University of Washington

Abstract

Identifying Idiomatic Language at Scale

Grace Muzny

Idiomatic use of language is a naturally occurring phenomena that is ubiquitous in human language. This is problematic in many tasks within Natural Language Processing (NLP) because idiomatic phrases do not literally mean what they say—e.g. when someone is a “diamond in the rough”, they are a person whose goodness is hidden by their surface appearance. Language becomes idiomatic depending on the context that is occurs in—e.g. the phrase “diamond in the rough” can literally mean an uncut diamond. Given a dictionary entry, a human can easily distinguish between idiomatic and literal definitions, however, doing this in an automatic fashion is difficult because it requires asking whether the meaning represented in the definition corresponds with the literal meaning of the phrase.

This research leverages Wiktionary, an extremely large, collaboratively authored dictionary, to perform idiom identification in a scalable manner through supervised machine learning algorithms. It takes advantage of the annotations already present in Wiktionary to perform learning with the perceptron algorithm. The proposed models are able to learn from highly noisy training data to detect whether certain sense of a phrase is idiomatic or literal, achieving an F-measure of 67.20%—an increase over the best baseline performance of more than 200%.

This research is then extended to create a fully realized end-to-end system, that, given a phrase and an example sentence, can identify what sense that phrase is being used in and whether or not it is being used idiomatically. This component draws from Word Sense Disambiguation (WSD) to determine which sense a phrase is used in and then applies the learned models to that sense to determine whether the phrase has been used idiomatically. Ultimately, this system achieves an F-measure of 81.02% and an accuracy of 73.26% in the idiom identification task using only naive WSD algorithms.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Chapter 1: Introduction</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1 Definition of an Idiom</td>
<td>2</td>
</tr>
<tr>
<td>1.2 Motivation</td>
<td>3</td>
</tr>
<tr>
<td>1.3 Previous Work</td>
<td>4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Chapter 2: Background Information</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1 Perceptron Algorithm</td>
<td>7</td>
</tr>
<tr>
<td>2.2 Word Sense Disambiguation</td>
<td>7</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Chapter 3: Identifying Idiomatic Definitions</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1 Problem Definition</td>
<td>11</td>
</tr>
<tr>
<td>3.2 Experimental Setup</td>
<td>12</td>
</tr>
<tr>
<td>3.3 Features</td>
<td>13</td>
</tr>
<tr>
<td>3.4 Models &amp; Error Analysis</td>
<td>14</td>
</tr>
<tr>
<td>3.5 Results</td>
<td>17</td>
</tr>
<tr>
<td>3.6 Conclusions</td>
<td>18</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Chapter 4: Multi-Word Expression Sense Disambiguation for Idiom Detection</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1 Problem Definition</td>
<td>21</td>
</tr>
<tr>
<td>4.2 Multi-Word Expression Sense Disambiguation Results</td>
<td>22</td>
</tr>
<tr>
<td>4.3 Idiom Detection Results</td>
<td>23</td>
</tr>
<tr>
<td>4.4 System Limitations</td>
<td>24</td>
</tr>
<tr>
<td>4.5 Future &amp; Continuing Work</td>
<td>25</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Chapter 5: Conclusions</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bibliography</td>
<td>27</td>
</tr>
<tr>
<td>Appendix A: Annotation Guidelines</td>
<td>30</td>
</tr>
</tbody>
</table>
I would like to thank all people who have helped and supported me in my endeavors. Unfortunately, such an enumeration would be impossible.

My advisor for this research, Luke Zettlemoyer, who provided guidance, perspective, and advice—without which I would have been lost.

Kate Starbird and Leysia Palen, with whom I worked on my first “grown-up” research projects, and who give me unwavering support.

My parents, for first introducing me to the world of research and humoring me when I found it necessary to grow mold in pursuit of scientific knowledge.

And finally, I would like to thank my family and friends simply for being who they are. These are truly fantastic people and I would like all of them to understand this.
Chapter 1
INTRODUCTION

Figurative language presents a special problem in NLP because it does not mean what it literally says, which frequently results in errors in larger NLP systems [9]. While previous work has been done on automatic interpretation and identification of metaphors, it has been with considerable grammatical constraints, limiting the scope of identification to phrases with specific syntactic relationships. For example, much previous work has focused on Verb-Noun Constructions such as *kick the bucket* or *spill the beans*. This research concentrates on identification of figurative language—without grammatical restrictions—in a scalable manner.

This research leverages Wiktionary—an extremely large, collaboratively authored dictionary—to achieve scalability [24]. Wiktionary enables entries to be marked as a certain type, or category, of language—one of these categories is “English Idioms”. However, these labels depend on human annotation and there is no guarantee that all entries within Wiktionary that are English idioms are marked as such. In fact, the studies conducted in this research show that there are 2–3 times more idioms within Wiktionary than are marked as such.

The overall goal of this research is knowledge discovery—to build scalable, automated algorithms for identifying entries that have not yet been categorized as idioms, but that are in fact idiomatic. Further goals extend this research to the task of idiom detection, identifying if a particular phrase has been used idiomatically in a particular sentence.

The current state of the art in identification of figurative language is rather nebulous because of inconsistencies in performance reporting. The work of Shutova et al. [8] achieves a precision of .79 does not report true recall because it lacks a fully annotated data set to test on. The work of Diab and Bhutada [6] was, in 2009, state of the art and achieved an overall F-measure of 84.58%. These two models represent unsupervised and supervised approaches to the problem, respectively. They also represent two different paradigms in idiom identification research: that of phrase classification (in which all instances of phrase $p$ must be idiomatic), and that of token classification (in which
this particular occurrence of phrase $p$ is idiomatic, some examples of which are shown in Table 1.1). The work of Shutova et al. follows the first paradigm of phrase classification whereas the work of Diab and Bhutada follows the latter—and is the paradigm that this work is based on.

<table>
<thead>
<tr>
<th>Phrase</th>
<th>Example Sentence</th>
</tr>
</thead>
<tbody>
<tr>
<td>put up</td>
<td>That last fighter put up quite a fight.</td>
</tr>
<tr>
<td>kick in</td>
<td>You have to push the switch hard to get the heater to kick in.</td>
</tr>
<tr>
<td>back up</td>
<td>I couldn’t see how to finish the project, so I backed up and tried it another way.</td>
</tr>
</tbody>
</table>

Table 1.1: Idioms that Require Context

To the best of my knowledge, this research is the first to attempt the classification of definitions as idiomatic or literal senses of the phrase that they are associated with. Practically, this presents a reasonable model of one way to produce an automated system to help improve the coverage of collaboratively built resources and discover new idioms. In addition, this research presents a novel approach to idiom detection within running text, positioning this task as a problem of WSD. As a by product of this formulation as a WSD task, this work presents a novel study of WSD in the context of Multi-Word Expressions (MWEs). Together, the components of this research present what is the most scalable approach to identification of idiomatic language to date. In addition, it supports the classification paradigm that a full idiom classification system must take into account the fact that phrases are idiomatic based on *usage* rather than necessarily *inherent* characteristics.

