations can be applied to clustering problems such that after solving a SDP, final clusters can be computed by projecting the data onto the space spanned by the first few eigenvectors of the SDP solution. For large-scale problems, there is a tremendous opportunity for exploiting special structures in problems, as those suggested in (Ben-Tal and Nemirovski, 2004; Nesterov, 2003).

Assume $\text{rank}(X) = 1$. Since X is symmetric, one can show that $\text{rank}(X) = 1$ iff $X = ww^t$ for some vector w. Therefore, we can recover w from X as follows. Select any column (say the ith column) of X such that $X(1,i) \neq 0$, and let

$$w = X(:,i)/X(1,i), \tag{7}$$

where $X(:,i)$ denotes the ith column of the matrix X. Thus, our goal here is to ensure the solution X to the above constraint optimization problem has rank at most 1.

One way to guarantee $\text{rank}(X) = 1$ is to use $\text{rank}(X) = 1$ as an additional constraint in the optimization problem. However, the constraint $\text{rank}(X) = 1$ is not convex and the resulting problem is difficult to solve. It turns out that the above formulation (6) is sufficient to ensure that the rank of the optimal solution X to Eq. (6) is one, i.e., $\text{rank}(X) = 1$.

Theorem 1. *Let X be the solution to the semi-definite program (6). Also, let $rank(X) = r$. Then $r = rank(X) = 1$.*

The proof of the theorem is in Appendix A. The theorem states that our procedure for computing w from the matrix X (Eq. 7) is guaranteed to produce the correct answer. We call our algorithm SDP-LDA. An attractive property associated with our algorithm is that it does not have any procedural parameters. Thus, it does not require expensive cross-validation to determine its optimal performance.

4 Dataset

In our experiments, we used the dataset described by Fazly et al. (2009). This is a dataset of verb-noun combinations extracted from the British National Cor-

pus (BNC, Burnard (2000)). The VNC tokens are annotated as either literal, idiomatic, or unknown. The list contains only those VNCs whose frequency in BNC was greater than 20 and that occurred at least in one of two idiom dictionaries (Cowie et al., 1983; Seaton and Macaulay, 2002). The dataset consists of 2,984 VNC tokens[3].

Since our task is framed as sentence classification rather than MWE extraction and filtering, we had to translate this data into our format. Basically, our dataset has to contain sentences with the following tags: I (=idiomatic sentence), L (=literal), and Q (=unknown). Translating the VNC data into our format is not trivial. A sentence that contains a VNC idiomatic construction can be unquestionably marked as I (=idiomatic); however, a sentence that contains a non-idiomatic occurrence of VNC cannot be marked as L since these sentences could have contained other types of idiomatic expressions (e.g., prepositional phrases) or even other figures of speech. So, by marking automatically all sentences that contain non-idiomatic usages of VNCs, we create an extremely noisy dataset of literal sentences. The dataset consists of 2,550 sentences, of which 2,013 are idiomatic sentences and the remaining 537 are literal sentences.

5 Experiments

We first apply the bag-of-words model to create a term-by-sentence representation of the 2,550 sentences in a 6,844 dimensional term space. The Google stop list is used to remove stop words.

We randomly choose 300 literal sentences and 300 idiomatic sentences as training and randomly choose 100 literals and 100 idioms from the remaining sentences as testing. Thus the training dataset consists of 600 examples, while the test dataset consists of 200 examples. We train our model on the training data and obtain one discriminant subspace. We then project both training and test data on the chosen subspace. Note that for the two class case (literal vs. idiom), one dimensional subspace is sufficient. In the reduced subspace, we compare three classifiers: the three nearest neighbor (3NN) classifier, the quadratics classifier that fits multivariate normal densities with covariance estimates stratified by classes (Krzanowski, 1988), and support vector machines (SVMs) with the Gaussian kernel (Cristianini and Shawe-Taylor, 2000). The kernel parameter was chosen through 10 fold cross-validation. We repeat the experiment 10 times to obtain the average accuracy rates registered by the three methods. The following table shows the accuracy rates over the ten runs.

We compare the proposed technique against a random baseline approach. The baseline approach flips a fair coin. If the outcome is head, it classifies a given sentence as idiomatic. If the outcome is tail, it classifies a given sentence as a regular sentence.

