Automatic Idiom Recognition with Word Embeddings

Jing Peng(✉) and Anna Feldman

Department of Computer Science and Department of Linguistics,
Montclair State University, Montclair, NJ 07043, USA
pengj@mail.montclair.edu

Abstract. Expressions, such as *add fuel to the fire*, can be interpreted literally or idiomatically depending on the context they occur in. Many Natural Language Processing applications could improve their performance if idiom recognition were improved. Our approach is based on the idea that idioms and their literal counterparts do not appear in the same contexts. We propose two approaches: (1) Compute inner product of context word vectors with the vector representing a target expression. Since literal vectors predict well local contexts, their inner product with contexts should be larger than idiomatic ones, thereby telling apart literals from idioms; and (2) Compute literal and idiomatic scatter (covariance) matrices from local contexts in word vector space. Since the scatter matrices represent context distributions, we can then measure the difference between the distributions using the Frobenius norm. For comparison, we implement [8,16,24] and apply them to our data. We provide experimental results validating the proposed techniques.

Keywords: Idiom recognition · Vector space models · Distributional semantics · Word embeddings

1 Introduction

Natural language is filled with emotion and implied intent, which are often not trivial to detect. One specific challenge are idioms. Figurative language draws off of prior references and is unique to each culture and sometimes what we don't say is even more important than what we do. This, naturally, presents a significant problem for many Natural Language Processing (NLP) applications as well as for big data analytics.

Idioms are conventionalized expressions whose figurative meanings cannot be derived from literal meaning of the phrase. There is no single agreed-upon definition of idioms that covers all members of this class [3,10,13,19,22,25]. At the same time, idioms do not form a homogeneous class that can be easily defined. Some examples of idioms are *I'll eat my hat* (I'm confident), *Cut it out* (Stop talking/doing something), *a blessing in disguise* (some bad luck or misfortune results in something positive), *kick the bucket* (die), *ring a bell* (sound

J.A. Lossio-Ventura and H. Alatrista-Salas (Eds.): SIMBig 2015/2016, CCIS 656, pp. 17–29, 2017.
DOI: 10.1007/978-3-319-55209-5_2

familiar), *keep your chin up* (remain cheerful), *piece of cake* (easy task), *miss the boat* (miss out on something), *(to be) on the ball* (be attentive/competent), *put one's foot in one's mouth* (say something one regrets), *rake someone over the coals* (to reprimand someone severely), *under the weather* (sick), *a hot potato* (controversial issue), *an arm and a leg* (expensive), *at the drop of a hat* (without any hesitation), *barking up the wrong tree* (looking in the wrong place), *beat around the bush* (avoiding main topic).

It turns out that expressions are often ambiguous between an idiomatic and a literal interpretation, as one can see in the examples below[1]:

(A) After the last page was sent to the printer, an editor would **ring a bell**, walk toward the door, and holler "Good night!" (Literal) (B) His name never fails to **ring a bell** among local voters. Nearly 40 years ago, Carthan was elected mayor of Tchula... (Idiomatic)

(C) ... that caused the reactor to literally **blow its top**. About 50 tons of nuclear fuel evaporated in the explosion... (Literal) (D) ... He didn't pound the table, he didn't **blow his top**. He always kept his composure. (Idiomatic)

(E) ... coming out of the fourth turn, slid down the track, **hit** the inside **wall** and then hit the attenuator at the start of pit road. (Literal) (F) ... job training, research and more have **hit** a Republican **wall**. (Idiomatic)

[8] analysis of 60 idioms from the British National Corpus (BNC) has shown that close to half of these also have a clear literal meaning; and of those with a literal meaning, on average around 40% of their usages are literal. Therefore, idioms present great challenges for many Natural Language Processing (NLP) applications. Most current translation systems rely on large repositories of idioms. Unfortunately, more frequently than not, MT systems are not able to translate idiomatic expressions correctly.

In this paper we describe an algorithm for automatic classification of idiomatic and literal expressions. Similarly to [21], we treat idioms as semantic outliers. Our assumption is that the context word distribution for a literal expression will be different from the distribution for an idiomatic one. We capture the distribution in terms of covariance matrix in vector space.