This formulation of idiom identification is a non-trivial task because it requires automatically modeling the semantics of both the phrase in question and it’s constituent words. While semantics of words are often obvious to humans, they are not simple to model for many reasons—e.g. the meaning of a word is often context dependent, two words may be closely related in meaning but appear far apart due to difference in word choice, and the models of the semantics of a word often depend on hand-built resources.

### 1.1 Definition of an Idiom

The term “idiom” is sometimes problematic because it is often not well-defined. The defining lines between other figures of speech such as metaphors, similes, and proverbs and idioms are
often confused. In this work, we use the following definition of an idiom from the *Oxford English Dictionary* to determine whether a phrase is idiomatic or not [7].

*idiom, n.*: A form of expression, grammatical construction, phrase, etc., used in a distinctive way in a particular language, dialect, or language variety; *spec.* a group of words established by usage as having a meaning not deducible from the meanings of the individual words.

Given this definition, an idiom is a phrase that breaks the principle of compositionality—the principle that the meaning of any phrase $p$ is determined by the meanings of the words that it is composed of. This is a broad definition that means that the set of idioms is a superset of other sets of figurative language like metaphors, similes, set phrases, and proverbs.

Since this research seeks to classify phrase senses as idiomatic or literal, literal language must also be defined. The simplest approach is the one that we take—we define everything that is not idiomatic as literal. Further notes on the distinctions made between literal and idiomatic phrases can be found in section 3.4.2.

### 1.2 Motivation

The motivation for idiom identification and the detection of figurative language in general resides in its applicability to larger NLP systems. Sag et al. [21] cite the non-conformance of idiomatic expressions to the literal meanings of the words that they are composed of as sources of errors in parsing, while Shutova et al. [9] cite them as one source of errors in automatic translation systems. These errors are due to the fact that idiomatic expressions frequently function differently than their literal counterparts. In many cases, these expressions should be treated as one unit rather than as separate words because they carry meaning distinct from those of their words and are often syntactically less flexible than other word sequences. Parsing in particular suffers when idiomatic expressions are split incorrectly, causing phrases to become separated when they shouldn’t be [21]. True understanding of idiomatic expressions seeks not only to understand the meaning of them, but also the sentiment that they carry, their other-language equivalents, and how syntactically-fixed each expression is.
1.3 Previous Work

Previous work in NLP in identifying figurative language has been limited in scope, often focusing on specific types of phrases whose constituent words have specific parts of speech (POS) and using relatively small amounts of data. The one exception to this is the work of Fothergill and Baldwin [11]—it is that which uses the largest data set previous to this work and doesn’t constrain itself to phrases with specific syntactic structures. While true comparison of these two models is impossible due to the use of different data sets, the model described in this research attains a higher accuracy with a larger gain over the majority classifier baseline.

With the notable exceptions of Katz and Giesbrecht [16] and Diab and Bhutada [6], previous work in idiom identification has focused only on the composition of the idiomatic phrase itself—it does not take into account the context surrounding the phrase. Such an approach misses idiomatic use cases wherein the phrase becomes idiomatic because of the context it occurs in, as shown in Table 1.1.

Wilks [25] defines metaphors according to violation of selectional restrictions in a given context. In this work, verbs that generally take one kind of argument—e.g. animate things—are seen as acting metaphorically when the objects that they take do not fall into the specified category (e.g. my car drinks gasoline). In so doing, he defined a selectional preference metric that has been used in subsequent research to identify metaphors.

The work of Gedigian et al. [13] used a maximum entropy classifier to classify metaphors from the Wall Street Journal corpus. This work specifically concentrated on frames of spatial motion, manipulation, and health. This work achieves a score of 95.12% correct, improving over a majority-classifier baseline by 5%, meaning that within these frames and this corpus more than 90% of the examples were metaphoric.

Birke and Sarkar [2] propose a clustering approach for identifying non-literal language. Their system, TroFi (Trope Finder), is based off the word-sense disambiguation algorithm of Karov and Edelman [15], and treats the identification problem as a disambiguation problem between the literal sense and the non-literal sense for a word. This work, like that of Gedigian et. al [13] restricts itself only to metaphors expressed by a verb.

Though the work of Fazly and Stevenson [10] deals with idioms in a more lexically-variable
way, they also restrict their system to work only with what they refer to as “verb+noun idiomatic combinations”—phrases in which there is a verb with a noun in its direct object position. Though this work does not restrict itself to phrases of length two as that of Shutova et. al, it restricts itself to only this category, and is therefore more limited in scope [8].

Fazly and Cook argue that most idioms are lexically fixed and thus measure the degree of flexibility of a given phrase to determine idiomaticity [4]. In their identification schema, they measure the degree of lexical, syntactic, and overall fixedness for a given phrase. This work tests each measure in isolation, performing classification based upon threshold-setting. In a dataset made of an equal number of idiomatic and literal phrases, their overall fixedness measure—a measure that takes a weighted average of lexical and syntactic fixedness—achieves an accuracy of 74%. This dataset is further restricted by the literal examples being chosen only “semi-randomly,” with all literal examples containing a verb that involves a physical act corresponding to its basic semantics. This work takes context into account to the extent that it uses context to determine the “canonical” form of a phrase.

Krishnakumaran and Zhu [17] propose a method for classifying sentences as “metaphoric” or “normal” using WordNet and bigram counts. They use hyponymy relations as annotated in WordNet and bigram counts to make their determination. This work is focuses on IS-A metaphors and is tested only on sentences gathered from the Master Metaphor List\(^1\), whereby they are not in fact distinguishing between “metaphoric” and “literal” but rather between less commonly used metaphors and more commonly used ones.

The work of Shutova et. al [8] is important because it suggests a powerful unsupervised approach to metaphor identification, but it nevertheless is restricted to two-word phrases that have a subject-verb (e.g. *campaign surged*) or a verb-direct object (e.g. *stir excitement*) relationship. This work uses the verb-clustering algorithms of Sun and Korhonen [23] which rely on selectional preference to perform clustering. This work achieves a precision of .79 and attains 7.69 times more coverage than the WordNet-based baseline.

The work that does take into account the context that a phrase occurs in, and thus considers the

---

\(^1\)The Master Metaphor List is a hand-compiled resource that aims to list metaphors in terms of source to target domain mappings [19] ("STATES ARE LOCATIONS": “He is in love.”, “He is at rest.”, etc…). Given that it is a hand-compiled resource, and according to Lakoff himself (estimating that it contains 20% of the material he had to compile [19]), this resource is incredibly sparse.
fact that a phrase can be both idiomatic and literal, depending on context, began with that of Katz and Giesbrecht [16]. In this work, a phrase’s context is modeled using Latent Semantic Analysis and its meaning vector is compared to the meaning vectors of each of its constituent words. Though this work is tested only on a small set of Preposition-Noun-Verb phrases, it describes ideas important to this work because it relies on breaches of the principle of compositionality to determine idiomaticity.

