A Bigger Fish to Fry

Haagsma, Hessel

DOI:
10.33612/diss.131057087

A Bigger Fish to Fry

Scaling up the Automatic Understanding of Idiomatic Expressions

Hessel Haagsma

university of groningen

CLCG
Center for Language and Cognition Groningen

The work in this thesis was carried out under the auspices of the Center for Language and Cognition Groningen (CLCG) of the Faculty of Arts of the University of Groningen.

Groningen Dissertations in Linguistics 182
ISSN: 0928-0030
ISBN: 978-94-034-2526-9 (printed version)
ISBN: 978-94-034-2525-2 (electronic version)

Document prepared with LaTeX 2$_{\varepsilon}$ and typeset by pdfTeX (Erewhon and Raleway fonts)
Cover: Jaques' Illustrated Proverbs, 1870 & 1885.
Printed by Ipskamp Printing on 115g G-print paper.

rijksuniversiteit
groningen

A Bigger Fish to Fry

Scaling up the Automatic Understanding of Idiomatic Expressions

Proefschrift

ter verkrijging van de graad van doctor aan de
Rijksuniversiteit Groningen
op gezag van de
rector magnificus prof. dr. C. Wijmenga
en volgens besluit van het College voor Promoties.

De openbare verdediging zal plaatsvinden op

donderdag 3 september 2020 om 16.15 uur

door

Hessel Haagsma

geboren op 6 maart 1992
te Heerenveen

Promotores
Prof. dr. ing. Johan Bos
Prof. dr. Malvina Nissim

Beoordelingscommissie
Prof. dr. Petra Hendriks
Prof. dr. Laura Kallmeyer
Prof. dr. Caroline Sporleder

Acknowledgements

It's been 4½ years since I started this PhD-trajectory, not knowing what I was getting myself into. I still don't know, but I do know it's finished! So, it is time to say thanks to those people who got me to the end.

First and foremost, I'm grateful to my supervisors, Johan and Malvina. Johan, thanks for giving me the opportunity to be part of a very cool research project, but also to find my own research interests. Malvina, thanks for your optimism and always coming up with new ideas and side-projects. Thanks to the both of you for your kind-but-honest criticism, for being patient with me during these 4½ years, and, on the less scientific side, for the garden parties and table football games!

Further thanks go out to the members of my reading committee: Petra Hendriks, Laura Kallmeyer, and Caroline Sporleder, for finding the time to read this whole book (and approving it, of course!).

Work is just work, but what makes it enjoyable is good company. Luckily, the Alfa-Informatica department (a.k.a. Informatiekunde, Computational Linguistics, Information Science – it's complicated) is full of good company. I really enjoyed commiserating, drinking and pub quizzing with my fellow PhDs, post-docs, interns, and other assorted office mates: Ahmet, Anna$_1$, Anna$_2$, Chunliu, Dieke, Duy, Fabrizio, Gosse, Johannes, Kilian, Lasha, Lukas, Martijn, Masha, Pauline, Pierre, Prajit, Rik, Rob, Steven, Stéphan, and Teja. Rik and Masha deserve a special mention for not just being colleagues, but agreeing to dress up all fancy and be my paranymphs. Of course, not to forget the rest of the department: Andreas, Antonio, Arianna, Barbara, Gertjan, Gosse, Gregory, Johan, Leonie, Malvina, Martijn, and Tommaso, thanks to all of you for the lunches, uitjes and endless reading group discussions, providing structure and perspective in the empty open sea that is a PhD-project sometimes.

At times, it's good to be reminded that there is life outside the Groningen science bubble. Thanks to my family for doing just that, Heit en Mem, Femke, Pieter, Jurre en Mette, Pake en Oate, *dankewol*! Finally, Inge, thanks for being there for me on good days and bad days, and for not allowing me to give up.

Groningen, March 2020

Contents

CHAPTER 1

Introduction

Louis van Gaal is a former Dutch football manager who coached Manchester United from 2014 to 2016. A native Dutch speaker, he spoke to the press in English, his second language. During his stint as manager, he coined many previously unheard phrases in English, some of which have since become common parlance in English football media. The most (in)famous of these stem from Van Gaal's predilection towards using set phrases, even going so far as to translate them into English directly from Dutch. Some examples of these are presented below (idioms marked in bold):

(1) *Ja, ehh…* It's **again the same song**. We have created a lot of chances, but we don't finish these chances.

(2) Now we have to play against Chelsea. In the Netherlands they say: '**that's another cook**'.

(3) Of course, we were also unlucky, because they score out of our errors. At once. And then you are always **running behind the facts**.

To monolingual English speakers, these phrases may pose a problem, even though they would probably be able to understand them when encountered in context. This problem is much smaller in other sentences by Van Gaal in which he does not use set phrases, such as Example 4.

(4) But, when you see overall, the long ball, and what is the percentage of that, West Ham United have played 71% of the long balls to the forwards and we 49.

So, what makes Examples 1–3 much more challenging? The answer is straightforward: all three phrases are *idiomatic expressions*. Or, to put it more precisely, they are Dutch idiomatic expressions. Their Dutch equivalents are, in order, *weer hetzelfde liedje* 'the same thing again and again', *dat is andere koek* 'a completely different matter', and *achter de feiten aanlopen* 'to lag behind events'.

