Automatic Detection of Idiomatic Clauses

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

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

Abstract. We describe several experiments whose goal is to automatically identify idiomatic expressions in written text. We explore two approaches for the task: 1) idiom recognition as outlier detection; and 2) supervised classification of sentences. We apply principal component analysis for outlier detection. Detecting idioms as lexical outliers does not exploit class label information. So, in the following experiments, we use linear discriminant analysis to obtain a discriminant subspace and later use the three nearest neighbor classifier to obtain accuracy. We discuss pros and cons of each approach. All the approaches are more general than the previous algorithms for idiom detection – neither do they rely on target idiom types, lexicons, or large manually annotated corpora, nor do they limit the search space by a particular type of linguistic construction.

1 Introduction

Idioms are conventionalized expressions that have figurative meanings that cannot be derived from the literal meaning of the phrase. The prototypical examples of idioms are expressions like *I'll eat my hat*, *He put his foot in his mouth*, *Cut it out*, *I'm going to rake him over the coals*, *a blessing in disguise*, *a chip on your shoulder*, or *kick the bucket*. Researchers have not come up with a single agreed-upon definition of idioms that covers all members of this class (Glucksberg, 1993; Cacciari, 1993; Nunberg et al., 1994; Sag et al., 2002; Villavicencio et al., 2004; Fellbaum et al., 2006). The common property ascribed to the idiom is that it is an expression whose meaning is different from its simple compositional meaning.

Some idioms become frozen in usage, and they resist change in syntactic structure, while others do allow some variability in expression (Fellbaum, 2007; Fazly et al., 2009). In addition, Fazly et al. (2009) have argued that in many situations, a Natural Language Processing (NLP) system will need to distinguish a usage of a potentially-idiomatic expression as either idiomatic or literal in order to handle a given sequence of words appropriately. We discuss previous approaches to automatic idiom detection in section 3.1.

2 Our Approach

Following Degand and Bestgen (2003), we have identified three important properties of idioms. (1) A sequence with literal meaning has many neighbors, whereas

a figurative one has few. (2) Idiomatic expressions should demonstrate low semantic proximity between the words composing them. (3) Idiomatic expressions should demonstrate low semantic proximity between the expression and the preceding and subsequent segments.

Based on the properties of idioms outlined above, we have experimented with two ideas: (1) The problem of idiom recognition be reduced to the problem of identifying a *semantic outlier*. By an *outlier* we mean an observation which appears to be inconsistent with the remainder of a set of data. We apply principal component analysis (PCA) (Jolliffe, 1986; Shyu et al., 2003) for outlier detection. (2) We view the process of idiom detection as a binary classification of sentences (idiomatic vs literal sentences). We use linear discriminant analysis (LDA) (Fukunaga, 1990) to obtain a discriminant subspace and later use the three nearest neighbor classifier to obtain accuracy. We first provide a few words on previous work.

3 Related Work

3.1 Automatic idiom detection

Previous approaches to automatic idiom detection can be classified into two major groups: 1) Type-based extraction, i.e., detecting idioms at the type level; 2) token-based detection, i.e., detecting idioms in context.

Type-based extraction is based on the idea that idiomatic expressions exhibit certain linguistic properties that can distinguish them from literal expressions. Sag et al. (2002); Fazly et al. (2009), among many others, discuss various properties of idioms. Some examples of such properties include 1) lexical fixedness: e.g., neither 'shoot the wind' nor 'hit the breeze' are valid variations of the idiom *shoot the breeze*. and 2) syntactic fixedness: e.g., *The guy kicked the bucket* is idiomatic whereas *The bucket was kicked* is not idiomatic anymore; and of course, 3) non-compositionality. While many idioms do have these properties, many idioms fall on the continuum from being compositional to being partly unanalyzable to completely non-compositional (Cook et al., 2008). Fazly et al. (2009); Li and Sporleder (2010), among others, notice that type-based approaches do not work on expressions that can be interpreted idiomatically or literally depending on the context and thus, an approach that considers tokens in context is more appropriate for the task of idiom recognition. A number of token-based approaches has been discussed in the literature, both supervised (Katz and Giesbrecht, 2006), weakly supervised (Birke and Sarkar, 2006) and unsupervised (Fazly et al., 2009; Sporleder and Li, 2009).

Li and Sporleder (2009); Sporleder and Li (2009) propose a graph-based model for representing the lexical cohesion of a discourse. Nodes correspond to tokens in the discourse, which are connected by edges whose value is determined by a semantic relatedness function. They experiment with two different approaches to semantic relatedness: 1) Dependency vectors, as described in Pado and Lapata (2007); 2) Normalized Google Distance (NGD) (Cilibrasi and

Vitányi, 2007). Li and Sporleder (2009) show that this method works better for larger contexts (greater than five paragraphs).