In their work, Diab and Bhutada [6] classify Verb-Noun Constructions (VNCs), producing a model that achieves an F-measure of 84.58% overall—89.96% for idiomaticity identification and 62.03% for literal identification. They perform supervised learning using a set of features based on a context window, character n-grams, and named-entity recognition, among others. They test on the data set of Cook et al. [5] and achieve state of the art results.

Fothergill and Baldwin [11] work on classifying Japanese MWEs as idiomatic or literal. They use a supervised machine-learning approach and use intuitions from WSD to develop the features they describe the phrases with. This work has the largest data set of any previous work with 23,392 MWE-tokens (associated with example sentences), and achieves an accuracy of .647 with a gain of 5.3 percentage points over the baseline. This work is the only previous work that has not restricted itself to a specific phrase-type.
Chapter 2

BACKGROUND INFORMATION

2.1 Perceptron Algorithm

In this work, we use the averaged perceptron algorithm, shown as algorithm 1, to perform supervised machine learning. This algorithm was chosen due to simplicity and transparency in terms of the weights that it assigns to each feature.

This algorithm is an online machine learning algorithm—it goes through the training examples one at a time and learns from each one. During learning, each training instance, already associated with a label $y_i$, is assigned a label by the algorithm $y^*$. If $y^* \neq y_i$, weight updates are recorded based on the difference between the feature vector for $(x_i, y_i)$ and $(x_i, y^*)$. If the classification for an example is correct, the weights are not updated. We used a version of the algorithm that averages updates and performs one update of the entire weight vector after each loop through all training instances, a technique that prevents large classification boundary swings with each training example and makes the algorithm more robust to noisy training data.

2.2 Word Sense Disambiguation

WSD is the problem of discerning which sense (dictionary definition) a given word refers to. Extending this problem to MWEs, this becomes the problem of discerning which sense a certain phrase refers to. In its most specific incarnation, this problem aims to point to exactly which definition a phrase refers to. Algorithms that have been important to single-word word sense disambiguation are the Lesk Algorithm [20], the extended Lesk algorithm ($elesk$) [1], and, showing more recent advances, algorithms that use latent semantics such as that of Guo and Diab ($wmfvec$) [14].
Algorithm 1 Averaged perceptron algorithm

**Input:**
\( D = \) set of training examples \( \{(x_i, y_i) : i \leq n\} \)
\( T = \) number of iterations

1: procedure \textsc{train}(\( D, T \))

2: \textbf{Initialize:}

3: All weights \( \theta_i \in \theta \leftarrow 0 \)

4: \hspace{1em} for \( t = 1 \rightarrow T \) do

5: \hspace{2em} All weights \( \theta_t \in \theta_t \leftarrow 0 \)

6: \hspace{2em} for \( i = 1 \rightarrow n \) do

7: \hspace{3em} \( y^* \leftarrow \arg \max_{y \in Y} \theta \cdot \phi(x_i, y) \)

8: \hspace{3em} if \( y^* \neq y_i \) then

9: \hspace{4em} \( \theta_t \leftarrow \theta_t + \phi(x_i, y_i) - \phi(x_i, y^*) \)

10: \hspace{3em} end if

11: \hspace{2em} end for

12: \hspace{1em} for \( \theta_t \in \theta_t \) do

13: \hspace{2em} \( \theta_i \leftarrow \frac{1}{n} \theta_t \)

14: \hspace{2em} end for

15: end for

16: end procedure

**Output:** weights \( \theta \)

\( \triangleright \) The length of \( \theta \) is the number of features times the number of classes \( y \).

\( \triangleright \) The function \( \phi \) is computed based on the features defined in section 3.3.
2.2.1 Single Word Sense Disambiguation

The Lesk algorithm concentrates on computing the overlap between the target word’s definition and the sentence that it occurs in. This algorithm computes overlap based only on exact string match—even with a good stemmer, it is easily susceptible to errors based on difference in word choice.

Algorithm 2 Simple Lesk algorithm

Input:

$w =$ word to be disambiguated
$s =$ sentence $w$ occurs in

1: procedure $\text{DISAMBIGUATE}(w, s)$
2: Initialize:
3: $\text{max\_overlap} \leftarrow 0$
4: $\text{best\_sense} \leftarrow \text{most frequent sense of } s$
5: $C_s \leftarrow \text{set of words in } s$
6: $D_w \leftarrow \text{set of senses associated with } w$
7: for $d \in D_w$ do
8: $\quad C_x \leftarrow \text{set of words in gloss of } x$
9: $\quad \text{overlap}_x \leftarrow |C_s \cap C_x|$
10: if $\text{overlap}_x > \text{max\_overlap}$ then
11: $\quad \text{max\_overlap} \leftarrow \text{overlap}_x$
12: $\quad \text{best\_sense} \leftarrow x$
13: end if
14: end for
15: end procedure

Output: Best sense for $w$ in $s$, $\text{best\_sense}$

The extended Lesk algorithm is based off of the Lesk algorithm, but seeks to make it more robust to differences in word choice by taking advantage of the structure of WordNet. It looks at the gloss (definition) of not just the target word, but the glosses of all words in the target word’s synset. In addition, it calculates overlap between the target word’s context and the concatenated glosses of the words in the synsets connected to the target word via relations such as hypernymy and meronymy, depending on the part of speech of the target word. This algorithm weights overlaps in a quadratic fashion—an overlap of length 2 is worth 4 whereas one of length 3 is worth 9. Banerjee and Pedersen [1] report an F-measure of .346 for $\text{elsea}$ versus .183 for original $\text{lesk}$. 
The recent work of Guo and Diab [14], published in 2012, seeks to learn the latent semantics of a concept from its definition and determine sense disambiguation based on these. Rather than taking into account only the words that do occur in the definition associated with the target word, this work also takes into account the words that don’t occur in the target word’s definition, taking these to be examples of things that the word is not associated with. This work also takes advantage of WordNet’s graph structure to build a more complete sense of the latent semantics of the target word. This work reports a gain of 2.9 – 4.5 points over basic elesk depending on the test data set [14].