2 Our Approach

Our idea is simple: idiomatic expressions and their literal counterparts do not occur in the same contexts. We formulate two hypotheses.

1. Projection Based on Local Context Representation
   We hypothesize that words in a given text segment that are representatives of the local context are likely to associate strongly with a literal expression in the segment, in terms of projection (or inner product) of word vectors onto the vector representing the literal expression.

[1] These examples are extracted from the Corpus of Contemporary American English (COCA) (http://corpus.byu.edu/coca/).

2. Local Context Distributions
   We hypothesize that the context word distribution for a literal expression in word vector space will be different from the distribution for an idiomatic one. This hypothesis also underlies the distributional approach to meaning [11,15].

We want to emphasize that our approach is applicable to any type of syntactic constructions, but the experiments described below are based on the data released by [8], i.e., verb-noun constructions (VNCs). Thus, we can directly compare the performance of our model to [8] work.

2.1 Projection Based on Local Context Representation

The local context of a literal target verb-noun construction (VNC) must be different from that of an idiomatic one. We propose to exploit recent advances in vector space representation to capture the difference between local contexts [17,18].

A word can be represented by a vector of fixed dimensionality q that best predicts its surrounding words in a sentence or a document [17,18]. Given such a vector representation, our first proposal is the following. Let v and n be the vectors corresponding to the verb and noun in a target verb-noun construction, as in *blow whistle*, where $v \in \Re^q$ represents *blow* and $n \in \Re^q$ represents *whistle*. Let $\sigma_{vn} = v + n \in \Re^q$. Thus, σ_{vn} is the word vector that represents the composition of verb v and noun n, and in our example, the composition of *blow* and *whistle*. As indicated in [18], word vectors obtained from deep learning neural net models exhibit linguistic regularities, such as additive compositionality. Therefore, σ_{vn} is justified to predict surrounding words of the composition of, say, *blow* and *whistle*. Our hypothesis is that on average, inner product $\sigma_{blowwhistle} \cdot v$, where vs are context words in a literal usage, should be greater than $\sigma_{blowwhistle} \cdot v$, where vs are context words in an idiomatic usage.

For a given vocabulary of m words, represented by matrix

$$V = [v_1, v_2, \cdots, v_m] \in \Re^{q \times m}, \tag{1}$$

we calculate the projection of each word v_i in the vocabulary onto σ_{vn}

$$P = V^t \sigma_{vn} \tag{2}$$

where $P \in \Re^m$, and t represents transpose. Here we assume that σ_{vn} is normalized to have unit length. Thus, $P_i = v_i^t \sigma_{vn}$ indicates how strongly word vector v_i is associated with σ_{vn}. This projection, or inner product, forms the basis for our proposed technique.

Let

$$D = \{d_1, d_2, \cdots, d_l\}$$

be a set of l text segments (local contexts), each containing a target VNC (i.e., σ_{vn}). Instead of generating a term by document matrix, where each term is

tf-idf (product of term frequency and inverse document frequency), we compute a term by document matrix $M_D \in \Re^{m \times l}$, where each term in the matrix is

$$p \cdot idf, \tag{3}$$

the product of the projection of a word onto a target VNC and inverse document frequency. That is, the term frequency (tf) of a word is replaced by the projection (inner product) of the word onto σ_{vn} (2). Note that if segment d_j does not contain word v_i, $M_D(i,j) = 0$, which is similar to *tf-idf* estimation. The motivation is that topical words are more likely to be well predicted by a literal VNC than by an idiomatic one. The assumption is that a word vector is learned in such a way that it best predicts its surrounding words in a sentence or a document [17,18]. As a result, the words associated with a literal target will have larger projection onto a target σ_{vn}. On the other hand, the projections of words associated with an idiomatic target VNC onto σ_{vn} should have a smaller value.