Li and Sporleder (2010) assume that literal and figurative data are generated by two different Gaussians, literal and non-literal, and the detection is done by comparing which Gaussian model has a higher probability to generate a specific instance. The approach assumes that the target expressions are already known and the goal is to determine whether this expression is literal or figurative in a particular context. The important insight of this method is that figurative language in general exhibits fewer semantic cohesive ties with the context than literal language. Their results are inconclusive, unfortunately, due to the small size of the test corpus. In addition, it relies heavily on target expressions. One has to have a list of potential idioms a priori to model semantic relatedness.

4 Idea 1: Idioms as outliers

We first explore the hypothesis that the problem of automatic idiom detection can be formulated as the problem of identifying an outlier in a dataset. We investigate principal component analysis (PCA) (Jolliffe, 1986; Shyu et al., 2003) for outlier detection.

4.1 Idiom Detection Based on Principal Component Analysis

The approach we are taking for idiom detection is based on principal component analysis (PCA) (Jolliffe, 1986; Shyu et al., 2003). PCA has several advantages in outlier detection. First, it does not make any assumptions regarding data distributions. Many statistical detection methods assume a Gaussian distribution of normal data, which is far from reality. Second, by using a few principal modes to describe data, PCA provides a compact representation of the data, resulting in increased computational efficiency and real time performance.

PCA computes a set of mathematical features, called principal components, to explain the variance in the data. These principal components are linear combinations of the original variables describing the data and are orthogonal to each other. The first principal component corresponds to the direction along which the data vary the most. The second principal component corresponds to the direction along which the data vary the second most, and so on. Furthermore, total variance in all the principal components explains total variance in the data.

Let $\mathbf{z} = \{\mathbf{x}_i\}_{i=1}^m$ be a set of data points. Each $\mathbf{x}_i = (x_i^1, \cdots, x_i^q)^t$, where t denotes the transpose operator. That is, each data point is described by q attributes or variables. PCA computes a set of eigenvalue and eigenvector pairs $\{(\lambda_1, e_1, \cdots, (\lambda_q, e_q)\}$ with $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_q$ by performing singular value decomposition of the covariance matrix of the data: $\Sigma = \sum_{i=1}^m (\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^t$, where $\bar{x} = 1/m \sum_{i=1}^m$. Then the ith principal component of an observation $\mathbf{x}$ is given by $y_i = e_i^t(\mathbf{x} - \bar{\mathbf{x}})$.

Note that the major components correspond strongly to the attributes having relatively large variance and covariance. Consequently, after projecting the data

onto the principal component space, idioms that are outliers with respect to the major components usually correspond to outliers on one or more of the original attributes. On the other hand, minor (last few) components represent a linear combination of the original attributes with minimal variance. Thus, the minor components are sensitive to observations that are inconsistent with the variance structure of the data but are not considered to be outliers with respect to the original attributes (Jobson, 1992). Therefore, a large value along the minor components strongly indicates a potential outlier that otherwise may not be detected based solely on large values of the original attributes.

Our technique (we call it "principal minor component analysis": PMC) computes two functions for a given input $\mathbf{x}$. The first one is computed along major components: $f(x) = \sum_{i=1}^{p} \frac{y_i^2}{\lambda_i}$. The second one is computed along minor components: $g(x) = \sum_{i=q-r+1}^{q} \frac{y_i^2}{\lambda_i}$. Here y_i are projections along each component. p represents the number of major components and captures sufficient variance in the data, while r denotes the number of minor components. Both p and r can be determined through cross-validation.

It can be seen from our earlier discussion that $f(\mathbf{x})$ captures extreme observations with large values along some original attributes. On the other hand, $g(\mathbf{x})$ measures observations that are outside of the normal variance structure in the data, as measured by minor components. Thus, the strength of our approach is that it detects an outlier that is either extremely valued along the major components, or does not confirm to the same variance structure along the minor components in the data.

