# GEOMETRY OF COMPOSITIONALITY

BY

HONGYU GONG

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Electrical and Computer Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2017

Urbana, Illinois

Advisers:

Professor Pramod Viswanath
Assistant Professor Suma Bhat

# ABSTRACT

Word embedding is a popular representation of words in vector space, and its geometry reveals the lexical semantics. This thesis further explores the interesting geometric properties of word embedding, and looks into its interaction with the context representation. We propose an innovative method to detect whether a given word or phrase is used literally in a specific context. This work focuses on three specific applications in natural language processing: idiomaticity, sarcasm and metaphor detection. Extensive experiments have shown that this embedding-based method achieves good performance in multiple languages.

*To My Father and Mother*

# ACKNOWLEDGMENTS

This work would not have been possible without the guidance of my advisers: Prof. Pramod Viswanath and Prof. Suma Bhat, who contributed many good ideas and valuable applications to this thesis. Great discussions with my partners in Natural Language Processing, Tarek Sakakini and Jiaqi Mu, have also been of great help.

# TABLE OF CONTENTS

# LIST OF TABLES

# LIST OF FIGURES

# CHAPTER 1

# INTRODUCTION

Idiomatic expressions are used frequently, acting as an indispensable part of natural language. One type of such expressions is multiword expressions (MWEs) which are semantically idiosyncratic phrases [1]. For example, *kick the bucket*, *last straw* and *a hot potato* are multiword expressions. The semantic meaning of these phrases cannot be inferred from its component words, and therefore they are termed idiomatic or *non-compositional* phrases. In contrast, *compositional* phrases are those whose meaning is a composition of the component words' meaning.

An challenging aspect of MWEs is that their semantics and compositionality degree are highly context-sensitive [2]. For example, consider two contexts where the phrase *free lunch* occurs:
(1) Travelers on highways in the United States have enjoyed what felt like a *free lunch*.
(2) You can get something awesome at those pizza factory restaurants: a *free lunch* or a free scrumptious, lunch buffet on Veterans Day.
In (1), *free lunch* carries the non-compositional meaning of "something acquired without due effort or cost", while in (2) it has the interpretation of "lunch which is free".

The context-sensitive compositionality of a multiword expression is regarded as a key problem in many applications of natural language processing (NLP), especially for machine translation and information retrieval. In machine translation, the English phrase *kick the bucket* cannot be translated word by word, since the phrasal meaning is totally irrelevant with single words "kick" and "bucket". As for information retrieval, the retrieved documents containing words "beans" turn out to be unrelated with the query of "spill the beans" [3] . As such, accurate detection of non-compositional MWEs is a necessary step for a variety of NLP tasks.

In another example, the word *glad* displays opposite meanings in two dif-

ferent contexts. The first sentence is: "*Glad* that I spent extra money to buy my Brad Paisley tickets so early when there's plenty left now." The word has a sarcastic (hence non-compositional) sense which actually conveys the meaning of "upset" or "disappointed". In the second sentence: "Really *glad* to hear that youthcamp was so awesome! To God be the glory!", the word has a literal meaning of "happy" or "pleased". As is seen, the lexical compositionality relies heavily on the local context.

Another special case of non-compositional usage is *metaphors*. Here is the word *wear* used in two contexts:
(1) Teenagers *wear* attitude like a uniform.
(2) We always *wear* helmets when we are riding bikes.
In the first sentence, *wear* has a figurative (hence non-compositional) usage, while in the second sentence, the word carries the literal sense of "put on". This example again shows that compositionality is quite sensitive to the local context.

In this thesis, we focus on the compositionality detection of a given word or a phrase based on its context. Our method proposes to represent the local context as a linear subspace, and further uses the distance between the target phrase and the context space to quantify the compositionality degree. This approach brings two key innovations: (1) it integrates the contextual information into the compositionality detection; and (2) it is only based on word embeddings, not relying on other linguistic resources. We have achieved comparable or superior performance to recent works on different tasks including idiomaticity detection of MWEs [4], sarcasm detection [5] and metaphor detection [6].

This work explores the geometry of word embeddings, and uses it for the phrasal compositionality detection. Here are two primary questions in our study:
(1) What is a good representation of a long sentence?
(2) How can we quantify the degree of compositionality given the representations of the target phrase and its context?

We answer these questions through the geometry of word/context embedding in the vector space. The key insight is that word vectors in a context lie in a linear subspace, and that the phrasal compositionality can be measured by the distance between the target phrase embedding and its local context space.

We begin next with an introduction of word embeddings, and then discuss the context representation and the geometry of compositionality which leads to our algorithm of compositionality detection. Lastly empirical results are shown as a key justification for our context-based approach.

# CHAPTER 2

# RELATED WORKS

## 2.1 Semantic Representations

### 2.1.1 Word Embeddings

Words are traditionally modeled as atomic units, but a real-valued representation shows its power in many applications of natural language processing (NLP). One understanding of lexical semantics comes from a linguistic hypothesis: "a word is characterized by the company it keeps" [7]. This hypothesis inspires some distributed representations which extraordinarily successful in encoding semantics, and state-of-the-art works in word embeddings are word2vec and Glove [8, 9, 10, 11]. We use word2vec embeddings in our work, which is introduced next.

Word2vec trains word representations based on the contexts. It provides two models for embedding training, one is the continuous Skip-Gram model (SG), and the other is the Continuous Bag-of-Words model (CBOW).CBOW uses the context to predict the central word, and the training objective is to maximize the probability of the central word given its context. Suppose that there is a sentence denoted as a list of words, $l = \{w_{t-p}, \ldots, w_{t-1}, w_t, w_{t+1}, \ldots, w_{t+p}\}$, and the context around $w_t$ is $s = \{w_{t-p}, \ldots, w_{t-1}, w_{t+1}, \ldots, w_{t+p}\}$ . The probability of predicting the target word $w_t$ from the context is $p(w_t|s)$. The training objective is $L_{CBOW} = \sum_{w_t} \log p(w_t|s)$. Let $u(\cdot)$ be the embedding of a target word, and $v(\cdot)$ be the embedding of a context word. The average context vector, $\hat{v}(s)$, is defined as $\hat{v}(s) = \frac{1}{2p} \sum_{c=-p, c\neq 0}^{p} v(w_{t+i})$. The training

objective function of the CBOW model is:

$$L_{CBOW} = \sum_t \log p(w_t|s) = \sum_{w_t} \log(\text{softmax}(u(w_t)^T \hat{v})) \tag{2.1}$$
$$= \sum_{w_t} \log \frac{\exp(u(w_t)^T v)}{\sum_{w'} \exp(u(w')^T v)}$$