We also propose a variant of $p \cdot idf$ representation. In this representation, each term is a product of p and typical *tf-idf*. That is,

$$p \cdot tf \cdot idf. \tag{4}$$

2.2 Local Context Distributions

Our second hypothesis states that words in a local context of a literal expression will have a different distribution from those in the context of an idiomatic one. We propose to capture local context distributions in terms of scatter matrices in a space spanned by word vectors [17,18].

Let $d = (w_1, w_2 \cdots, w_k) \in \Re^{q \times k}$ be a segment (document) of k words, where $w_i \in \Re^q$ are represented by a vectors [17,18]. Assuming w_is have been centered, we compute the scatter matrix

$$\Sigma = d^t d, \tag{5}$$

where Σ represents the local context distribution for a given target VNC.

Given two distributions represented by two scatter matrices Σ_1 and Σ_2, a number of measures can be used to compute the distance between Σ_1 and Σ_2, such as Choernoff and Bhattacharyya distances [12]. Both measures require the knowledge of matrix determinant. In our case, this can be problematic, because Σ (5) is most likely to be singular, which would result in a determinant to be zero.

We propose to measure the difference between Σ_1 and Σ_2 using matrix norms. We have experimented with the Frobenius norm and the spectral norm. The Frobenius norm evaluates the difference between Σ_1 and Σ_2 when they act on a standard basis. The spectral norm, on the other hand, evaluates the difference when they act on the direction of maximal variance over the whole space.

3 Data Preprocessing

We use BNC [2] and a list of verb-noun constructions (VNCs) extracted from BNC by [6,8] and labeled as L (Literal), I (Idioms), or Q (Unknown).

The list contains only those VNCs whose frequency was greater than 20 and that occurred at least in one of two idiom dictionaries [7,23]. The dataset consists of 2,984 VNC tokens. For our experiments we only use VNCs that are annotated as I or L. We only experimented with idioms that can have both literal and idiomatic interpretations. We should mention that our approach can be applied to any syntactic construction. We decided to use VNCs only because this dataset was available and for fair comparison – most work on idiom recognition relies on this dataset.

We use the original SGML annotation to extract paragraphs from BNC. Each document contains three paragraphs: a paragraph with a target VNC, the preceding paragraph and following one.

Since BNC did not contain enough examples, we extracted additional ones from COCA, COHA and GloWbE (http://corpus.byu.edu/). Two human annotators labeled this new dataset for idioms and literals. The inter-annotator agreement was relatively low (Cohen's kappa = .58); therefore, we merged the results keeping only those entries on which the two annotators agreed.

Table 1. Datasets: Is = idioms; Ls = literals

Expression	Train	Test
BlowWhistle	20 Is, 20 Ls	7 Is, 31 Ls
LoseHead	15 Is, 15 Ls	6 Is, 4 Ls
MakeScene	15 Is, 15 Ls	15 Is, 5 Ls
TakeHeart	15 Is, 15 Ls	46 Is, 5 Ls
BlowTop	20 Is, 20 Ls	8 Is, 13 Ls
BlowTrumpet	50 Is, 50 Ls	61 Is, 186 Ls
GiveSack	20 Is, 20 Ls	26 Is, 36 Ls
HaveWord	30 Is, 30 Ls	37 Is, 40 Ls
HitRoof	50 Is, 50 Ls	42 is, 68 Ls
HitWall	90 Is, 90 Ls	87 is, 154 Ls
HoldFire	20 Is, 20 Ls	98 Is, 6 Ls
HoldHorse	80 Is, 80 Ls	162 Is, 79 Ls

4 Experiments

We have carried out an empirical study evaluating the performance of the proposed techniques. The goal is to predict the idiomatic usage of VNCs.

4.1 Methods

For comparison, the following methods are evaluated.

1. $tf \cdot idf$: compute term by document matrix from training data with $tf \cdot idf$ weighting.
2. $p \cdot idf$: compute term by document matrix from training data with proposed $p \cdot idf$ weighting (3).
3. $p{\cdot}tf{\cdot}idf$: compute term by document matrix from training data with proposed p * tf-idf weighting (4).
4. $CoVAR_{Fro}$: proposed technique (5) described in Sect. 2.2, the distance between two matrices is computed using Frobenius norm.
5. $CoVAR_{Sp}$: proposed technique similar to $CoVAR_{Fro}$. However, the distance between two matrices is determined using the spectral norm.
6. *Context+ (CTX+)*: supervised version of the CONTEXT technique described in [8] (see below).