Our technique then decides an input $\mathbf{x}$ as outlier if $f(\mathbf{x}) \geq T_f$ or $g(\mathbf{x}) \geq T_g$, where T_f and T_g are outlier thresholds that are associated with the false positive rate α (Kendall et al., 2009). Suppose that the data follow the normal distribution. Define $\alpha_f = \Pr\{\sum_{i=1}^{p} \frac{y_i^2}{\lambda_i} > T_f | \mathbf{x} \text{ is normal}\}$, and $\alpha_g = \Pr\{\sum_{i=q-r+1}^{q} \frac{y_i^2}{\lambda_i} > T_g | \mathbf{x} \text{ is normal}\}$. Then $\alpha = \alpha_f + \alpha_g - \alpha_f \alpha_g$. The false positive rate has the following bound (Kendall et al., 2009) $\alpha_f + \alpha_g - \sqrt{\alpha_f \alpha_g} \leq \alpha \leq \alpha_f + \alpha_g$. Different types of outliers can be detected based on the values of α_f and α_g. If $\alpha_f = \alpha_g$, α can be determined by solving a simple quadratic equation. For example, if we want a 2% false positive rate (i.e., $\alpha = 0.02$), we obtain $\alpha_f = \alpha_g = 0.0101$.

Note that the above calculation is based on the assumption that our data follow the normal distribution. This assumption however is unlikely to be true in practice. We therefore determine α_f and α_g values based on the empirical distributions of $\sum_{i=1}^{p} y_i^2/\lambda_i$ and $\sum_{i=q-r+1}^{q} y_i^2/\lambda_i$ in the training data. That is, for a false positive rate of 2%, T_f and T_g represent the 0.9899 quantile of the empirical distributions of $\sum_{i=1}^{p} y_i^2/\lambda_i$ and $\sum_{i=q-r+1}^{q} y_i^2/\lambda_i$, respectively.

5 Idea 2: Supervised classification for idiom detection

Another way to look at the problem of idiom detection is one of the supervised classification.

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 ran-

domly 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.

5.1 Idiom Detection based on Discriminant Analysis

A similar approach has been discussed in Peng et al. (2010). 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) that concentrates on finding 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, as previously mentioned, it does not make any assumptions 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.

6 Datasets

6.1 Dataset 1: Outlier Detection

For the outlier detection, it is important that the training corpus is free of any idiomatic or metaphoric expressions. Otherwise, idiomatic or metaphoric expressions as outliers can distort the variance-covariance structure of the semantics of the corpus.

Our training set consists of 1,200 sentences (22,028 tokens) randomly extracted from the British National Corpus (BNC, http://www.natcorp.ox.ac.uk/). The first half of the data comes from the social science domain [3] and the other is defined in BNC as "imaginative" [4]. Our annotators were asked to identify clauses containing (any kind of) metaphors and idioms and paraphrase them literally. We used this paraphrased corpus for training. [5] The training data contains 139 paraphrased sentences.

[3] This is an excerpt from *Cities and Plans: The Shaping of Urban Britain in the Nineteenth and Twentieth Centuries* by Gordon Emanuel Cherry.

[4] This is an excerpt taken from *Heathen*, a thriller novel written by Shaun Hutson.

[5] We understand that this task is highly subjective, but the inter-annotator agreement was relatively high for this task (Cohen's kappa: 75%)

Our test data are 99 sentences extracted from the BNC social science (non-fiction) section, annotated as either literal or figurative and additionally labeled with the information about the figures of speech they contain (idioms (I), dead metaphors (DM), and living metaphors (LM)). The annotator has identified 12 idioms, 22 dead metaphors, and 2 living metaphors in that text.

6.2 Dataset 2: LDA

In the LDA experiments, we used the dataset described by Fazly et al. (2009). This is a dataset of verb-noun combinations extracted from the British National Corpus (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[6].

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 automatically marking 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.

7 Experiments

7.1 Detecting Outliers

We compare the proposed technique (PMC) against a random baseline approach. The baseline approach flips a fair coin. If the outcome is head, it classifies a given sentence as outlier (idiom, dead metaphor or living metaphor). If the outcome is tail, it classifies a given sentence as a regular sentence. The outlier thresholds T_f and T_g at a given false positive rate are determined from the training data by setting $\alpha_f = \alpha_g$.

In this experiment, we treat each sentence as a document. We created a bag-of-word model for the data set, i.e., we use TF-IDF to represent the data. Single value decomposition is then applied to the bag of words and the number of principal modes for representing the latent semantics space is calculated that capture 100% variance in the data.

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

Experimental Results: PCA The following table shows the detection rates of the two competing methods at a given false positive rate. The results reported here were based on 10% of major components (p in fuction f) and 0.1% minor components (r in function g). It turns out that the technique is not sensitive to p values, while r represents a trade-off between detection and precision.