The Skip-Gram model, different from CBOW model, uses the central word to predict its context. Its training objective is to maximize the probability of context words given the central word, and context words are assumed to be independent for simplicity. The probability of predicting words in context $s$ based on the central word $w_t$ is

$$p(s|w_t) = p(w_{t-p}, \ldots, w_{t-1}, w_{t+1}, \ldots, w_{t+p}|w_t) = \prod_{c=-p, c\neq 0}^{p} p(w_{t+c}|w_t)$$

Further, the training objective of SG is

$$L_{SG} = \sum_t \log p(s|w_t) = \sum_t \sum_{c=-p, c\neq 0}^{p} \log p(w_{t+c}|w_t)$$

Denote the central word as $w$ and its context word as $c$. Let $\sigma$ as sigmoid function, i.e., $\sigma(x) = \frac{1}{1+e^{-x}}$. With negative sampling, two types of probabilities are re-defined as below:

$$p(c, w \text{ co-occur}) = p(c|w) = \sigma(u(w)^T v(c))$$
$$p(c, w \text{ not co-occur}) = 1 - p(c|w) = \sigma(-u(w)^T v(c))$$

The training objective of skip-gram model with negative sample is defined as:

$$L_{SG}(u, v) = \sum_{(w,c)} n(w, c) \log(\sigma(u(w)^T v(c))) + k\mathbb{E}_{c'} \log(\sigma(-u(w)^T v(c'))) \tag{2.2}$$

where $n(w, c)$ is the number of occurrences of word-context pairs $(w, c)$, $k$ is the number of negative samples, and $c'$ is a randomly generated word that does not co-occur with $w$ through an empirical unigram distribution.

Word embedding $u$ is trained to maximize the objective function in (2.2). Suppose that two words are synonyms $w_1$ and $w_2$. It means that $w_1$ can be replaced with $w_2$ in the context which $w_1$ occurs. In the SG model, for any context word $c$, we have

$$u(w_1)^T v(c) - u(w_2)^T v(c) = \log\left(\frac{p(c|w_1)}{p(c|w_2)}\right) \approx 0$$

It implies that two vectors $u(w_1)$ and $u(w_2)$ should be close. Empirically the representations of semantically similar or relevant words are close in vector space. This is a basic geometric property underlying our compositionality detection algorithm.

### 2.1.2 Sentence Representation

Latent semantic analysis (LSA) is a classic method for representation of long sentences such as documents [12]. After generating a document-word count matrix, it applies singular value decomposition to generate low-dimension vectors for each document. The weakness of LSA is that it does not work well for short sentences since many entries in the word count matrix are 0's.

Based on the trained word embeddings, sentence representation can be generated by averaging component words. Such an average embedding method turns out to be a simple but robust way to encode variable-length sentences into fixed-dimension vectors. It has applications in measuring text similarity as well as phrase similarity [13, 14]. It is also used to predict surrounding sentences as an input to the neural network [15]. Average embeddings are shown to work well in answer selection for the given questions [16]. In the part of idiomaticity detection, we use average embedding as a baseline and show that it performs reasonably well, although the subspace representation proposed in this work is statistically significantly superior.

Besides average vector representations, another popular sentence embedding comes from the internal representations by the hidden units in neural networks such as LSTM. These two methods are compared on different low-level text prediction tasks in [17, 18], and it is pointed out that each method has its strengths. The average of CBOW embeddings is effective in content representations without considering word order, while LSTM representation

is good at preserving the order information.

Many other sentence representations have been proposed in recent studies. Doc2Vec, built on an idea similar to word2vec that trained word embedding to predict co-occurring words, trains sentence embedding by the prediction of its component words [19]. Doc2vec generates word embeddings and sentence embeddings at the same time.

## 2.2 Compositionality Detection

### 2.2.1 Idiom Detection

Detection of idioms is the first step of semantic understanding. Take the polysemous phrase "blue sky" as an example. It refers to "impractical things" when used idiomatically, whereas referring to "the sky that is blue" when used compositionally. Early approaches on idiom detection are statistical methods which involve the computation of word co-occurrence probabilities [20]. Some approaches make use of linguistic resources, relying on phrasal syntactic properties for idiom detection [21, 22, 23].

Wikitionary is a popular resource, which provides a list of idiomatic phrases, idiom tags, definitions as well as the synonyms [24]. Some recent works also turn to multilingual resources, using machine translation to decode the real semantics of target phrases [25, 26]. These methods are heavily resource dependent, and they have limited applicability since they may not work for other languages which do not have rich linguistic resources.

One resource-independent work explores compositionality with the combination of word embeddings [4]. It detects idiomaticity by measuring the difference between the phrase embedding and the component word embedding. One limitation of this method is that idiomaticity does not only rely on the phrase itself, but also on its context especially for some polysemous phrases. This motivates us to develop a context-based detection method.

## 2.3 Sarcasm Detection

Sarcasm, conveying opposite meaning to its literal sense, is a special case of non-compositional usage of languages. As an important part of sentiment analysis, sarcasm detection has been studied in many recent works [27]. A semi-supervised sarcasm identification algorithm (SASI) is proposed to deal with sarcasms in two stages: sarcastic pattern recognition and pattern classification [28]. Another method first divides training sentences and phrases into positive and negative usages as related to sarcasms [29]. The system classifies a test example by measuring how similar it is to the training examples. Another method takes unigram, bigram and trigrams as features into a supervised winnow classifier for sarcasm detection [30]. Sarcasm detection can also be considered as a word sense disambiguation task [31]. This work trains word embeddings with a large labeled Twitter dataset. The disambiguation decision is based on the similarity of the test context and the trained literal and sarcastic contexts.

## 2.4 Metaphor Detection

Metaphor is a figurative expression which refers to one thing by mentioning another [32], and is another special case of non-compositional language usage. CorMet is a system proposed to reveal metaphorical relations among words by establishing a mapping from a source domain to a target domain. Some works suggest that metaphor usage is related with psychology, and they rely on the MRC Psycholinguistic Database Machine Usable Dictionary (MRCPD).

One method uses this psycholinguistic database to measure the abstractness of sentences [33], and another work extracts lexical imaginability and topic clustering from this resource. Besides the abstractness and imaginability, a work shows that lexical supersenses are also critical features in metaphor detection, and supersenses are obtained from WordNet [6]. These previous works are heavily reliant on external linguistic resources.

# CHAPTER 3

# COMPOSITIONALITY AND GEOMETRY OF WORD EMBEDDINGS

In this chapter, we formulate a general method for context-based compositionality detection. This algorithm is built on the geometric properties of a word/phrase and its context embeddings. We start with the intuition of applying principal components for sentence representation. Then we would provide empirical evidence for the interesting geometry of contexts. Lastly, we would introduce our detection method in details.