For methods from **1** to **3**, we compute a latent space from a term by document matrix obtained from the training data that captures 80% variance. To classify a test example, we compute cosine similarity between the test example and the training data in the latent space to make a decision.

For methods **4** and **5**, we compute literal and idiomatic scatter matrices from training data (5). For a test example, compute a scatter matrix according to (5), and calculate the distance between the test scatter matrix and training scatter matrices using the Frobenius norm for method **4**, and the spectral norm for method **5**.

Method **6** corresponds to a supervised version of CONTEXT described in [8]. CONTEXT is unsupervised because it does not rely on manually annotated training data, rather it uses knowledge about automatically acquired canonical forms (C-forms). C-forms are fixed forms corresponding to the syntactic patterns in which the idiom normally occurs. Thus, the gold-standard is "noisy" in CONTEXT. Here we provide manually annotated training data. That is, the gold-standard is "clean." Therefore, CONTEXT+ is a supervised version of CONTEXT. We implemented this approach from scratch since we had no access to the code and the tools used in the original article and applied this method to our dataset and the performance results are reported in Table 3.

4.2 Word Vectors

For our experiments reported here, we obtained word vectors using the word2vec tool [17,18] and the text8 corpus. The text8 corpus has more than 17 million words, which can be obtained from mattmahoney.net/dc/text8.zip. The following shows a sample text8 corpus:

> anarchism originated as a term of abuse first used against early working class radicals including the diggers of the english revolution and the sans culottes of the french revolution whilst the term is still used in a pejorative way to describe any act that used violent means to destroy the organization of society it has also been taken up as a positive label by self defined anarchists the word anarchism is derived from the greek

without archons ruler chief king anarchism as a political philosophy is the belief that rulers are unnecessary and should be abolished although there are differing interpretations of what this means anarchism also refers to related social movements that advocate the elimination of authoritarian institutions particularly the state the word anarchy as most anarchists use it does not imply chaos nihilism or anomie but rather a harmonious anti authoritarian society in place of what are regarded as authoritarian political structures and coercive economic institutions anarchists advocate social relations based upon voluntary association of autonomous individuals mutual aid

The resulting vocabulary has 71,290 words, each of which is represented by a $q = 200$ dimension vector. Thus, this 200 dimensional vector space provides a basis for our experiments.

4.3 Datasets

Table 1 describes the datasets we used to evaluate the performance of the proposed technique. All these verb-noun constructions are ambiguous between literal and idiomatic interpretations. The examples below (from the corpora we used) show how these expressions can be used *literally*.

BlowWhistle: *we can immediately turn towards a high-pitched sound such as whistle being blown. The ability to accurately locate a noise is particularly important for the animals which use sound to find their way around.*

LoseHead: *Here are several degrees of deception. The simplest involves displaying a large eye-spot marking somewhere at the rear end of the body. This looks as eye-like to the predator as the real eye and gives the prey a fifty-fifty chance of losing its head. That was a very nice bull I shot, but I lost his head.*

MakeScene: *In another analogy our mind can be thought of as a huge tapestry in which the many episodes of life were originally isolated and there was no relationship between the parts, but at last we must make a unified scene of our whole life.*

TakeHeart: *He sacrifices four lambs . . . then takes two inside and kills them by slitting open the throat and the chest and cutting off one of the forelegs at the shoulder so the heart can be taken out still pumping and offered to the god on a plate.*

BlowTop: *Yellowstone has no large sources of water to create the amount of steam to blow its top as in previous eruptions.*

5 Results

Table 2 shows the average precision, recall and accuracy of the competing methods on **BlowWhistle**, **LoseHead**, **MakeScene**, **BlowTop**, **BlowTrumpet**, and **GiveSack** over 20 runs. Table 3 shows the performance on **HitRoof**, **HitWall**, **HoldFire**, **TakeHeart**, **HaveWord**, and **HoldHorse**. The best performance is in bold face. The best model is identified by considering precision,

Table 2. Average precision, recall, and accuracy by each method on **BlowWhistle**, **LoseHead**, **MakeScene**, **BlowTop**, **BlowTrumpet**, and **GiveSack** datasets.