False Positive Rate	PMC					Baseline
1%	47%	44%	45%	43%	100%	50%
2%	53%	44%	55%	52%	100%	50%
4%	63%	56%	63%	62%	100%	50%
6%	70%	67%	73%	71%	100%	50%
8%	73%	77%	73%	71%	100%	50%
10%	87%	89%	86%	86%	100%	50%

Table 1. The detection rates of the two competing methods at a given false positive rate: Second column: idioms and metaphors; Third column: idioms only; Fourth column: metaphors (dead and living); Fifth column: dead metaphors only; and Sixth column: living metaphors only.

7.2 LDA and supervised classification

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, 2000), 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 20 times to obtain the average accuracy rates registered by the three methods. We also include the performance by a random baseline approach. The baseline approach (BL) 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. The results are the following–*3NN*: 0.802, *Quadratic*: 0.769, *SVMs*: 0.789, and *BL*: 0.50.

Table 7.2 shows the precision, recall and accuracy for the nearest neighbor performance as a function of false positive rates. For the nearest neighbor

method, the false positive rates are achieved by varying the number of nearest neighbors in predicting idioms. By varying the number of nearest neighbors from 1 to 39, the false positive rates from 15% to 20% are obtained.

False Positive Rate	recall	precision	accuracy
15%	76%	83%	81%
16%	77%	83%	81%
17%	77%	82%	80%
20%	76%	79%	78%

Table 2. LDA performance in more detail

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

8 Comparison of the two methods

Unfortunately, the direct comparison of the two methods discussed above is not possible because they are not using the same datasets. To make the comparison fair, we decided to run the PMC outlier detection algorithm on the data used for LDA. In section 6.2 we already described how the data was created. Unlike the dataset used for the PCA outlier detection algorithm, where human annotators were given the task to make the training data as literal as possible and eliminate any possible figurative expressions, the data used by the LDA alogorithm was noisy. It was crudely translated by labeling all sentences that contained a VNC idiom from Cook et al. (2008) as idiomatic; the rest were labeled as literal. Unfortunately, this approach creates an extremely noisy dataset of literal sentences that can potentially contain other types of figurative expressions, other types of idioms etc. Another observation that our human annotators made when preparing the dataset of paraphrases used by PMC is that it was really difficult to avoid figurative language when paraphrasing and that it was also difficult to notice figurative expressions because some of them had become conventional.

So, having taken the issues discussed above into consideration, we decided to reverse our PMC approach and hypothesize that most sentences contain some figurative language and true outliers are in fact the literal sentences. So, we train the PMC algorithm on 1,800 sentences containing idioms (used by the LDA approach) rather than on literals. We test PMC on randomly selected 150 idiomatic and 150 literal sentences. We repeat the experiment 20 times to obtain the average accuracy rates. The numbers are reported in Table 8.

The performance of the LDA approach is better than that of PMC. However, the LDA method is supervised and requires training data, i.e., sentences marked

False Positive Rate	PMC recall	PMC precision	PMC accuracy
1%	93.3%	51.3%	52.3%
2%	91.9%	51.5%	52.6%
4%	91.0%	51.8%	53.2%
6%	89.5%	52.1%	53.6%
8%	88.1%	52.1%	53.6%
10%	81.2%	52.8%	54.3%

Table 3. Performance of the PMC algorithm on the LDA dataset

as idiomatic or literal, while PMC does not use class label information. However, PMC's detection rates (recall) are higher than the detection rates of the LDA algorithm. Depending on an application, the higher recall might be preferred. For example, if a researcher looks for sentences containing figurative speech for the purpose of linguistic analysis, after some postediting, s/he might wind up with more interesting examples.

9 Qualitative analysis of the results

Our error analysis of the two methods 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 s/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*.

9.1 Inter-annotator agreement

To gain insights into the performance of the second approach, 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 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 it was more difficult to make the judgments. 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)[7] 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.

10 Conclusion

The binary classification approach, offered in this paper, has multiple practical applications. 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, by incorporating textual cohesion and compositionality measures into our models.

Acknowledgements

This material is based in part upon work supported by the National Science Foundation under Grant Numbers 0916280, 1033275, and 1048406. The findings and opinions expressed in this material are those of the authors and do not reflect the views of the NSF.

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

Bibliography

Birke, J. and A. Sarkar (2006). A clustering approach to the nearly unsupervised recognition of nonliteral language. In *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL'06)*, Trento, Italy, pp. 329–226.

Burnard, L. (2000). *The British National Corpus Users Reference Guide*. Oxford University Computing Services.

Cacciari, C. (1993). The Place of Idioms in a Literal and Metaphorical World. In C. Cacciari and P. Tabossi (Eds.), *Idioms: Processing, Structure, and Interpretation*, pp. 27–53. Lawrence Erlbaum Associates.