## 3.1 Sentence Subspace Representation

Empirically, embeddings of semantically similar or related words are close in terms of cosine similarity. It indicates that the vector norm does not make any difference to semantic similarity, and naturally word embeddings are insensitive to scaling operations. Phrasal embeddings are shown to be well approximated by the addition of component word embeddings. Furthermore, phrasal embeddings can be improved by tuning the weights of its components in the linear combination [4]. It is natural to consider a weighted linear combination as the sentence representation. We allow the weights to be tunable, and thus the sentence is actually a linear subspace specified by its component word vectors. Suppose a sentence consisting of $n$ words: $\{w_1, \ldots, w_n\}$. Let $v(w)$ be the embedding of a given word $w$, and the subspace is represented as a matrix $S$: $S = [v(w_1), \ldots, v(w_n)]$.

Later we realize that this representation is too noisy. Consider the case that the vector dimension is 300, and that the words in a sentence are as many as 300 words, so the linear subspace contains almost all word vectors because the matrix $S$ is close to full rank. This motivates us to refine the subspace representation, specifically, we want to find a new set of $m (m < n)$ vectors for sentence representation $T$: $T = [v'_1, \ldots, v'_m]$. The matrix $T$ is an

approximation of matrix $S$, aiming to keep the most important information in the original sentence representation $S$.

Since the matrix $T$ represents a linear subspace, for each vector $v \in S$, we can use a vector $v_{\text{approx}}$ which lies in subspace $T$ to approximate $v$ with the minimal approximation error. The approximation error is defined as $\|v - v_{\text{approx}}\|^2$. It is easy to compute that $v_{\text{approx}} = TT^T v$. For all vectors in $S$, the minimal approximation error achieved by $T$ is $\|S - TT^T S\|^2$. We need to find the best matrix $T^*$ such that

$$T^* = \arg\min_T \|S - TT^T S\|^2$$

Without loss of generality, we can assume that vectors in $T$ are orthonormal, i.e., $t_i^T t_j = \mathbf{1}[i = j]$. The solution to this optimization problem is exactly the principal directions of matrix $S$ [34].

The top $m$ principal directions of matrix $S$ can minimize the reconstruction error and thus keep as much sentential information as possible. Now we can use the linear subspace spanned by $T$ as a refined/denoised representation of the sentence. If a word $c$ is used compositionally, then it should be semantically related with other words in the same sentence. Since these words can be approximated by $T$ with small error, its embedding $w$ should be close to its approximation $w_{\text{approx}}$, i.e., $\|w - w_{\text{approx}}\|$ is small or their cosine similarity is large. Naturally, we can use their cosine similarity as the degree of lexical compositionality in the given context. In the next section, we will empirically validate the proposed compostionality metric with the geometric illustration.

## 3.2 Geometry of Contexts

When the target phrase carries a literal meaning, the compositional embedding of this phrase should be close to its contextual linear subspace consisting of principal directions (obtained from context word embeddings with principal component analysis). This phenomenon happens because target phrases are usually closely related or co-occur frequently with their context words.

We illustrate this phenomenon with the following example. Here is the phrase "acid test" occurring in two contexts:

(1) (compositional) Like the testing of gold with nitric acid, *acid test* is a chemical test for distinguishing gold from metals.
(2) (non-compositional) This is seen as an *acid test* of the government commitment to protecting our most valuable environments.

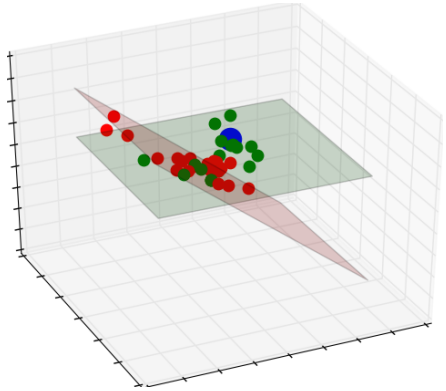


Figure 3.1: The phrasal embedding of *acid test* is shown by a blue point, and embeddings of compositional and non-compositional context words are denoted by green and red points respectively. The compositional context subspace is represented by the green plane, and the non-compositional context subspace by the red plane. Note that the phrase embedding is close to the compositional context plane while far from the non-compositional plane.

When words like *nitric, chemical* and *metals* appear in the context, "acid test" tends to carry a compositional sense of "specific chemical experiment". Conversely, when words like *government, commitment* and *protecting* are in the context, "acid test" is more likely to be an idiom, referring to general "verification". We plot the embeddings of the phrase and the two contexts in three-dimension space to visualize their geometry in Figure 3.1. As is shown, the phrase embedding resides roughly in the same subspace as the compositional context, whereas it is far away from the subspace of the non-compositional context. The detailed explanation of PCA operation, context representation and compositionality metric in this example will be discussed next.

## 3.3 Algorithm

The algorithm is inspired by a geometric property of vector embeddings of context words: they roughly occupy a linear subspace whose basis vectors can

be empirically extracted via a standard dimensionality reduction technique: principal component analysis (PCA) [34].

Suppose that the target phrase consists of words $u_1, ..., u_r$ ($r$ is usually $1 \sim 4$), and its contexts words are $w_1, \ldots, w_n$ ($n$ is usually $10 \sim 50$). Let $v(\cdot)$ denote the embedding of a given word.

**Phrase representation**: The phrase is represented as a vector $v_p$: $v_p = \frac{1}{r}\sum_{i=1}^{r} v(u_i)$, averaging vectors of all component words as in [35].

**PCA subspace representation**: Given $n$ words $(w_1, \ldots, w_n)$ in a sentence, their $d-dimension$ embeddings form an $n \times d$ matrix $X = (v(w_1), \ldots, v(w_n))$. Given $X$, PCA returns a $d \times m$ ($m < n$) matrix $X'$ to capture as much data variance in $X$ as possible. Here $X'$ consists of $m$ basis vectors, $(v'_1, ..., v'_m)$, where $v'_i$ is also a $d$-dimension vector, and $m$ is a hyperparameter to control how much information of original context should be kept. The principal directions in $X'$ are used as a subspace context representation for the given sentence. It is a key innovation in this work to represent a sentence with subspace instead of with a single vector. Subspace representation is shown to be important to our empirical results in a variety of tasks.

**Relevance metric**: Now given the vector representation $v_p$ of the target phrase, and the subspace representation $X'$ of the context, we can find the projection $v'_p$ of $v_p$ onto subspace $X'$:

$$v'_p = \arg \max_{v \in X', \|v\|=1} \frac{v^T v_p}{\|v\| \cdot \|v_p\|}$$

The relevance score $rs$ between the target and the context is the cosine similarity between the phrase vector and its projection:

$$rs = \frac{v_p^T v'_p}{\|v'_p\| \cdot \|v_p\|} \tag{3.1}$$

The phrase representation $v_p$ corresponds to its literal meaning, while the projection $v'_p$ corresponds to its true meaning in the given context. When $v_p$ is similar to $v'_p$, the phrase is used compositionally in the context. As such, relevance score $s$ measures the degree to which the word/phrase meaning is relevant with its context: *the larger the score s, the more the compositionality.*

According to the distributional hypothesis in linguistics, we know that the actual meaning of a word or a phrase can be inferred from its local context.