2.2.2 Multi-Word Expression Sense Disambiguation

MWE sense disambiguation (MWESD) is a largely unexplored task. To the best of my knowledge, no algorithms have been developed specifically for MWESD prior to this work. In the context of idiomatic language identification, several approaches have drawn from the insights of WSD to perform identification [16] [6] [11], but none have directly performed MWESD.
Chapter 3

IDENTIFYING IDIOMATIC DEFINITIONS

3.1 Problem Definition

We assume data in form \{(p_i, d_i, y_i) : i \ldots n\} where \(p_i\) is the phrase associated with definition \(d_i\) and \(y_i \in \{\text{literal, idiomatic}\}\). Together, \(p_i, d_i\) form sense \(i\). The goal of this task is for each \(p_i, d_i\), assign class \(y^*\) where \(y^* \in \{\text{literal, idiomatic}\}\) to sense \(i\) such that \(y^* = y_i\). Our data were gathered from the English-language Wiktionary dump from November 13th, 2012, and are accessed using JWKTL [24]. Our set of \{(p_i, d_i) : i \ldots n\} was gathered according to the following restrictions: the title of the Wiktionary entry must be English, \(p_i\) must composed of two or more words \(w\) as defined by white space, and \(p_i, d_i\) must be in the “base form”—senses that are not defined as a different tense of a phrase—e.g. the sense that denotes \(p, d\) ( “weapons of mass destruction”, “Plural form of weapon of mass destruction” ) was removed while the pair ( “weapon of mass destruction”, “A chemical, biological, radiological, nuclear or other weapon that is designed to cause death or serious injury to large numbers of civilians.” ) was kept. From this data, a development set of 5% of the data and a test set of 10% of the data were created. Each pair \(p_i, d_i\) was assigned label \(y_i\) according to the rule: \(y_i = \text{idiomatic}\) iff “idiomatic” annotation \(\in d_i\) else \(y_i = \text{literal}\). In practice, this produces a noisy assignment of \(y\) because a majority of the idiomatic senses that occur in Wiktionary are not marked as such. Therefore we expect a large number of \(p, d\) to be incorrectly assigned \(y = \text{literal}\).

Table 3.1: Sense Information Breakdown for Data Sets

<table>
<thead>
<tr>
<th>Data Set</th>
<th>Literal Senses</th>
<th>Idiomatic Senses</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Base form, 2+ words, English (full data set)</td>
<td>56,037</td>
<td>7,764</td>
<td>63,801</td>
</tr>
<tr>
<td>Train</td>
<td>47,633</td>
<td>6,600</td>
<td>54,233</td>
</tr>
<tr>
<td>Development</td>
<td>2,801</td>
<td>388</td>
<td>3,189</td>
</tr>
<tr>
<td>Test</td>
<td>5,603</td>
<td>776</td>
<td>6,379</td>
</tr>
</tbody>
</table>
3.2 Experimental Setup

3.2.1 Iteration Optimization

The averaged perceptron algorithm, explained in 2.1, was optimized in terms of iterations $T$ by maximizing the F-measure obtained on the development set. For settings of $T$ from 1 to 100, the perceptron was trained with $T$ iterations, the resultant classification model was used to classify the development set, and the F-measure was obtained. The “optimal” setting for $T$ was then determined by selecting $T$ that is associated with the highest F-measure.

3.2.2 Resources

WordNet

WordNet is a lexical database of English words. Words are grouped into synsets based on synonymy relations and synsets are linked to one another via a variety of relations such as hypernymy, hyponymy, and many others. WordNet contains glosses of the words within it and links senses as well as word forms.

This work uses WordNet 3.0, accessed via the Java API for WordNet (JAWS)\(^1\), written by Brett Spell.

Wiktionary

Wiktionary is an extremely large, collaboratively authored dictionary. Each page within Wiktionary has a specific hierarchical structure. The structure that will be used to describe information from Wiktionary follows the terms used in Java Wiktionary Library (JWKTL) \([26]\). A page is the highest level unit, is roughly synonymous to webpage, and has a title corresponding to the word or phrase that the page describes. A page is said to contain entries, each one of which is associated with a particular language and a particular part of speech (POS). Each entry contains one or more senses, each sense corresponds to one particular definition of the title of the page when it is used as the POS defined in the entry that contains it. A sense may have example sentences or quotations associated with it.

\(^1\)http://lyle.smu.edu/~tspell/jaws/index.html
Contributers to Wiktionary have the option, but are not required, to annotate each definition with descriptors of the definition. These descriptors can describe the context that the definition is used in (e.g. baseball, computing, etc...), a finer-grain POS marker (transitive), and whether or not the definition is an idiomatic sense of the title, among other things.

3.3 Features

The features that were developed fall into two general categories: lexical and graph-based. Lexical features are features that inspect both the words in the phrase $p$ and in its definition $d$ and attempt to compute semantic relatedness from just this information. Graph-based features are calculated based on the graph structure of WordNet 3.0. Lexical features differ from the graph-based features because they do not utilize information on pages that are not the page on which the sense in question is located.

3.3.1 Lexical Features

The lexical features were motivated by the intuition that literal phrases are more likely to have closely related words to each word in $p$ in $d$. This is because literal phrases do not break the principle of compositionality and are thus more likely to have definitions that are lexically related. For all lexical features, when words are compared to each other, they are in their stemmed form, using the Porter Stemmer.

These features are computed using the following equations:

$$\text{count}(w, text) = \text{number of times } w \text{ appears in } text$$

$$\text{capitals}(text) = \frac{\text{number of capital letters in } text}{\text{number of words in } text}$$

- synonym overlap: For each word $w_i$ in $p$, for all synonyms of $w_i$ in Wiktionary forming set $S$ with members $\{s_i : i...n\}$, synonym overlap = $\frac{1}{|S|} \sum_{s_i \in S} \text{count}(w_i, d)$.

- antonym overlap: For each word $w_i$ in $p$, for all antonyms of $w_i$ in Wiktionary forming set $A$ with members $\{a_i : i...n\}$, antonym overlap = $\frac{1}{|A|} \sum_{a_i \in A} \text{count}(w_i, d)$. 

• average num capitals: The value of \( \text{capitals}(p) \).

### 3.3.2 Graph-Based Features

The graph-based features follow a similar intuition to the lexically-based features except that they used the graph structure of WordNet to calculate path distances.

These features are based on the following equations:

\[
\text{distance}(w, v, \text{rel}, n) = \begin{cases} 
\text{minimum distance via rel links from } w \text{ to } v & \text{if } w \text{ is within a distance of } n \text{ to } v \\
0 & \text{otherwise}
\end{cases}
\]

- closest synonym, WordNet: \( \min_{w_i \in p, v_j \in d} \text{distance}(w_i, v_j, \text{synonym}, 5) \)
- closest antonym, WordNet: \( \min_{w_i \in p, v_j \in d} \text{distance}(w_i, v_j, \text{antonym}^2, 5) \)
- closest hypernym, WordNet: \( \min_{w_i \in p, v_j \in d} \text{distance}(w_i, v_j, \text{hypernym}, 5) \)
- average synonym distance, WordNet: \( \frac{1}{|p|} \sum_{w_i \in p, v_j \in d} \text{distance}(w_i, v_j, \text{synonym}, 5) \)
- average hyponym, WordNet: \( \frac{1}{|p|} \sum_{w_i \in p, v_j \in d} \text{distance}(w_i, v_j, \text{hyponym}, 5) \)
- synsets connected by antonym, WordNet: This is a boolean feature that takes a value of 1 if for the set of synsets \( \text{Syn}_p \) that includes all synsets from all words in \( p \) and the set of synsets \( \text{Syn}_d \) that includes all synsets from all words in \( d \), any two synsets, one from \( \text{Syn}_p \) and one from \( \text{Syn}_d \), are connected by a shared antonym.