Even though we used Fazly et al. (2009)'s dataset for these experiments (see Section 4), the direct comparison with their methods is impossible here because

[3] To read more about this dataset, the reader is referred to Cook et al. (2008)

3NN	Quadratic	SVMs	Baseline
0.8015	0.7690	0.7890	0.50

Table 1. Classification accuracy rates computed by the three competing methods compared against the baseline.

our tasks are formulated differently. Fazly et al. (2009)'s unsupervised model that relies on the so-called canonical forms (CForm) gives 72.4% (macro-)accuracy on the extraction of idiomatic tokens when evaluated on their test data.

6 Analysis

To gain insights into the performance of the proposed technique, we created a dataset that is manually annotated to avoid noise in the literal dataset. We asked three human subjects to annotate 200 sentences from the VNC dataset as idiomatic, non-idiomatic or unknown. 100 of these sentences contained idiomatic expressions from the VNC data. We then merged the result of the annotation by the majority vote.

We also measured the inter-annotator agreement (the Cohen kappa k, Cohen (1960); Carletta (1996)) on the task. Interestingly, the Cohen kappa coefficient was much higher for the idiomatic data than for the so-called literal data: k (idioms) = 0.91; k (literal) = 0.66. There are several explanations of this performance. First, the idiomatic data is much more homogeneous since we selected sentences that already contained VNC idiomatic expressions. The rest of the sentences might have contained metaphors or other figures of speech and thus the judgments were more difficult to do. Second, humans easily identify idioms, but the decision whether a sentence is literal or figurative is much more challenging. The notion of "figurativeness" is not a binary property (as might be suggested by the labels that were available to the annotators). "Figurativeness" falls on a continuum from completely transparent (= literal) to entirely opaque (=figurative)[4] Third, the human annotators had to select the label, literal or idiomatic, without having access to a larger, extra-sentential context, which might have affected their judgements. Although the boundary between idiomatic and literal expressions is not entirely clear (expressions do seem to fall on a continuum in terms of idiomaticity), some expressions are clearly idiomatic and others clearly literal based on the overall agreement of our annotators. By classifying sentences as either idiomatic or literal, we believe that this additional sentential context could be used to further investigate how speakers go about making these distinctions.

[4] A similar observation is made by Cook et al. (2008) with respect to idioms.

7 Discussion

Below we provide output sentences identified by our algorithm as either idiomatic or literal.

1. True Positives (TP): Idiomatic sentences identified as idiomatic
   - *We lose our temper, feel cornered and frightened, it can be the work of an instant.*
   - *Omanis made their mark in history as early as the third century.*
2. False Positives (FP): Non-idiomatic sentences identified as idiomatic
   - *We had words of the sixties, there were words of the seventies, there were words of the eighties, words of the nineties, and we're influencing by those words, actually that's reasonably in popularity and er increasing usage, and sometime we, people actually use it and they don't know what it means.*
   - *Therefore, taking the square root of this measure we get the correlation coefficient.*
3. True Negatives (TN): Non-idiomatic sentences identified as non-idiomatic
   - *It holds up to three horses and will be driven to and from London by Mrs. Charley from their home just outside Coventry.*
   - *The referee blew a toy trumpet and Harry Payne gave the golf club a mighty hit with his bat, breaking the shaft in two.*
4. False Negatives (FN): Idiomatic sentences identified as non-idiomatic
   - *It therefore has a long-term future.*
   - *It has also been agreed that Italy will pay a reciprocal visit to Dublin in April when they will take part in a Four Nations competition to replace the Home.*

Our error analysis reveals that many cases are fuzzy and clear literal/idiomatic demarcation is difficult.

In examining our false positives (i.e., non-idiomatic expressions that were marked as idiomatic by the model), it becomes apparent that the classification of cases is not clear-cut. The expression *words of the sixties/seventies/eighties/nineties* is not idiomatic; however, it is not entirely literal either. It is metonymic – these decades could not literally produce words. Another false positive contains the expression *take the square root.* While seemingly similar to the idiom take root in *plans for the new park began to take root*, the expression *take the square root* is not idiomatic. It does not mean "to take hold like roots in soil." Like the previous false positive, we believe *take the square root* is figurative to some extent. A person cannot literally take the square root of a number like he can literally take milk out of the fridge.