As such, the neighboring words in the context play a key role in decoding the actual sense of the target phrase. Previous works decided the phrasal compositionality regardless of the fact that it is highly context-dependent [36].

# CHAPTER 4

# THE DETECTION OF IDIOMS, SARCASMS AND METAPHORS

In this chapter, we evaluate the compositionality detection method empirically on three types of non-compositional language usages: idiom, sarcasm and metaphor. Specifically, we have three different tasks: (a) lexical/phrasal compositionality detection: deciding the literal usage or idiomatic usage of a given word/phrase in a sentence; (b) sarcasm detection: deciding whether a sentence conveys sarcastic meaning; (c) metaphor detection: deciding whether a given phrase is used in a metaphoric sense. For each of these tasks, we use standard datasets provided in recent works. Multiple languages such as English, German and Chinese are used in the evaluation, which can show the multilingual applicability of our algorithm.

We discuss idiomaticity detection in Section 4.1, sarcasm in Section 4.2, metaphor in Section 4.3. Our experiments are performed on standard datasets so as to compare our method with state-of-the-art results for each of these tasks. We also have datasets specifically designed for this work, and include datasets in German and Chinese besides those in English. We highlight the multilingual applicability of our algorithm via its good performance on multilingual datasets.

The embeddings used in our experiments are trained with the word2vec CBOW model, and the training corpus in English, German and Chinese are provided by polyglot [37]. CBOW provides one lexical embedding for each word. Some words have multiple senses which are quite different from each other, and we thus consider applying multiple sense embeddings instead of single lexical embedding. In addition to CBOW embeddings, we also train sense embeddings with the NP-MSSG model, which provides two sense embeddings for each word [11].

## 4.1 Idiom Detection

In this section, we focus on the idiomaticity detection task – phrasal idiomaticity detection and lexical idiomaticity detection. Here we formulate it as a binary classification problem, and decide whether a target word/phrase is idiomatic or compositional in the given sentence.

### 4.1.1 Phrasal Idiomaticity

Table 4.1: Examples of English Phrases, Whose Idiomaticity Depends on the Context.

| Phrase | Compositional Context | Idiomatic Context |
|---|---|---|
| **blue sky** | Above him was a clear **blue sky** and the sun floating on the surface of that milky sea of mist. | Unrealistic or impractical the author shows what is testable physics and **blue sky** nonsense. |
| **big fish** | There are many fish in the ocean. there are **big fish**. there are small fish. there are fast fish and slow fish. | He enjoys being a **big fish**, playing with the politicians who make a difference. |
| **black box** | Instead, the internet insurance co. will have constructed a **black box** it is overtly an algorithm that weighs various pieces of information about the applicant. | Her luggage consisted of a **black box**, and of a well worn leather bag which she carried in her hand. |

We start with polysemous phrases whose idiomaticity are highly context sensitive. Some examples of such phrases and their contexts are listed in Table 4.1, where the phrase *blue sky* is used compositionally in the first instance whereas used idiomatically in the second one. Here we have two sets of embeddings (lexical embeddings by CBOW and sense embeddings by MSSG), and two embedding composition methods for phrases ands sentences (average representation and subspace representation).

**Dataset**. We have two datasets specifically constructed for this work, one in English and one in Chinese. [1] Each dataset contains a list of phrases, and each phrase is accompanied by two contexts where it is used compositionally

[1]available at: `https://github.com/HongyuGong/Geometry-of-Compositionality.git`

Table 4.2: Accuracy Values (%) for Context-Based Phrasal Idiomaticity Detection.

| | English (CBOW) | English (MSSG) | Chinese (CBOW) | Chinese (MSSG) |
|---|---|---|---|---|
| average phrase average context | 80.3 | 82.7 | 78.1 | 50 |
| subspace phrase average context | 59.1 | 70.2 | 50.7 | 50.7 |
| average phrase subspace context | 82.7 | 84.6 | 80.5 | 75 |
| subspace phrase subspace context | 85.6 | **86.1** | 81.3 | **88.3** |

in one sentence, while idiomatically in the other. The English dataset has 104 phrases extracted from an English idiom dictionary [38], and the Chinese dataset has 64 phrases from a Chinese idiom dictionary [39]. Native English and native Chinese speakers select both compositional and idiomatic contexts for each phrase from these dictionaries and electronic books [40].

**Method**. We use a single vector as the phrase representation whether average or subspace representation (only keep the first principal direction) is used. As is defined in eq. (3.1), compositionality is the cosine similarity between the phrase vector and its projection onto the context subspace, when sentence is represented as a subspace. If sentence representation is approximated by the average vector, then the compositionality is measured by the cosine similarity between the phrase vector and sentence vector. Compositionality threshold is a hyperparameter in our method, a phrase is compositional if its compositionality is higher than the threshold; otherwise, it is idiomatic.

**Results**. We compare the predicted labels (compositional or idiomatic) with the human-annotated labels, and show the detection accuracy achieved by different representations and embeddings in Table 4.2. Given that average representation is shown to be good at capturing semantics in recent works [13, 14], we include the average representation as a baseline for the subspace representation. As is shown in both English and Chinese datasets, the best performance is achieved by subspace representations of phrases and sentences together with sense embeddings provided by MSSG. It validates that subspace representation is superior to the average representation.

The reason why subspace representation outperforms average approximation might be that principal directions are more robust to noise in the context.

Even when there are only a few contextual words related with the phrase of interest, average sentence approximation tends to decide that the phrase is literal. However, PCA approximation would filter the distracting noise, and capture the main information of the context.

### 4.1.2 Lexical Idiomaticity

Besides phrasal idiomaticity detection, here we also study the component-wise idiomaticity in bigram phrases. For example, "diamond" carries idiomatic sense whereas "wedding" carries its literal meaning in the phrase "diamond wedding". We have a single vector as the component word representation, and prepare both average and subspace representations for sentences. The compositionality measures are similar to that in phrasal idiomaticity detection. The lexical idiomaticitiy is decided via a comparison between the compositionality degree and the threshold. Here we tune the threshold on the training data instead of using fixed values.