Method	BlowWhistle			LoseHead			MakeScene		
	Pre	Rec	Acc	Pre	Rec	Acc	Pre	Rec	Acc
$tf \cdot idf$	0.23	0.75	0.42	0.27	0.21	0.49	0.41	0.13	0.33
$p \cdot idf$	0.29	0.82	0.60	0.49	0.27	0.48	0.82	0.48	0.53
$p \cdot tf \cdot idf$	0.23	0.99	0.37	0.31	0.30	0.49	0.40	0.11	0.33
$CoVAR_{Fro}$	**0.65**	**0.71**	**0.87**	0.60	0.78	0.58	**0.84**	**0.83**	**0.75**
$CoVAR_{sp}$	0.44	0.77	0.77	**0.62**	**0.81**	**0.61**	0.80	0.82	0.72
$CTX+$	0.17	0.56	0.40	0.55	0.52	0.46	0.78	0.0.37	0.45
	BlowTop			BlowTrumpet			GiveSack		
	Pre	Rec	Acc	Pre	Rec	Acc	Pre	Rec	Acc
$tf \cdot idf$	0.55	0.93	0.65	0.26	0.85	0.36	0.61	0.63	0.55
$p \cdot idf$	0.59	0.58	0.68	0.44	0.85	0.69	0.55	0.47	0.62
$p \cdot tf \cdot idf$	0.54	0.53	0.65	0.33	0.93	0.51	0.54	0.64	0.55
$CoVAR_{Fro}$	**0.81**	**0.87**	**0.86**	0.45	0.94	0.70	0.63	0.88	0.72
$CoVAR_{sp}$	0.71	0.79	0.79	0.39	0.89	0.62	0.66	0.75	0.73
$CTX+$	0.66	0.70	0.75	**0.59**	**0.81**	**0.81**	**0.67**	**0.83**	**0.76**

Table 3. Average precision, recall, and accuracy by each method on **HitRoof**, **HitWall**, **HoldFire**, **TakeHeart**, **HaveWord**, and **HoldHorse** datasets.

Method	HitRoof			HitWall			HoldFire		
	Pre	Rec	Acc	Pre	Rec	Acc	Pre	Rec	Acc
$tf \cdot idf$	0.42	0.70	0.52	0.37	0.99	0.39	0.91	0.57	0.57
$p \cdot idf$	0.54	0.84	0.66	0.55	0.92	0.70	0.97	0.83	0.81
$p \cdot tf \cdot idf$	0.41	0.98	0.45	0.39	0.97	0.43	**0.95**	**0.89**	**0.85**
$CoVAR_{Fro}$	**0.61**	**0.88**	**0.74**	**0.59**	**0.94**	**0.74**	0.97	0.86	0.84
$CoVAR_{sp}$	0.54	0.85	0.66	0.50	0.95	0.64	0.96	0.87	0.84
$CTX+$	0.55	0.82	0.67	0.92	0.57	0.71	0.97	0.64	0.64
	TakeHeart			HaveWord			HoldHorse		
	Pre	Rec	Acc	Pre	Rec	Acc	Pre	Rec	Acc
$tf \cdot idf$	0.65	0.02	0.11	0.52	0.33	0.52	0.79	0.98	0.80
$p \cdot idf$	0.90	0.43	0.44	0.52	0.53	0.54	0.86	0.81	0.78
$p \cdot tf \cdot idf$	0.78	0.11	0.18	0.53	0.53	0.53	0.84	0.97	0.86
$CoVAR_{Fro}$	0.95	0.61	0.62	0.58	0.49	0.58	**0.86**	**0.97**	**0.87**
$CoVAR_{sp}$	0.94	0.55	0.56	0.56	0.53	0.58	0.77	0.85	0.73
$CTX+$	**0.92**	**0.66**	**0.64**	**0.53**	**0.85**	**0.57**	**0.93**	**0.89**	**0.88**