### 3.4 Models & Error Analysis

When evaluating the classification system, we used the development set to determine the optimal number of iterations and feature combination based on maximizing the F-measure of the system. Reported in Table 3.5 are the test results for each model with each feature set and for each model with a combined feature set. Precision versus recall curves are shown in graphs 3.1, 3.2, and 3.3.

\(^2\)The first relation expanded in the search was the antonym relation. All subsequent relations expanded were synonym relations.
3.4.1 The Basic Model

From inspecting the precision versus recall curve for this model, shown in graph 3.1, it becomes clear that even when the classifier is only classifying the \( \langle p, d \rangle \) pairs that it is the most confident are idiomatic as idiomatic, this model’s precision is unimpressive. Error analysis reveals that this model often classifies \( \langle p, d \rangle \) as idiomatic when it is labeled as literal but is in fact idiomatic. Such classifications, though correct, lower the precision values for the perceptron and cause the observed precision versus recall curve.

The top ten examples that the basic classifier model using complete training was most confident were idiomatic are shown in table 3.2, along with the labels that they are associated with from Wiktionary. Though only 30% of the top 10 \( \langle p, d \rangle \) pairs are labeled as idiomatic, 90% of them break the principle of compositionality and thus fall into our definition of idiomatic. This causes the scores that are reported by this model to be falsely low because the classifier is identifying senses as idiomatic that were not previously marked as such—a task that is in fact one of the goals of this research.

<table>
<thead>
<tr>
<th>#</th>
<th>( \langle p, d \rangle )</th>
<th>Wiktionary Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(live down, “To get used to something shameful.”)</td>
<td>literal</td>
</tr>
<tr>
<td>2</td>
<td>(feel free, “You have my permission.”)</td>
<td>literal</td>
</tr>
<tr>
<td>3</td>
<td>(cut one loose, “to fart”)</td>
<td>idiomatic</td>
</tr>
<tr>
<td>4</td>
<td>(fluff out, “To plump up”)</td>
<td>literal</td>
</tr>
<tr>
<td>5</td>
<td>(make after, “To chase.”)</td>
<td>literal</td>
</tr>
<tr>
<td>6</td>
<td>(nail down, “To say something with difficulty.”)</td>
<td>idiomatic</td>
</tr>
<tr>
<td>7</td>
<td>(blow out, “To extinguish something, especially a flame.”)</td>
<td>literal</td>
</tr>
<tr>
<td>8</td>
<td>(get out, “To say something with difficulty.”)</td>
<td>literal</td>
</tr>
<tr>
<td>9</td>
<td>(put through, “to cause to endure”)</td>
<td>idiomatic</td>
</tr>
<tr>
<td>10</td>
<td>(give off, “To emit.”)</td>
<td>literal</td>
</tr>
</tbody>
</table>

3.4.2 Correction and Annotation

To accurately assess the signal that the features we developed gave for whether or not a \( \langle p, d \rangle \) pair is idiomatic, we manually corrected the labels following the guidelines in appendix A.

For this work, due to time constraints, I was the sole annotator of all development and test
data.

When the data sets were corrected, as shown in table 3.3, it became obvious that there are 2–3 times as many idioms within Wiktionary than are marked as such.

3.4.3 The Corrected Model

The corrected model uses the same training data, but evaluates based off of the corrected data sets, described in table 3.3. This model yields higher performance metrics than the basic model because it is evaluating off of data that accurately reflects idiomaticity.

3.4.4 A Specifically Oriented Filter

From error analysis of the corrected model as well as from intuitions gained during annotation, a simple filter was built depending on the context annotations that each Wiktionary definition may carry. These intuitions were that definitions that are marked as within a specific context (computing, biology, cricket, etc...) are either: 1) strongly not idiomatic or 2) ambiguously idiomatic. Therefore, a filter was built by gathering the subject-specific markers present in the development data set. This list of subjects formed a simple filter to help “clean”—or target—the classification task to those phrases that we are most interested in.

<table>
<thead>
<tr>
<th>Data Set</th>
<th>Literal Senses</th>
<th>Idiomatic Senses</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Train</td>
<td>47,633</td>
<td>6,600</td>
<td>54,233</td>
</tr>
<tr>
<td>Development</td>
<td>2,801</td>
<td>388</td>
<td>3,189</td>
</tr>
<tr>
<td>corrected Development</td>
<td>2,213</td>
<td>957</td>
<td>3,170</td>
</tr>
<tr>
<td>corrected+cleaned</td>
<td>1501</td>
<td>885</td>
<td>2,386</td>
</tr>
<tr>
<td>Test</td>
<td>5,603</td>
<td>776</td>
<td>6,379</td>
</tr>
<tr>
<td>corrected Test</td>
<td>4,514</td>
<td>1,829</td>
<td>6,343</td>
</tr>
<tr>
<td>corrected+cleaned Test</td>
<td>3,262</td>
<td>1,725</td>
<td>4,987</td>
</tr>
</tbody>
</table>

The cleaned model uses the same training data as the previous two models, but evaluates off of the corrected+cleaned data sets described in table 3.3.
3.5 Results

We report results in terms of precision, recall, and F-measure for all models in terms of the classification of idiomatic examples. This means that $\text{precision} = \frac{|\{(p,d), y_{\text{idiotic}}\} classified correctly|}{|\{(p,d), y\} classified as idiomatic|}$, $\text{recall} = \frac{|\{(p,d), y_{\text{idiotic}}\} classified correctly|}{|\{(p,d), y_{\text{idiotic}}\}|}$, and F-measure is their harmonic mean.

3.5.1 Baselines

The baselines used for comparison in these experiments are two-fold: a majority baseline, and a proportional baseline. The majority baseline classifies all $(p, d)$ pairs as literal, and will, for all data sets, garner an F-measure of 0 because its recall will also be 0. The proportional baseline classifies according to the proportions of literal versus idiomatic that it learns from the training set, ultimately learning two probabilities $\text{prob}_{\text{literal}}$ and $\text{prob}_{\text{idiotic}}$ that sum to 1. When this model is given $(p, d)$ to classify, it selects a random number in the range $[0, 1]$, and if that number is $\leq \text{prob}_{\text{literal}}$, the pair is classified as literal, otherwise idiomatic.