When it comes to classifying expressions as idiomatic or literal, our false negatives (i.e., idiomatic expressions that were marked as non-idiomatic by the model) reveal that human judgments can be misleading. For example, *It therefore has a long-term future* was marked as idiomatic in the test corpus. While our human annotators may have thought that an object could not literally have (or hold) a long-term future, this expression does not appear to be truly idiomatic. We do not consider it to be as figurative as a true positive like *lose our temper.*

Another false negative contains a case of metonymy *Italy will pay a reciprocal visit* and the verbal phrase *take part*. In this case, our model correctly predicted that the expression is non-idiomatic. Properties of metonymy are different from those of idioms, and the verbal phrase *take part* has a meaning separate from that of the idiomatic expression *take someone's part*.

Another interesting feature that we discovered in analyzing our false negatives is that some idiomatic expressions still retain their original meanings even when other words intervene and the idioms' component words are separated and reordered. For example, in the sentence *I was little better than a criminal on whom they must keep tabs*, the prepositional phrase *on whom* is removed from the end of the idiom *keep tabs on whom* and placed in an earlier position in the sentence. Despite this permutation, the idiom still maintains its idiomatic meaning.

All of these observations support Gibbs (1984)'s claim (based on experimental evidence) that the distinctions between literal and figurative meanings have little psychological validity. He views literal and figurative expressions as end points of a single continuum. This makes the task of idiom detection even more challenging because often, perhaps, there is no objective clear boundary between idioms and literal expressions.

8 Conclusion

In this study we did not want to restrict ourselves to idioms of a particular syntactic form. We applied this method to English and used the VNC (Fazly et al., 2009) corpus for our experiments. However, in principle, the technique is language- and structure-independent.

Our binary classification approach has multiple practical applications. It is useful for indexing purposes in information retrieval (IR) as well as for increasing the precision of IR systems. Knowledge of which sentences should be interpreted literally and which figuratively can also improve text summarization and machine translation systems. Applications such as style evaluation or textual steganography detection can directly benefit from the method proposed in this paper as well. Classified sentences are useful for language instruction as well, e.g., for effective demonstrations of contexts in which specific idioms might occur. We also feel that identifying idioms at the sentence level may provide new insights into the kinds of contexts that idioms are situated in. These findings could further highlight properties that are unique to specific idioms if not idioms in general.

Our current work is concerned with improving the detection rates of our model. At present, our model does not use text coherence as a feature, and we think we could significantly improve our performance if we considered a larger context. Once the detection rates of our model have been improved, we will extract idioms from the sentences our model has classified as idiomatic. We have yet to see which method will work the best for this task.

A Proof of Theorem 1

Proof. We rewrite $S_b - S_w = 2S_b - S_m$, where $S_m = S_b + S_w$. Let $null(A)$ denote the null space of matrix A. Since $null(S_m) \subseteq null(S_b)$, there exists a matrix $P \in \Re^{q \times s}$ that simultaneously diagonalizes S_b and S_m Fukunaga (1990), where $s \leq \min\{l-1, q\}$ is the rank of S_m.

The matrix P is given by

$$P = Q\Lambda_m^{-1/2}U,$$

where Λ_m and Q are the eigenvalue and eigenvector matrices of S_m, and U is the eigenvector matrix of $\Lambda_m^{-1/2}Q^tS_bQ\Lambda_m^{-1/2}$. Thus, the columns of P are the eigenvectors of $2\lambda S_b - S_m$ and the corresponding eigenvalues are $2\Lambda_b - I$. We then have

$$P^tS_bP = \Lambda_b, \qquad P^tS_mP = I. \tag{8}$$

where $\Lambda_b = \mathrm{diag}\{\sigma_1, \cdots, \sigma_s\}$.

Consider the range of P over $Y \in \Re^{s \times q}$ with $rank(Y) = s$. The range $W = PY$ includes all $q \times q$ matrices with $rank = s$. Then

$$\begin{aligned}\max_W tr(W^t(2S_b - S_m)W) &= \max_Y tr((PY)^t(2S_b - S_m)PY) \\ &= \max_Y tr(Y^t(2\Lambda_b - I)Y).\end{aligned}$$

It is straightforward to show that the maximum is attained by $Y = [e_1e_2 \cdots e_r; 0]$, where e_i is a vector whose ith component is one and the rest is 0. From this it is clear that $W = PY$ consists of the first r columns of P, i.e., the eigenvectors corresponding to $2\lambda\sigma_i - 1 > 0$.