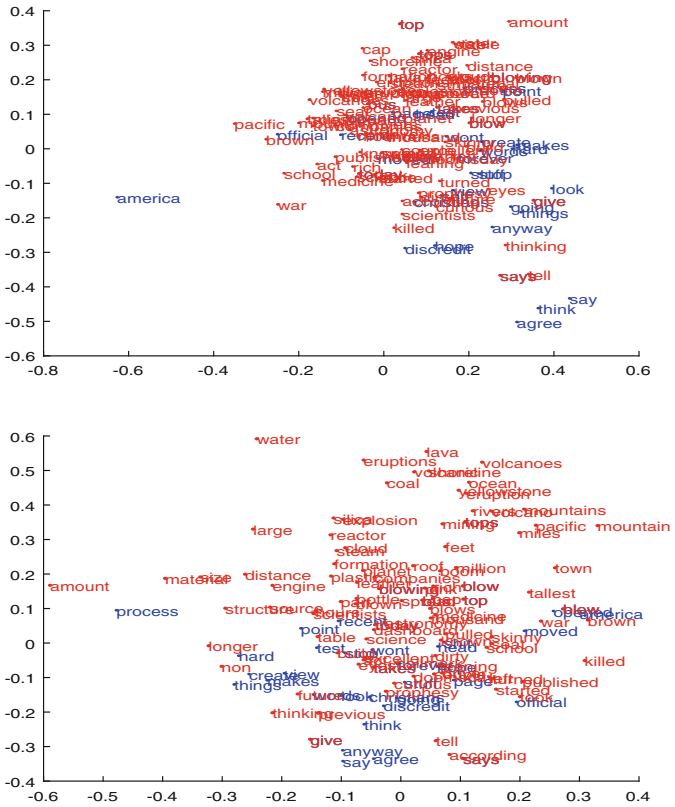

Fig. 1. Top: Projection of context words onto the subspace spanned by the first two eigenvectors of the idiomatic scatter. Bottom: Projection of context words onto the subspace of the literal scatter matrix. The blue words are idiomatic context, while the red words are literal context. (Color figure online)

recall, and accuracy together for each model. We calculate accuracy by adding true positives (idioms) and true negatives (literals) and normalizing the sum by the number of examples.

Figure 1 shows the projection of context words of **blow top** onto a subspace spanned by the first two eigenvectors of the scatter matrices (5). The top plot in Fig. 1 shows the projection onto the subspace of the idiomatic scatter matrix, while the lower plot shows the projection of the literal scatter matrix. The blue indicates idiomatic context words, while the red indicates literal context words. This shows that the scatter matrices (5) seem capable of capturing the difference in distributions between idiomatic context words and literal ones.

Interestingly, the Frobenius norm outperforms the spectral norm. One possible explanation is that the spectral norm evaluates the difference when two matrices act on the maximal variance direction, while the Frobenius norm evaluates on a standard basis. That is, Frobenius measures the difference along

all basis vectors. On the other hand, the spectral norm evaluates changes in a particular direction. When the difference is a result of all basis directions, the Frobenius norm potentially provides a better measurement. The projection methods ($p \cdot idf$ and $p \cdot tf \cdot idf$) outperform $tf \cdot idf$ overall but not as pronounced as $CoVAR$.

$CTX+$ demonstrates a very competitive performance. Since $CTX+$ is a supervised version of CONTEXT, we expect our proposed algorithms to outperform Fazly's CONTEXT method.