<table>
<thead>
<tr>
<th>Model</th>
<th>Data Set</th>
<th>Recall</th>
<th>Precision</th>
<th>F-measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>proportional</td>
<td>basic</td>
<td>.1217</td>
<td>.1186</td>
<td>.1201</td>
</tr>
<tr>
<td></td>
<td>corrected</td>
<td>.2951</td>
<td>.1252</td>
<td>.1758</td>
</tr>
<tr>
<td></td>
<td>corrected+cleaned</td>
<td>.3721</td>
<td>.1281</td>
<td>.1906</td>
</tr>
</tbody>
</table>

3.5.2 Model Results

All models increased over their corresponding baselines by more than 22 points, and all achieved the highest F-measure when using the lexical+graph feature set. Though in the case of the corrected and cleaned models, the F-measure of the lexical feature set is within one point of the lexical+graph feature set, an inspection of the precision versus recall curves in section 3.5.3 reveals the necessity of including both sets of features for optimal behavior.
Table 3.5: Final Model Performance

<table>
<thead>
<tr>
<th>Model</th>
<th>Feature Set</th>
<th>Recall</th>
<th>Precision</th>
<th>F-measure</th>
<th>Iterations</th>
</tr>
</thead>
<tbody>
<tr>
<td>basic</td>
<td>lexical</td>
<td>0.8582</td>
<td>0.2192</td>
<td>0.3492</td>
<td>34</td>
</tr>
<tr>
<td></td>
<td>graph</td>
<td>0.4961</td>
<td>0.2964</td>
<td>0.3711</td>
<td>86</td>
</tr>
<tr>
<td></td>
<td>lexical+graph</td>
<td>0.6082</td>
<td>0.3124</td>
<td>0.4128</td>
<td>52</td>
</tr>
<tr>
<td>corrected</td>
<td>lexical</td>
<td>0.8119</td>
<td>0.4919</td>
<td>0.6126</td>
<td>34</td>
</tr>
<tr>
<td></td>
<td>graph</td>
<td>0.6282</td>
<td>0.5213</td>
<td>0.5698</td>
<td>83</td>
</tr>
<tr>
<td></td>
<td>lexical+graph</td>
<td>0.7452</td>
<td>0.5289</td>
<td>0.6187</td>
<td>96</td>
</tr>
<tr>
<td>cleaned</td>
<td>lexical</td>
<td>0.8191</td>
<td>0.5675</td>
<td>0.6705</td>
<td>34</td>
</tr>
<tr>
<td></td>
<td>graph</td>
<td>0.7443</td>
<td>0.5368</td>
<td>0.6238</td>
<td>50</td>
</tr>
<tr>
<td></td>
<td>lexical+graph</td>
<td>0.8487</td>
<td>0.5562</td>
<td>0.6720</td>
<td>81</td>
</tr>
</tbody>
</table>

Figure 3.1: Precision vs. Recall for basic Model Feature Groups

3.5.3 Precision versus Recall

Shown here are the resulting precision versus recall graphs for each model in the previous section. The graphs show the curve for each set of features as well as the combined set. These graphs display an important result; they reveal the importance of using both lexical and graph-based features despite the apparent similarity of F-measure shown in Table 3.5.

3.6 Conclusions

From these experiments, we can conclude that a supervised approach to differentiating between idiomatic and literal senses is effective. Even with training data that is highly noisy (an estimated
2/3 of idioms in Wiktionary are marked as literal), a classification model that uses a small amount of correctly-labelled data for optimization can perform very well. Combining this data with knowledge about subject-specific phrases leads to models that perform even better and that achieve optimal performance with similarly-shaped precision versus recall curves. These experiments prove not only that there are a significant amount of unlabeled idioms within Wiktionary, but that they can be identified with supervised machine learning algorithms. Further, this work expands the task of idiom identification beyond syntactically-specified types of phrases, determining idiomaticity from syntactically-generalizable measurements of semantic compositionality.

Though this work is unable to discover idiomatic phrases that are not within Wiktionary, it
is able to discover idiomatic phrases that are novel to Wiktionary in that they are not marked as such. To the best of my knowledge, these are the most generalized and scalable results that use the largest data sets to date.
Chapter 4

MULTI-WORD EXPRESSION SENSE DISAMBIGUATION FOR IDIOM DETECTION

4.1 Problem Definition

For this task, we assume data in the form \{\langle p_i, e_i, d_i \rangle : i \ldots n \} where \( p_i \) is the phrase associated with example sentence \( e_i \) and \( d_i \) is the definition or sense in which \( p_i \) is used in \( w_i \). The goal of this task is for each \( \langle p_i, e_i \rangle \), assign definition \( d^* \) from \( D_p \), the set of all definitions associated with phrase \( p_i \) to sense \( i \) such that \( d^* = d_i \). This task assumes that, in set \( D_p \), definitions \( d_{p,j} \in D \) have an ordering to them such that \( d_{p,1} \) is first and that \( \forall d_{p,j} : j > 1 \), \( d_{p,j} \) is directly before \( d_{p,j-1} \). We assume that \( d_{p,1} \) represents the most frequently occurring sense of \( p \). Our data were gathered from the english-language Wiktionary dump from November 13th, 2012, are accessed using JWKTL [24], and consist of the example sentences associated with any of the \( \langle p, d \rangle \) pairs in the full data set described in Table 3.1.

For the second component of this task, we assume data in the form \{\langle p_i, e_i, y_i \rangle : i \ldots n \} where all other variables are as previously described, and \( y_i \) is the label of the type of usage of \( p_i \) in \( e_i \) and \( \{y_i \in \{idiotic, literal\} : i \ldots n \} \). For this component, the goal of the task is for each \( \langle p_i, e_i \rangle \), assign label \( y^* \) where \( y^* \in \{idiotic, literal\} \) to sense \( i \) such that \( y^* = y_i \).

The development and test data sets for this task are the example sentences associated with the corrected development and test data sets described in Table 3.3.

<table>
<thead>
<tr>
<th>Data Set</th>
<th>Literal</th>
<th>Idiomatic</th>
<th>Total</th>
<th>Unique Senses</th>
<th>Unique Phrases</th>
</tr>
</thead>
<tbody>
<tr>
<td>full</td>
<td></td>
<td></td>
<td>10,871</td>
<td>9,296</td>
<td>7,106</td>
</tr>
<tr>
<td>dev</td>
<td>172</td>
<td>329</td>
<td>501</td>
<td>444</td>
<td>433</td>
</tr>
<tr>
<td>test</td>
<td>361</td>
<td>694</td>
<td>1,055</td>
<td>911</td>
<td>886</td>
</tr>
</tbody>
</table>
4.2 Multi-Word Expression Sense Disambiguation Results

The problem that this task aims to solve takes the traditional definition of word sense disambiguation and the algorithms described in section 2.2.1 and generalizes them to disambiguate MWEs. To this end, this work applies the Lesk algorithm, shown in 2.2.1, and the Extended Lesk algorithm to the context of MWEs. Both of these algorithms were extended in a naive manner in which the word was simply substituted for a phrase instead.

4.2.1 Baselines

This task utilizes three baselines, each with a slightly different strategy.