Table 4.3: Experiments on ENC, EVPC and GNC Datasets.

| | | First Component | | | Second Component | | |
|---|---|---|---|---|---|---|---|
| Dataset | Method | Precision (%) | Recall (%) | F1 score (%) | Precision (%) | Recall (%) | F1 score (%) |
| ENC dataset | PMI | 50.0 | 100.0 | 66.7 | 40.4 | 100 | 57.6 |
| | LCS | 60.0 | 77.7 | 67.7 | 81.6 | 68.1 | 64.6 |
| | DS | 62.1 | 88.6 | 73.0 | 80.5 | 86.4 | 71.2 |
| | ALLDEFS+SYN | 81.2 | 88.1 | **84.5** | 87.3 | 80.6 | 69.8 |
| | ITAG+SYN | 64.5 | 90.9 | 75.5 | 61.8 | 94.4 | **74.7** |
| | Avg Cxt (CBOW) | 58.9 | 97.7 | 73.5 | 52.2 | 100 | 68.6 |
| | Avg Cxt (MSSG) | 68.5 | 79.5 | 73.7 | 61.2 | 83.3 | 70.6 |
| | Subspace (CBOW) | 78.4 | 90.9 | **84.2** | 67.44 | 80.6 | **73.44** |
| | Subspace (MSSG) | 68.9 | 90.9 | 78.4 | 57.6 | 94.4 | 71.6 |
| EVPC dataset | PMI | 22.2 | 68.4 | 33.5 | 53.0 | 80.2 | 63.8 |
| | LCS | 36.5 | 49.2 | 39.3 | 61.5 | 63.7 | 60.3 |
| | DS | 32.8 | 34.1 | 33.5 | 80.9 | 19.6 | 29.7 |
| | ALLDEFS+SYN | 37.4 | 70.9 | **48.9** | 80.4 | 65.9 | 63.0 |
| | ALLDEFS | 25.0 | 97.4 | 39.8 | 53.6 | 97.6 | **69.2** |
| | Avg Cxt (CBOW) | 27.1 | 92.1 | 41.9 | 58.7 | 74.4 | 65.6 |
| | Avg Cxt (MSSG) | 33.8 | 60.5 | 43.4 | 58.0 | 80.2 | 67.3 |
| | Subspace (CBOW) | 34.04 | 84.2 | **48.5** | 53.8 | 97.6 | 69.4 |
| | Subspace (MSSG) | 31.4 | 86.8 | **46.2** | 54.4 | 100 | **70.5** |
| GNC dataset | PMI | 44.2 | 99.0 | 61.1 | 26.4 | 98.4 | 41.7 |
| | Avg Cxt (CBOW) | 45.4 | 92.6 | 60.6 | 29.0 | 95.4 | 44.4 |
| | Avg Cxt (MSSG) | 44.0 | 77.8 | 56.2 | 31.7 | 67.7 | 43.1 |
| | Subspace (CBOW) | 45.8 | 95.4 | 61.8 | 32.2 | 75.4 | 45.1 |
| | Subspace (MSSG) | 45.5 | 99.1 | **62.4** | 30.9 | 86.2 | **45.5** |

**Dataset**. Three standard datasets used by previous works are available for lexical compositionality detection: English Noun Compounds (ENC), English Verb Particle Constructions (EVPC) and German Noun Compounds (GNC). The ENC dataset consists of 90 noun phrases whose compositionality is scored in a continuous range of $[0, 5]$ [36]. The EVPC dataset has 160 verb-particle phrases [41] which is assigned with binary labels either idiomatic or compositional. The GNC dataset provides 246 noun phrases, and each one is assigned a compositionality score on $[1, 7]$ scale. We formulate the compositionality detection as a binary classification task by setting a threshold of 2.5 to the ENC dataset and a threshold of 4 to the GNC dataset. Instances with compositionality higher than the threshold are taken as compositional phrases, and others are classified as idiomatic. We use these classification labels as gold results. On these three datasets, the task is to decide whether a component word is used compositionally in the given phrase. We report the precision, recall and f-score of classification results in Table 4.3.

**Results**. We give a comparison of different methods on the lexical idiomaticity detection. The following are methods of previous works shown in Table 4.3.

1. PMI (pointwise mutual information): a statistical measure to quantify the cohesiveness between two words. For the bigram phrase $w_1w_2$, PMI is defined as: $\text{PMI}(w_1w_2) = \log \frac{P(w_1w_2)}{P(w_1)P(w_2)}$, where $P(\cdot)$ is the likelihood of the word or phrase [42]. Higher PMI indicates that a phrase is more likely to be idiomatic.
2. LCS (longest common substring): a multilingual approach based on string similarity to idiomaticity detection [25].
3. DS (distributed similarity): a monolingual approach based on distributional similarity to idiomaticity detection [26].
4. ALLDEFS / ALLDEF+SYN: two state-of-the-art methods which make use of lexical and phrasal definitions, syntactic tags as well as synonyms provided by Wiktionary [24].
5. Avg Cxt (CBOW) / Avg Cxt (MSSG): averaged context representation using CBOW or MSSG embeddings.
6. Subspace (CBOW) / Subspace (MSSG): subspace context representation using CBOW or MSSG embeddings.

As we can see from empirical results in Table 4.3, the subspace representa-

tion achieves comparable or superior performance to state-of-the-art methods. It is an important advantage of our method that it does not depend on linguistic resources such as multilingual corpus and Wiktionary as recent methods [24, 26].

## 4.2 Sarcasm Detection

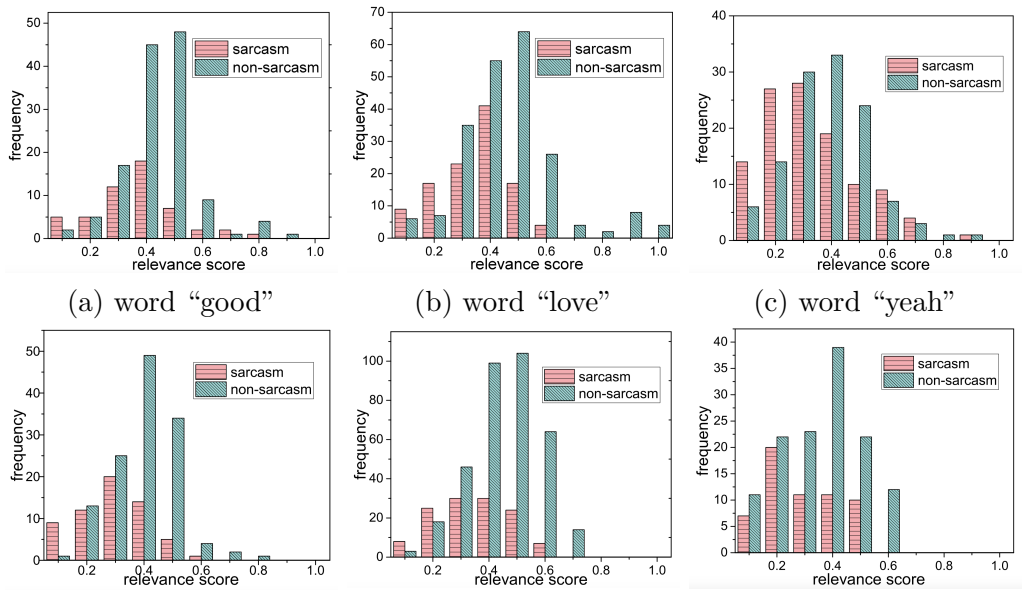
In this section, we apply context subspace representation to sarcasm detection. Sarcasms, also called irony, are expressions whose actual meaning is usually opposite to their literal meaning [28, 29]. Sarcasm is thus a special instance of non-compositional expressions. For example, the word "great" is used sarcastically in the sentence "Going to class on a empty stomach, sleepy and hungry, this will be great!" The speaker made complaints and expressed negative feelings with the positive word "great". We can infer the real sense of "great" from its context such as *sleepy* and *hungray*, which is more closely related with negative sentiments. Contexts provide key information in decoding the sarcasm, and such idea has been applied to previous works. These works have carefully designed contextual features as a basis of their sarcasm detection system [31]. We would use the relevance score generated by our compositionality detection algorithm as features directly for sarcasm detection.