Now, since $X = WW^t$, we have $X = \sum_{i=1}^r w_iw_i^t$. Thus,

$$tr(X) = \sum_{i=1}^{r} w_i^tw_i = r.$$

However, the constraint $I \cdot X = 1$ states that $tr(X) = 1$. It follows that $r = 1$. That is, $rank(X) = 1$.

References

1. Baldwin, T., C. Bannard, T. Tanaka, and D. Widdows (2002). An empirical model of multiword expression decomposability. In Proceedings of the ACL 03 Workshop on Multiword expressions: analysis, acquisition and treatment, pp. 89–96.
2. Bannard, C. (2007). A measure of syntactic flexibility for automatically identifying multiword expressions in corpora. In Proceedings of the ACL 07 Workshop on A Broader Perspective on Multiword Expressions, pp. 1–8.
3. Ben-Tal, A. and A. Nemirovski (2004). Non-euclidean restricted memory level method for large-scale convex optimization.
4. Birke, J. and A. Sarkar (2006). A clustering approach to the nearly unsuper- vised recognition of nonliteral language. In Proceedings of the 11th Confer- ence of the European Chapter of the Association for Computational Linguistics (EACL'06), Trento, Italy, pp. 329–226.
5. Burnard, L. (2000). The British National Corpus Users Reference Guide. Oxford University Computing Services.
6. Carletta, J. (1996). Assessing Agreement on Classification Tasks: The Kappa Statistic. Computational Linguistics 22(2), 249–254.
7. Cohen, J. (1960). A Coefficient of Agreement for Nominal Scales. Education and Psychological Measurement (20), 37–46.
8. Cook, P., A. Fazly, and S. Stevenson (2008, June). The VNC-Tokens Dataset. In Proceedings of the LREC Workshop: Towards a Shared Task for Multiword Expressions (MWE 2008), Marrakech, Morocco.
9. Cowie, A. P., R. Mackin, and I. R. McCaig (1983). Oxford Dictionary of Current Idiomatic English, Volume 2. Oxford University Press.
10. Cristianini, N. and J. Shawe-Taylor (2000). An Introduction to Support Vector Machines and other kernel-based learning methods. Cambridge, UK: Cam- bridge University Press.
11. Fazly, A., P. Cook, and S. Stevenson (2009). Unsupervised Type and Token Identification of Idiomatic Expressions. Computational Linguistics 35 (1), 61–103.
12. Fazly, A. and S. Stevenson (2006). Automatically Constructing a Lexicon of Verb Phrase Idiomatic Combinations. In Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL-2006), Trento, Italy, pp. 337–344.
13. Fukunaga, K. (1990). Introduction to statistical pattern recognition. Academic Press.
14. Gibbs, R. W. (1984). Literal Meaning and Psychological Theory. Cognitive Science 8, 275–304.
15. Horvath, B. M. (1985). Variation in Australian English. Cambridge: Cambridge University PRess.
16. Katz, G. and E. Giesbrecht (2006). Automatic identification of non- compositional multi-word expressions using latent semantic analysis. In Proceedings of the ACL'06 Workshop on Multiword Expressions: Identifying and Exploiting Underlying Properties, Sydney, Australia, pp. 12–19.
17. Krzanowski, W. (1988). Principles of Multivariate Analysis. UK: Oxford

University Press.
18. Lin, D. (1999). Automatic Identification of Non-compositional Phrases. In Proceedings of ACL, College Park, Maryland, pp. 317–324.
19. Nesterov, I. (2003). Smooth minimization of non-smooth functions.
20. Seaton, M. and A. Macaulay (Eds.) (2002). Collins COBUILD Idioms Dictionary(second ed.). HarperCollins Publishers.
21. Sporleder, C. and L. Li (2009). Lexical Encoding of MWEs. In Proceedings of EACL 2009.
22. Vandenberghe, L. and S. Boyd (1996). Semidefinite programming. SIAM Review 38(1), 49–95.
23. Vapnik, V. (1998). Statistical Learning Theory. New York: Wiley.
24. Woods, A., P. Fletcher, and A. Hughes (1986). Statistics in Language Studies. Cambridge: Cambridge University Press.