6 Related Work

Previous approaches to idiom detection can be classified into two 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 such as non-compositionality that can distinguish them from literal expressions [8,22]. 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 potentially idiomatic whereas *The bucket was kicked* is not idiomatic anymore; and of course, (3) non-compositionality. Thus, some approaches look at the tendency for words to occur in one particular order, or a fixed pattern. [14] identifies lexico-syntactic patterns that occur frequently, are recognizable with little or no precoded knowledge, and indicate the lexical relation of interest. [26] use Hearst's concept of lexicosyntactic patterns to extract idioms that consist of fixed patterns between two nouns. Basically, their technique works by finding patterns such as "thrills and spills", whose reversals (such as "spills and thrills") are never encountered.

While many idioms do have these properties, many idioms fall on the continuum from being compositional to being partly unanalyzable to completely non-compositional [5]. [8,16], 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 have been discussed in the literature, both supervised (Katz and Giesbrecht 2006), weakly supervised [1] and unsupervised [8,24]. [24] present a graph-based model for representing the lexical cohesion of a discourse. Nodes represent 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 [20]; (2) Normalized Google Distance [4]. [24] show that this method works better for larger contexts (greater than five paragraphs). [16] 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 less semantic cohesive ties with the context than literal language. Their results are inconclusive, due to the small size of the test corpus. [21] investigate the bag of words *topic* representation and incorporate an additional hypothesis–contexts in which idioms occur are more affective. Still, they treat idioms as semantic outliers.

[9] describe several approaches to automatic idiom identification. One of them is idiom recognition as outlier detection. They apply principal component analysis for outlier detection – an approach that does not rely on costly annotated training data and is not limited to a specific type of a syntactic construction, and is generally language independent.

7 Conclusions

In this paper we described an original algorithm for automatic classification of idiomatic and literal expressions. We also compared our algorithm against several competing idiom detection algorithms discussed in the literature. The performance results show that our algorithm generally outperforms [8] model. Note that $CTX+$ is a supervised version of [8], in that the training data here is the true "gold-standard," while in [8] is noisy. A research direction is to incorporate affect into our model. Idioms are typically used to imply a certain evaluation or affective stance toward the things they denote [19,22]. We usually do not use idioms to describe neutral situations, such as buying tickets or reading a book. Similarly to [21] we are exploring ways to incorporate affect into our idiom detection algorithm. Even though our method was tested on verb-noun constructions, it is independent of syntactic structure and can be applied to any idiom type. Unlike [8] approach, for example, our algorithm is language-independent and does not rely on POS taggers and syntactic parsers, which are often unavailable for resource-poor languages. Our next step is to expand this method and use it for idiom discovery. The move will imply an extra step – extracting multiword expressions first and then determining their status as literal or idiomatic.

Acknowledgements. This material is based upon work supported by the National Science Foundation under Grant No. 1319846.

References

1. Birke, J., Sarkar, A.: 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 2006), Trento, pp. 329–336 (2006)
2. Burnard, L.: The British National Corpus Users Reference Guide. Oxford University Computing Services, Oxford (2000)
3. Cacciari, C.: The place of idioms in a literal and metaphorical world. In: Cacciari, C., Tabossi, P. (eds.) Idioms: Processing, Structure, and Interpretation, pp. 27–53. Lawrence Erlbaum Associates, Hillsdale (1993)