### 4.2.1 Twitter Sarcasm

**Dataset**. Twitter provides an ideal forum for sarcasm to flourish, and some tweets are explicitly tagged with #sarcasm or #sarcastic hashtags. Recent works have collected tweets involving sarcasms, [31], of which we could download a part of them (due to privacy constraints). Some examples of sarcastic and literal usages of *good* in tweets are as follows:

1. Wow. Just wow. That's some damn **good** decision making. Good work ref! #sarcasm #NRLmancro
2. I better do fricken **good** on this midterm tomorrow based on the amount of studying I"ve done for it.

Six words are selected: "good", "nice", "love", "always", "yeah", and




(a) word "good" (b) word "love" (c) word "yeah"

(d) word "nice" (e) word "always" (f) word "glad"

Figure 4.1: Sarcasm detection in tweets.

Table 4.4: Twitter Sarcasm Detection.

| word | "good" | "love" | "yeah" | "nice" | "always" | "glad" |
|---|---|---|---|---|---|---|
| accuracy | 0.744 | 0.700 | 0.614 | 0.763 | 0.792 | 0.695 |
| F1 score | 0.610 | 0.64 | 0.655 | 0.623 | 0.605 | 0.582 |

"glad", which have enough occurrences in both literal and sarcastic senses in our downloaded dataset. We use subspace as sentence representation, and use compositionality detection method to derive the relevance score of the target in the given context. We divide the score range $[0, 1]$ into 10 equal-length segments: $[0, 0.1), [0.1, 0.2), \ldots$, and then count the occurrences in each score bin to see whether our algorithm could distinguish sarcastic usage from literal usage.

The histograms of the compositionality scores (for sarcastic and literal usages) are plotted in Figure 4.1. We can visually see that the two histograms (one for sarcastic usage and the other for literal usage) can be distinguished from each other, for each of the six words studied. The bar graph of sarcastic usage occupies the low-score region with peak at around 0.35, whereas the bar graph of literal usage occupies the high-score region peaking at around 0.45. It shows that this simple subspace-based compositionality detection method can be extended to sarcasm detection task.

We cast Twitter sarcasm detection as a binary classification problem. We

Table 4.5: Sarcasm Detection on Reddit Dataset.

| | Baseline | Subspace (JJ) | Subspace (VB) | Subspace (JJ+ RB) | Subspace (JJ+ RB+VB) |
|---|---|---|---|---|---|
| features | >50,000 | 2 | 2 | 3 | 4 |
| precision | 0.315 | 0.278 | 0.289 | 0.279 | 0.278 |
| recall | 0.496 | 0.936 | 0.844 | 0.98 | 0.747 |
| F1 score | 0.383 | 0.426 | 0.396 | **0.434** | 0.393 |

set a threshold, and decide the target word to be sarcastic if its relevance score is lower than the threshold. Besides the visual illustration, we also report accuracy and the F1 score on each target word achieved by this simple detection method in Table 4.4. Note that we only use a very small dataset in this experiment to show that the our compositionality detection method could distinguish the sarcastic and literal usage to some extent. The classification performance can be further improved when more complicated features and algorithms are incorporated into the current system.

### 4.2.2 Reddit Sarcasm

**Dataset**: A dataset on sarcasm detection task is made up of 3020 user comments (10401 sentences in total) collected from the Reddit website [5]. Human annotators labeled each comment with "ironic", "don't know" and "unironic". A sarcastic instance in the Reddit dataset is "Democrats don't know how to manage money? Shocking!". The task is to decide whether a given piece of comment is ironic. The state-of-art method generates rich linguistic features, and classifies comments using a support vector machine (SVM) with linear kernel [5].

**Method**. Content words such as adjectives (e.g. "great"), adverbs (e.g. "interestingly") and verbs (e.g. "like") are more likely to have ironic meaning than functional words such as prepositions and pronouns. As such, for each piece of user comment, we select the words with these POS tags: JJ (adjective), RB (adverb), and VB (verb) from a given comment. Their relevance scores in the local context can be obtained by our compositionality detection method. The lowest $k$ ($k = 2, 3, 4$) scores are selected as features for sarcasm classification. We use a SVM classifier with linear kernel and fivefold cross validation as the recent work [5] for a fair comparison. The empirical results of our method and state-of-the-art are shown in Table 4.5.

Table 4.6: Metaphor Detection.

| | | features | accuracy | f1 score |
|---|---|---|---|---|
| SVO | state-of-art | 279 | **0.82** | **0.86** |
| | Subspace original sentence | 4 | 0.729 | 0.744 |
| | Subspace longer sentence | 4 | 0.809 | 0.806 |
| AN | state-of-art | 360 | **0.86** | **0.85** |
| | Subspace original sentence | 3 | 0.735 | 0.744 |
| | Subspace longer sentence | 3 | 0.80 | 0.798 |

**Results**. As can be seen, our method achieves better performance with many fewer features in sarcasm detection, 5% higher in F1 score. The best features come from adjectives and adverbs.

## 4.3 Metaphor Detection

The last task we are tackling with is metaphor detection with the context space representation. Metaphors are figurative speech that refers to one thing by mentioning another. Hence they are used in an abstract and non-compositional way. For example, the word "kills" expresses the meaning of "opposes" in the sentence "Hawaii *kills* proposal for home energy loans.". Metaphors are naturally an instance of non-compositional semantics, and can also be studied under our framework of compositionality detection.