**baseline0** is defined by the set of equations:

\[
\text{disambiguate}(p, e) = \begin{cases} 
  d_{p,1} & \text{if } |D_p| = 1 \\
  \text{none} & \text{otherwise}
\end{cases}
\]

**baseline1** is defined by the equation:

\[
\text{disambiguate}(p, e) = d_{p,1}
\]

**baseline2** is defined by the set of equations:

\[
\text{disambiguate}(p, e) = \begin{cases} 
  d_{p,1} & \text{if } |D_p| = 1 \\
  d_{p,\text{random}} & \text{otherwise}
\end{cases}
\]

4.2.2 Results

Reported below are recall, precision, and F-measure for the MWESD task. In this context, \( \text{precision} = \frac{|(p,e) \text{ matched to correct } d|}{| (p,e) \text{ matched to any } d |} \), \( \text{recall} = \frac{|(p,e) \text{ matched to correct } d|}{| (p,e) |} \), and F-measure is their harmonic mean.

The results of the MWESD experiments reveal important characteristics of the data set. While the traditional WSD task evaluates on a data set in which the words to be disambiguated all have

\[ ^1 \text{All results reported for baseline2 are an average of 10 runs of the specified experiment. } \]
multiple senses, baseline0 reveals that 51.56% of \((p,e)\) pairs have only one possible definition \(d\) to match. This means that the scores for the baselines (in particular baseline1) are very high.

<table>
<thead>
<tr>
<th>Method</th>
<th>Recall</th>
<th>Precision</th>
<th>F-measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>baseline0</td>
<td>1.0</td>
<td>.5156</td>
<td>.6804</td>
</tr>
<tr>
<td>baseline1</td>
<td>.7128</td>
<td>.7128</td>
<td>.7128</td>
</tr>
<tr>
<td>baseline2</td>
<td>.6971</td>
<td>.6971</td>
<td>.6971</td>
</tr>
<tr>
<td>lesk</td>
<td>.7194</td>
<td>.7194</td>
<td>.7194</td>
</tr>
<tr>
<td>elesk</td>
<td>.7204</td>
<td>.7204</td>
<td><strong>.7204</strong></td>
</tr>
</tbody>
</table>

### 4.3 Idiom Detection Results

The extension of MWESD to idiom detection relies on a model that can differentiate between literal and idiomatic definitions of a phrase. For all \((p,e)\) pairs, they are first assigned a definition \(d\) by the designated MWESD algorithm. If no \(d_i\) is assigned for pair \(\langle p_i, e_i \rangle\), this pair is not classified as either literal or idiomatic. The pair \(\langle p_i, d_i \rangle\) is then assigned a class \(y_i\) by the definition classification model, which is in turn assigned to \(\langle p_i, e_i \rangle\) to form \(\langle (p_i, e_i), y_i \rangle\).

#### 4.3.1 Results

In this task, we will use a majority baseline as a baseline classification model. This baseline classifies all \(\langle p, e \rangle\) as idiomatic. The **corrected** and **cleaned** classification models are tested with each MWESD algorithm.

The precision, recall, and F-measure reported below are defined as they were for the definition classification task, except that recall is defined as: 

\[
\text{recall} = \frac{|\{(p,d),y_{idiomatic}\} \text{ classified correctly}|}{|\{(p,d),y_{idiomatic}\}| + |\{(p,e)\} \text{ unassigned to any definition } d|}
\]
Table 4.3: Idiom Detection Results

<table>
<thead>
<tr>
<th>MWESD Method</th>
<th>Classification Model</th>
<th>Recall</th>
<th>Precision</th>
<th>F-measure</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>none</td>
<td>majority</td>
<td>1</td>
<td>.6578</td>
<td>.7936</td>
<td>.6578</td>
</tr>
<tr>
<td>baseline0</td>
<td>corrected</td>
<td>.3175</td>
<td>.7829</td>
<td>.4518</td>
<td>.3697</td>
</tr>
<tr>
<td>baseline0</td>
<td>cleaned</td>
<td>.3366</td>
<td>.8829</td>
<td>.4874</td>
<td>.3839</td>
</tr>
<tr>
<td>baseline1</td>
<td>corrected</td>
<td>.7610</td>
<td>.7925</td>
<td>.7764</td>
<td>.7139</td>
</tr>
<tr>
<td>baseline1</td>
<td>cleaned</td>
<td>.7318</td>
<td>.8775</td>
<td>.7981</td>
<td>.7217</td>
</tr>
<tr>
<td>baseline2</td>
<td>corrected</td>
<td>.7687</td>
<td>.8019</td>
<td>.7849</td>
<td>.7169</td>
</tr>
<tr>
<td>baseline2</td>
<td>cleaned</td>
<td>.7463</td>
<td>.8862</td>
<td><strong>.8102</strong></td>
<td><strong>.7326</strong></td>
</tr>
<tr>
<td>leks</td>
<td>corrected</td>
<td>.7584</td>
<td>.7785</td>
<td>.7683</td>
<td>.7042</td>
</tr>
<tr>
<td>leks</td>
<td>cleaned</td>
<td>.7331</td>
<td>.8598</td>
<td>.7914</td>
<td>.7144</td>
</tr>
<tr>
<td>elesk</td>
<td>corrected</td>
<td>.7607</td>
<td>.7804</td>
<td>.7704</td>
<td>.7028</td>
</tr>
<tr>
<td>elesk</td>
<td>cleaned</td>
<td>.7360</td>
<td>.8683</td>
<td>.7967</td>
<td>.7168</td>
</tr>
</tbody>
</table>

The results from Table 4.3 reveal that even with naive MWESD algorithms, the overall detection system achieves precision that rivals that of the state of the art idiom identification systems. Furthermore, these results show that MWESD in combination with detection classification successfully determine whether or not a phrase in an example sentence is being used in an idiomatic sense. Though the results reported here are incomparable to results from other previous work due to completely different data sets, this research does prove that this classification pipeline is a viable one. The other previous work that has attempted to identify idiomatic language at a comparable scale—that of Fothergill and Baldwin [11]— achieved an accuracy of .647 while we achieved an accuracy of .7326 with a gain over the majority baseline of 7.48 percentage points.