Carletta, J. (1996). Assessing Agreement on Classification Tasks: The Kappa Statistic. *Computational Linguistics 22*(2), 249–254.

Cilibrasi, R. and P. M. B. Vitányi (2007). The google similarity distance. *IEEE Trans. Knowl. Data Eng. 19*(3), 370–383.

Cohen, J. (1960). A Coefficient of Agreement for Nominal Scales. *Education and Psychological Measurement* (20), 37–46.

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.

Cowie, A. P., R. Mackin, and I. R. McCaig (1983). *Oxford Dictionary of Current Idiomatic English*, Volume 2. Oxford University Press.

Cristianini, N. and J. Shawe-Taylor (2000). *An Introduction to Support Vector Machines and other kernel-based learning methods*. Cambridge, UK: Cambridge University Press.

Degand, L. and Y. Bestgen (2003). Towards Automatic Retrieval of Idioms in French Newspaper Corpora. *Literary and Linguistic Computing 18*(3), 249–259.

Fazly, A., P. Cook, and S. Stevenson (2009). Unsupervised Type and Token Identification of Idiomatic Expressions. *Computational Linguistics 35*(1), 61–103.

Fellbaum, C. (2007). The Ontological Loneliness of Idioms. In A. Schalley and D. Zaefferer (Eds.), *Ontolinguistics*. Mouton de Gruyter.

Fellbaum, C., A. Geyken, A. Herold, F. Koerner, and G. Neumann (2006). Corpus-based Studies of German Idioms and Light Verbs. *International Journal of Lexicography 19*(4), 349–360.

Fukunaga, K. (1990). *Introduction to statistical pattern recognition*. Academic Press.

Glucksberg, S. (1993). Idiom Meanings and Allusional Content. In C. Cacciari and P. Tabossi (Eds.), *Idioms: Processing, Structure, and Interpretation*, pp. 3–26. Lawrence Erlbaum Associates.

Jobson, J. (1992). *Applied Multivariate Data Analysis, Volume II: Categorical and Multivariate Methods*. Springer Verlag.

Jolliffe, I. (1986). *Principal Component Analysis*. New York: Springer-Verlag.

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.

Kendall, M., A. Stuart, and J. Ord (2009). *Kendall's Advanced Theory of Statistics: Volume 1:Distribution Theory*. John Wiley and Sons.

Krzanowski, W. J. (2000). *Principles of Multivariate Analysis*. Oxford University Press.

Li, L. and C. Sporleder (2009). A Cohesion Graph Based Approach for Unsupervised Recognition of Literal and Non-literal Use of Multiword Expresssions. In *Proceedings of the 2009 Workshop on Graph-based Methods for Natural Language Processing (ACL-IJCNLP)*, Singapore, pp. 75–83.

Li, L. and C. Sporleder (2010). Using Gaussian Mixture Models to Detect Figurative Language in Context. In *Proceedings of NAACL/HLT 2010*.

Nunberg, G., I. A. Sag, and T. Wasow (1994). Idioms. *Language 70*(3), 491–538.

Pado, S. and M. Lapata (2007). Dependency-based construction of semantic space models. *Computational Linguistics 33*(2), 161–199.

Peng, J., A. Feldman, and L. Street (2010). Computing linear discriminants for idiomatic sentence detection. *Research in Computing Science, Special issue: Natural Language Processing and its Applications 46*, 17–28.

Sag, I. A., T. Baldwin, F. Bond, A. Copestake, and D. Flickinger (2002). Multiword expressions: A Pain in the Neck for NLP. In *Proceedings of the 3rd International Conference on Intelligence Text Processing and Computational Linguistics (CICLing 2002)*, Mexico City, Mexico, pp. 1–15.

Seaton, M. and A. Macaulay (Eds.) (2002). *Collins COBUILD Idioms Dictionary* (second ed.). HarperCollins Publishers.

Shyu, M., S. Chen, K. Sarinnapakorn, and L. Chang (2003). A novel anomaly detection scheme based on principal component classifier. In *Proceedings of IEEE International Conference on Data Mining*.

Sporleder, C. and L. Li (2009). Unsupervised Recognition of Literal and Non-literal Use of Idiomatic Expressions. In *EACL '09: Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, Morristown, NJ, USA, pp. 754–762. Association for Computational Linguistics.

Villavicencio, A., A. Copestake, B. Waldron, and F. Lambeau (2004). Lexical Encoding of MWEs. In *Proceedings of the Second ACL Workshop on Multiword Expressions: Integrating Processing*, Barcelona, Spain, pp. 80–87.