4. Cilibrasi, R., Vitányi, P.M.B.: The Google similarity distance. IEEE Trans. Knowl. Data Eng. **19**(3), 370–383 (2007)
5. Cook, P., Fazly, A., Stevenson, S.: Pulling their weight: exploiting syntactic forms for the automatic identification of idiomatic expressions in context. In: Proceedings of the ACL 2007 Workshop on A Broader Perspective on Multiword Expressions, pp. 41–48 (2007)
6. Cook, P., Fazly, A., Stevenson, S.: The VNC-tokens dataset. In: Proceedings of the LREC Workshop: Towards a Shared Task for Multiword Expressions (MWE 2008), Marrakech, June 2008
7. Cowie, A.P., Mackin, R., McCaig, I.R.: Oxford Dictionary of Current Idiomatic English, vol. 2. Oxford University Press, Oxford (1983)
8. Fazly, A., Cook, P., Stevenson, S.: Unsupervised type and token identification of idiomatic expressions. Comput. Linguist. **35**(1), 61–103 (2009)
9. Feldman, A., Peng, J.: Automatic detection of idiomatic clauses. In: Gelbukh, A. (ed.) CICLing 2013. LNCS, vol. 7816, pp. 435–446. Springer, Heidelberg (2013). doi:10.1007/978-3-642-37247-6_35
10. Fellbaum, C., Geyken, A., Herold, A., Koerner, F., Neumann, G.: Corpus-based studies of German idioms and light verbs. Int. J. Lexicogr. **19**(4), 349–360 (2006)
11. Firth, J.R.: A synopsis of linguistic theory, 1930–1955 (1957)
12. Fukunaga, K.: Introduction to Statistical Pattern Recognition. Academic Press, New York (1990)
13. Glucksberg, S.: Idiom meanings and allusional content. In: Cacciari, C., Tabossi, P. (eds.) Idioms: Processing, Structure, and Interpretation, pp. 3–26. Lawrence Erlbaum Associates, Hillsdale (1993)
14. Hearst, M.A.: Automatic acquisition of hyponyms from large text corpora. In: Proceedings of the 14th Conference on Computational Linguistics (COLING 1992), vol. 2, pp. 539–545. Association for Computational Linguistics, Stroudsburg (1992). http://dx.doi.org/10.3115/992133.992154
15. Katz, G., Giesbrecht, E.: Automatic identification of non-compositional multiword expressions using latent semantic analysis. In: Proceedings of the ACL/COLING-06 Workshop on Multiword Expressions: Identifying and Exploiting Underlying Properties, pp. 12–19 (2006)
16. Li, L., Sporleder, C.: Using Gaussian mixture models to detect figurative language in context. In: Proceedings of the NAACL/HLT 2010 (2010)
17. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. In: Proceedings of Workshop at ICLR (2013)
18. Mikolov, T., Sutskever, I., Chen, K., Corrado, G., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Proceedings of the NIPS (2013)
19. Nunberg, G., Sag, I.A., Wasow, T.: Idioms. Language **70**(3), 491–538 (1994)
20. Pado, S., Lapata, M.: Dependency-based construction of semantic space models. Comput. Linguist. **33**(2), 161–199 (2007)
21. Peng, J., Feldman, A., Vylomova, E.: Classifying idiomatic and literal expressions using topic models and intensity of emotions. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 2019–2027. Association for Computational Linguistics, Doha, October 2014. http://www.aclweb.org/anthology/D14-1216
22. Sag, I.A., Baldwin, T., Bond, F., Copestake, A., Flickinger, D.: 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, pp. 1–15 (2002)

23. Seaton, M., Macaulay, A. (eds.): Collins COBUILD Idioms Dictionary, 2nd edn. HarperCollins Publishers (2002)
24. Sporleder, C., Li, L.: Unsupervised recognition of literal and non-literal use of idiomatic expressions. In: Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2009), pp. 754–762. Association for Computational Linguistics, Morristown (2009)
25. Villavicencio, A., Copestake, A., Waldron, B., Lambeau, F.: Lexical encoding of MWEs. In: Proceedings of the Second ACL Workshop on Multiword Expressions: Integrating Processing, Barcelona, pp. 80–87 (2004)
26. Widdows, D., Dorow, B.: Automatic extraction of idioms using graph analysis and asymmetric lexicosyntactic patterns. In: Proceedings of the ACL-SIGLEX Workshop on Deep Lexical Acquisition (DeepLA 2005), pp. 48–56. Association for Computational Linguistics, Stroudsburg (2005). http://dl.acm.org/citation.cfm?id=1631850.1631856

Springer

http://www.springer.com/978-3-319-55208-8

Information Management and Big Data
Second Annual International Symposium, SIMBig 2015, Cusco, Peru, September 2-4, 2015, and Third Annual International Symposium, SIMBig 2016, Cusco, Peru, September 1-3, 2016, Revised Selected Papers
Lossio-Ventura, J.A.; Alatrista-Salas, H. (Eds.)

2017, XI, 147 p. 46 illus., Softcover
ISBN: 978-3-319-55208-8