**Dataset**. There are English datasets with both metaphoric and literal phrases provided by [6]. The datasets focus on phrases with two specific syntactic structures: Subject-Verb-Object (SVO) structure and Adjective-Noun (AN) structure. An SVO metaphor is "excitement filled streets" in the sentence "For a brief moment this week, excitement filled the streets of both Havana and Miami." An AN metaphor is "dirty word" in the sentence "Solidarity of the European is now a dirty word in Germany when people talk about politics."

**Method**. The state-of-the-art work [6] performed feature engineering based on external resources like WordNet and the MRC psycholinguistic database. We again apply the context subspace representation and obtain relevance scores of target words. Departing from previous unsupervised clas-

sification of relevance scores, we generate syntactic features on the basis of relevance scores. Then we do binary classification to detect the metaphorical usage with random forest classifier.

Now we describe the features used for instances of certain syntactic structures. When an SVO or AN expression is a metaphor, there should be at least one word which does not agree with its context, i.e., it has a low relevance score. For the subject-verb-object instance, we have relevance scores for subject, verb and object respectively. The features we derive from these relevance scores are the following:

1. verb score: since a verb is likely to be the word inconsistent with the context, verb score is an informative feature.
2. the lowest score among subject, verb and object score: the lowest score captures the metaphoric word among three words.
3. the ratio between the lowest score and the highest score: relative ratio is more robust than the absolute score.
4. $\min\left(\frac{\text{verb score}}{\text{subj score}}, \frac{\text{subj score}}{\text{verb score}}, \frac{\text{verb score}}{\text{obj score}}, \frac{\text{obj score}}{\text{verb score}}\right)$: it selects the most metaphoric word among subject, verb and object via the ratio.

For the adjective-noun instance, we again obtain relevance scores for the adjective and the noun respectivey. The features we derive from them are the following:

1. the lowest score between adjective and noun: captures the most metaphoric word in the expression.
2. the highest score: captures the most literal word in the expression.
3. the ratio between the lowest and the highest score: relative ratio is more robust than the absolute score.

**Results**. The accuracy achieved by our method and state-of-the-art is in shown Table 4.6. In both the original SVO dataset and original AN dataset, although our performance is below the baseline, we only use a few features without reliance on external resources in contrast to the large number of features generated from rich resources in the baseline method.

With a closer look into the datasets, we realize that they contain many short sentences, e.g. "Jim closed the book". Short sentences are not able to provide high-quality contexts, and thus degrade the performance of the context-based detection. We replace sentences shorter than seven words with

longer sentences which contain the same phrase as the original sentences. These new sentences are obtained from online books [40]. The context-based metaphor detection has improved a lot in accuracy on the refined dataset than in the original dataset, as is shown in Table 4.6. We note that although our performance is still below the baseline, our method has a key advantage of resource-independence.

# CHAPTER 5

# CONCLUSION AND FUTURE WORK

## 5.1 Conclusion

We explore the curious geometry of word embeddings within a sentence: they are roughly lying in a linear subspace constructed by sentential principal directions. Bases on the sentence representation, we formulate a general method to measure the compositionality of the target word/phrase in a specific context. The compositionality computed with our method can be used directly to decide the lexical and phrasal idiomaticity. It can also be used to generate features in sarcasm and metaphor detection systems. Our method has achieved performance comparable or superior to state-of-the-art in these three tasks.

This is a lightweight detection method which can be easily applied to multiple languages. We show its applicability to English, German and Chinese in this thesis. Also, it has no reliance on external linguistic resources, which makes it useful for some resource-limited languages.

## 5.2 Future Work

**Applications of sentence representation.** We have shown the application of subspace representation to three applications: idiomaticity, sarcasm and metaphor detection. We can further apply the subspace representations to other applications such as context-based error correction, which could automatically perform corrections on lexical usages based on contextual semantics. Typos are frequently seen in online platforms. For example, "blu" is a typo in sentence "the sky is blu". Words "blue" and "blur" both have only one word different from "blu". But according to the context, we can

easily pick "blue" as the right word. The compositionality detection method could suggest such appropriate corrections in an efficient and accurate way.

**Extensions beyond bag-of-words model.** The subspace representation proposed in this work is a bag-of-words model, i.e., it does not consider the word order in the sentence. Word order does decide the semantics in some cases. For example, "I have to read a book" differs from "I have a book to read". In the future work, we will explore how to integrate the order information into sentence representations.

**Representations from neural networks.** Many recent works are applying neural networks to sentence/paragraph/document vector representations. In particular, Long Short-Term Memory (LSTM) is shown to well capture long dependencies in texts and keep the sequential information [43]. Also, it generates fixed-dimension vectors for input sentence of variable lengths. The connection between the neural network structure and vector representation power is a challenging avenue of future research.

# REFERENCES


[1] I. A. Sag, T. Baldwin, F. Bond, A. Copestake, and D. Flickinger, "Multiword expressions: A pain in the neck for NLP," in *Computational Linguistics and Intelligent Text Processing*. Springer, 2002, pp. 1–15.

[2] H. Gong, S. Bhat, and P. Viswanath, "Geometry of compositionality," in *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA.*, 2017. [Online]. Available: http://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14699 pp. 3202–3208.

[3] D. A. Evans and C. Zhai, "Noun-phrase analysis in unrestricted text for information retrieval," in *Proceedings of the 34th Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, 1996, pp. 17–24.

[4] B. Salehi, P. Cook, and T. Baldwin, "A word embedding approach to predicting the compositionality of multiword expressions," in *The North American Chapter of the Association for Computational Linguistics*, 2015, pp. 977–983.

[5] B. C. Wallace, L. K. Do Kook Choe, L. Kertz, and E. Charniak, "Humans require context to infer ironic intent (so computers probably do, too)." in *the Association for Computational Linguistics*, 2014, pp. 512–516.

[6] Y. Tsvetkov, L. Boytsov, A. Gershman, E. Nyberg, and C. Dyer, "Metaphor detection with cross-lingual model transfer," 2014.

[7] J. R. Firth, "A synopsis of linguistic theory," *Studies in Linguistic Analysis*, pp. 1–32, 1957.

[8] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in Neural Information Processing Systems*, 2013, pp. 3111–3119.

[9] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *CoRR*, vol. abs/1301.3781, 2013. [Online]. Available: http://arxiv.org/abs/1301.3781

[10] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation." in *EMNLP*, vol. 14, 2014, pp. 1532–1543.

[11] A. Neelakantan, J. Shankar, A. Passos, and A. McCallum, "Efficient non-parametric estimation of multiple embeddings per word in vector space," *Conference on Empirical Methods in Natural Language Processing*, 2014.

[12] G. Katz and E. Giesbrecht, "Automatic identification of non-compositional multi-word expressions using latent semantic analysis," in *Proceedings of the Workshop on Multiword Expressions: Identifying and Exploiting Underlying Properties*. Association for Computational Linguistics, 2006, pp. 12–19.