### 4.4 System Limitations

Though this system is robust in many ways—it is blind to the syntactic structure of the phrases being identified and it is scalable because it draws from Wiktionary, an incredibly large resource—it is also limited by its design. For this model of idiom detection to be able to detect whether or not a phrase is idiomatic, it must have a definition within Wiktionary, which is not true for all phrases. For instance, the example "My car drinks gasoline." would at this stage be undetectable by the system that we have described. Therefore, we suggest that the optimal idiom detection system would be a combination of the one described here and others such that the strengths and weaknesses of the component systems are complementary.
4.5 Future & Continuing Work

To make this work and technique comparable to other idiom identification systems, it must be tested against the VNC data set; this is ongoing work. In addition, there is much work to be done in the context of MWESD algorithms. The ones described in this thesis are naive and do not take advantage of recently published research, such as that of Guo and Diab [14]. For a full investigation of how well traditional WSD algorithms work in the context of MWEs, not only must the methods based on latent semantics be tested, a more adequate data set must be developed. This data set should more closely align with those of single-word disambiguation data sets, insofar as the average proportion of senses to phrases should be more similar. Once these tasks are accomplished, future work can investigate new, specifically tailored MWESD algorithms.
Chapter 5

CONCLUSIONS

This work presents the first attempt at identifying definitions as idiomatic or literal senses of the titles that they are associated with, to the best of my knowledge. This task is then practically extended, presenting a novel approach for identifying idiomatic language, an approach that is shown to be an effective method for the task. In the experiments conducted, we demonstrated that the definition classification model benefits from both lexical and graph-based features, that it is robust to learning from noisy training data, and that it is a competent solution for classifying such data at scale, achieving an F-measure of 67.20%. Using this model and simple WSD algorithms, we are able to achieve high precision idiom identification, demonstrating the validity of the system developed.

Future work will be in integrating bootstrapping into the definition classification task and comparing the resultant models with the ones described in this thesis. In addition, the MWESD algorithms for idiom detection will be further investigated by testing this system on the VNC data set, which will enable the results of this task to be directly comparable to the current state of the art systems. This work will be exposed to the research community by releasing the new data sets gathered from this work—the annotated data sets for definition classification task as well as the idiom detection task—to the public.
BIBLIOGRAPHY


Appendix A
ANNOTATION GUIDELINES

When annotating whether or not a \( \langle p, d \rangle \) is idiomatic or not, follow the following guidelines:

\[ idiom, \ n.: \ A \ form \ of \ expression, \ grammatical \ construction, \ phrase, \ etc., \ used \ in \ a \ distinctive \ way \ in \ a \ particular \ language, \ dialect, \ or \ language \ variety; \ spec. \ a \ group \ of \ words \ established \ by \ usage \ as \ having \ a \ meaning \ not \ deducible \ from \ the \ meanings \ of \ the \ individual \ words. \]

- If \( \langle p, d \rangle \) is already marked as idiomatic, it does not need to be corrected.

- If \( d \) cannot be discerned through the semantic composition of each word in \( p \), mark it as idiomatic, otherwise, mark it as literal.

  - When considering semantic composition, be extra sensitive to the meanings that verbs and prepositions take. Phrases are often both idiomatic and commonplace, which can fool you into thinking that phrases like \( \langle \text{make after, “To chase”} \rangle \) are literal.
  - If you do not know the definition of a word in the phrase, look it up and judge from the definitions that are listed.

- Finer-grained categories of phrases that are often idiomatic: slang, similes, metaphors, proverbs, euphemisms, and rhetorical questions.

- Finer-grained categories of phrases that are often literal: geographic locations, names of plants, animals, proteins, etc... (that are not slang).

- If unable to discern idiomaticity after following the above steps and the guidelines below, mark as unknown.

- If the gloss \( (d) \) of \( p \) consists only of punctuation, mark as unknown.
All \((p, d)\) pairs are marked as belonging to one of three categories: idiomatic, literal, or unknown.

Enumerated below are some commonly occurring cases that can seem ambiguous. Even when a marking is indicated below, this is just a guideline, always use your judgement, and, if unsure, mark as unknown.

1. Nonsense phrases:
   examples:
   \(\langle \text{tut tut, “The sound of such an exclamation of disapproval.”} \rangle\)
   marking: literal

2. Phrases specific to a certain area (a certain sport, legalese, etc...) (most commonly occurring):
   examples:
   \(\langle \text{wrist spinner, “A bowler who specializes in wrist spin”} \rangle\)
   \(\langle \text{off drive, “a type of batsman”s shot played by swinging the bat vertically and hitting the ball along the ground in the direction of mid off”} \rangle\)
   marking: literal (see exception below)

3. Terms that are specific to a certain area (sports, etc.), but that are phrases that have a different meaning in “real life” or have a meaning beyond the words in the phrase.
   examples:
   \(\langle \text{punch out, “To rule (by an umpire) that a pitch is a called third strike, often done emphatically.”} \rangle\)
   \(\langle \text{daisy chain, “(electrical engineering) Placing several electrical or electronic devices in parallel by either their power connections or their data connections or both.”} \rangle\)
   \(\langle \text{yellow jersey, “The yellow-colored shirt worn by the leader of a cycling race at a certain time. It is used in the Tour de France”} \rangle\)
   \(\langle \text{two fat ladies, “A traditional call for the number eighty-eight.”} \rangle\)
   marking: idiomatic

4. Phrases that are used in English but that are not English:
   examples:
   \(\langle \text{chacun à son goût: “Used to acknowledge that different people have different tastes or preferences.”} \rangle\)
   \(\langle \text{coup de grâce, “A final blow or shot given to kill a wounded person or animal.”} \rangle\)
   \(\langle \text{bona fides, “sincerity, good faith”} \rangle\)
   marking: literal

5. Different forms of phrases:
   examples:
   \(\langle \text{wild goose chase, “Alternative spelling of wild-goose chase”} \rangle\)
   \(\langle \text{follicularly challenged, “A common, but etymologically incorrect, form of follicly challenged.”} \rangle\)
marking: mark according to whether the base form is literal or idiomatic

6. Nicknames:
   examples:
   ⟨ Mickey D, “Nickname for McDonald”s, chain of fast-food restaurants.” ⟩
   marking: literal

7. Expression phrases:
   examples:
   ⟨ horse hockey, “An expression of disbelief or disgust.” ⟩
   marking: idiomatic

8. Cockney rhyming slang:
   examples:
   ⟨ Plates of meat, “The feet” ⟩
   marking: idiomatic

9. Time expressions:
   examples:
   ⟨ twenty to, “twenty minutes before the next hour” ⟩
   ⟨ quarter past, “Fifteen minutes past any hour.” ⟩
   marking: idiomatic (except when they are either nonsense phrases or not in English)

10. “Proper” nouns:
    examples:
    ⟨ Lady Bountiful, “An over-patronising woman, showing off her wealth by acts of overwhelming generosity.” ⟩
    ⟨ Dear John letter, “A letter from a wife or girlfriend to her husband or boyfriend, informing that the relationship is over. Usually precipitated by the writer”s having met a third party.” ⟩
    ⟨ Typhoid Mary, “A person who carries and spreads a dangerous disease, especially one who refuses to cooperate to minimize the risk of infection.” ⟩
    marking: idiomatic

11. Phrases that are highly colloquial:
    examples:
    ⟨ give it some welly, “To increase fuel or power to an engine, as to a car by depressing the gas pedal.” ⟩
    ⟨ red biddy, “A cheap alcoholic drink made from mixing red wine and methanol.” ⟩
    ⟨ ticky tacky “It may also refer to tar paper.” ⟩
    marking: idiomatic