[13] T. Kenter and M. de Rijke, "Short text similarity with word embeddings," in *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*. ACM, 2015, pp. 1411–1420.

[14] S. J. Gershman and J. B. Tenenbaum, "Phrase similarity in humans and machines," in *Proceedings of the 37th Annual Conference of the Cognitive Science Society*. Citeseer, 2015.

[15] T. Kenter, A. Borisov, and M. de Rijke, "Siamese CBOW: Optimizing word embeddings for sentence representations," *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, p. 941951, 2016.

[16] L. Yu, K. M. Hermann, P. Blunsom, and S. Pulman, "Deep learning for answer sentence selection," *arXiv preprint arXiv:1412.1632*, 2014.

[17] Y. Adi, E. Kermany, Y. Belinkov, O. Lavi, and Y. Goldberg, "Fine-grained analysis of sentence embeddings using auxiliary prediction tasks," *CoRR*, vol. abs/1608.04207, 2016. [Online]. Available: http://arxiv.org/abs/1608.04207

[18] J. Wieting, M. Bansal, K. Gimpel, and K. Livescu, "Towards universal paraphrastic sentence embeddings," *arXiv preprint arXiv:1511.08198*, 2015.

[19] Q. Le and T. Mikolov, "Distributed representations of sentences and documents," in *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, 2014, pp. 1188–1196.

[20] D. Lin, "Automatic identification of non-compositional phrases," in *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics on Computational Linguistics*. Association for Computational Linguistics, 1999, pp. 317–324.

[21] D. McCarthy, B. Keller, and J. Carroll, "Detecting a continuum of compositionality in phrasal verbs," in *Proceedings of the ACL 2003 Workshop on Multiword Expressions: Analysis, Acquisition and Treatment*. Association for Computational Linguistics, 2003, pp. 73–80.

[22] P. Cook, A. Fazly, and S. Stevenson, "Pulling their weight: Exploiting syntactic forms for the automatic identification of idiomatic expressions in context," in *Proceedings of the Workshop on a Broader Perspective on Multiword Expressions*. Association for Computational Linguistics, 2007, pp. 41–48.

[23] A. Fazly and S. Stevenson, "Distinguishing subtypes of multiword expressions using linguistically-motivated statistical measures," in *Proceedings of the Workshop on a Broader Perspective on Multiword Expressions*. Association for Computational Linguistics, 2007, pp. 9–16.

[24] B. Salehi, P. Cook, and T. Baldwin, "Detecting non-compositional MWE components using Wiktionary," in *Conference on Empirical Methods in Natural Language Processing*, 2014, pp. 1792–1797.

[25] B. Salehi and P. Cook, "Predicting the compositionality of multiword expressions using translations in multiple languages," in *Second Joint Conference on Lexical and Computational Semantics*, vol. 1, 2013, pp. 266–275.

[26] B. Salehi, P. Cook, and T. Baldwin, "Using distributional similarity of multi-way translations to predict multiword expression compositionality." in *European Chapter of the Association for Computational Linguistics*, 2014, pp. 472–481.

[27] D. Maynard and M. A. Greenwood, "Who cares about sarcastic tweets? Investigating the impact of sarcasm on sentiment analysis." in *LREC*, 2014, pp. 4238–4243.

[28] D. Davidov, O. Tsur, and A. Rappoport, "Semi-supervised recognition of sarcastic sentences in Twitter and Amazon," in *Proceedings of the Fourteenth Conference on Computational Natural Language Learning*. Association for Computational Linguistics, 2010, pp. 107–116.

[29] E. Riloff, A. Qadir, P. Surve, L. De Silva, N. Gilbert, and R. Huang, "Sarcasm as contrast between a positive sentiment and negative situation." in *EMNLP*, vol. 13, 2013, pp. 704–714.

[30] C. Liebrecht, F. Kunneman, and A. van den Bosch, "The perfect solution for detecting sarcasm in tweets #not," in *Proceedings of the 4th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, WASSA@NAACL-HLT 2013, 14 June 2013, Atlanta, Georgia, USA*, 2013. [Online]. Available: http://aclweb.org/anthology/W/W13/W13-1605.pdf pp. 29–37.

[31] D. Ghosh, W. Guo, and S. Muresan, "Sarcastic or not: Word embeddings to predict the literal or sarcastic meaning of words," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, 2015. [Online]. Available: http://aclweb.org/anthology/D/D15/D15-1116.pdf pp. 1003–1012.

[32] G. Lakoff and M. Johnson, "Conceptual metaphor in everyday language," *The Journal of Philosophy*, vol. 77, no. 8, pp. 453–486, 1980.

[33] T. Liu, K. Cho, G. A. Broadwell, S. Shaikh, T. Strzalkowski, J. Lien, S. M. Taylor, L. Feldman, B. Yamrom, N. Webb, U. Boz, I. Cases, and C. Lin, "Automatic expansion of the MRC psycholinguistic database imageability ratings," in *Proceedings of the Ninth International Conference on Language Resources and Evaluation, LREC 2014, Reykjavik, Iceland, May 26-31, 2014.*, 2014. [Online]. Available: http://www.lrec-conf.org/proceedings/lrec2014/summaries/178.html pp. 2800–2805.

[34] J. Shlens, "A tutorial on principal component analysis," *CoRR*, vol. abs/1404.1100, 2014. [Online]. Available: http://arxiv.org/abs/1404.1100

[35] J. Mitchell and M. Lapata, "Composition in distributional models of semantics," *Cognitive Science*, vol. 34, no. 8, pp. 1388–1429, 2010.

[36] S. Reddy, D. McCarthy, and S. Manandhar, "An empirical study on compositionality in compound nouns," in *IJCNLP*, 2011, pp. 210–218.

[37] R. Al-Rfou, B. Perozzi, and S. Skiena, "Polyglot: Distributed word representations for multilingual NLP," in *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*. Sofia, Bulgaria: Association for Computational Linguistics, August 2013. [Online]. Available: http://www.aclweb.org/anthology/W13-3520 pp. 183–192.

[38] TheFreeDictionary, Available at: http://idioms.thefreedictionary.com, 2016, accessed: 2016-04-20.

[39] ChineseDictionary, Available at: http://www.chinese-dictionary.org, 2016, accessed:2016-05-01.

[40] GoogleBooks, Available at: https://books.google.com, 2016, accessed: 2016-05-03.

[41] C. J. Bannard, "Acquiring phrasal lexicons from corpora," Ph.D. dissertation, University of Edinburgh, 2006.

[42] C. Manning and H. Schütze, "Collocations," *Foundations of Statistical Natural Language Processing*, pp. 141–77, 1999.

[43] K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink, and J. Schmidhuber, "LSTM: A search space odyssey," *arXiv preprint arXiv:1503.04069*, 2015.