Idiomatic expressions are particularly troublesome, since one of their main characteristics is that their meaning does not follow directly from the combination of the meaning of its component words. As such, translating it word-by-word does not generate the same meaning in the target language.[1] So, the Dutch expression *weer hetzelfde liedje*, when translated word for word, becomes 'again the same song', which does not elicit the meaning 'the same thing again and again'. The main reason for this is the word *liedje*, which means 'song', even though the meaning of the phrase is unrelated to any kind of song or music.

Although idioms have other distinctive characteristics, their non-compositional meaning is what poses most problems for non-native speakers of a language. Similarly, this makes idioms problematic for computers dealing with language, which is more commonly known as natural language processing (NLP). In this thesis, we are concerned with exactly this topic, namely how the handling of idiomatic expressions within NLP should be approached.

In recent years, great progress has been made in the quality of NLP systems, both in accuracy and practical applicability, mainly due to the surge of deep neural network methods. Generally, mainstream text in

[1] Unless of course, the target language happens to have the same idiomatic expression, as with Dutch *ergens de vinger op leggen* and English *put one's finger on something*.

major languages can now be processed reliably, meaning that it is time for research to move on to more challenging topics. This includes non-canonical domains, such as social media text, under-resourced and minority languages, and challenging language phenomena like sarcasm, metaphor and idiom. Due to their relative scarcity, idioms might seem a marginal area for research, but they do in fact pose a significant problem for a wide range of applications in natural language processing (Sag et al., 2002). These include machine translation (Salton et al., 2014a; Isabelle et al., 2017; Fadaee et al., 2018), semantic parsing (Fischer and Keil, 1996), and sentiment analysis (Williams et al., 2015; Liu et al., 2017; Spasić et al., 2017; Hwang and Hidey, 2019).

In addition to directly NLP-related applications, better processing of idioms can also benefit other areas of linguistics. For example, Liu and Hwa (2016) explore the possibility of automatically replacing idioms by literal paraphrases of their meaning, in order to aid language learners in understanding the text. Another example is the work by Liu et al. (2019), who, instead of reading, focus on writing, by building a system which automatically recommends idioms to use in the writing of Chinese essays. Finally, it can benefit the overall understanding of idioms, since better automatic processing facilitates large-scale corpus-linguistic investigations. These, in turn, can provide evidence regarding hypotheses about the usage, distribution, and behaviour of idioms.

Chapter Guide

In this thesis, we aim to improve the automatic processing of idioms in two main ways. First, collect a large number of idiom instances to get a more representative picture, which in turn can inform additional idiom processing models. Second, we come up with models which can detect the meaning of idiom instances in text in a general way, dealing well with both unseen and seen expressions. This work contains six content chapters, organised in three parts, dealing with the following four research questions:

RQ 1 What constitutes a potential idiom extraction system, and how can it be evaluated?

RQ 2 To what extent do automatic pre-extraction and crowdsourced annotation facilitate the construction of a large-scale idiom corpus?

RQ 3 Can unsupervised idiom disambiguation methods, enriched with additional information, rival supervised methods' performance?

RQ 4 Do deep neural network methods provide the same performance improvements for idiom processing as for the processing of non-idiomatic text?

Part I - Background

Part I provides an overview of existing work on idiomatic expressions, both from corpus linguistics and NLP perspectives. This provides a primer on idiomatic expressions as a linguistic phenomenon and the background for our work on the automatic processing of idioms. Chapter 2 provides an overview of corpus-linguistic insights on idiom, covering their overall frequency, form variation, and cross-genre distribution. Idioms are discussed from a different angle in Chapter 3, which discusses existing datasets, tasks, and approaches for idioms within NLP.

Part II - Corpus Construction

In Part II, we focus on building a large corpus of potentially idiomatic expressions with sense annotations, with the end goal of enabling the testing of hypotheses about the distribution of idioms, the training of data-hungry idiom disambiguation models, and more fine-grained evaluation of such models. Chapter 4 describes the development of a wide-coverage extraction system of potentially idiomatic expressions and the building of a small corpus to evaluate that system, providing an answer to **RQ 1**. Chapter 5 describes the building of a large corpus of idioms using crowdsourced annotation, focusing on the challenges involved in crowdsourcing and analysing the contents of the corpus. This, combined with Chapter 4 helps to answer **RQ 2**.

Part III - PIE Disambiguation

Naturally, Part III, containing the remaining content chapters, then serves to answer **RQ 3** and **RQ 4**. In Chapter 6, we discuss the benefits of unsupervised methods for idiom disambiguation and extend an existing unsupervised method based on lexical cohesion to improve its performance to rival that of supervised methods. However, a performance gap remains, so we focus on supervised methods in Chapter 7. There, we explore the novel use of deep learning approaches (LSTMs) for idiom disambiguation, which is made possible by the size of the corpus developed in Chapter 5.

Part IV - Conclusions

In the last chapter, Chapter 8, we provide an overview of the conclusions drawn from this work and answer the research questions posed above, based on those conclusions. Finally, we suggest directions for future research on idiomatic expressions in NLP.

PART I

Background

do your homework examine thoroughly the details and background of a subject or topic, especially before giving your own views on it. ● *The speaker had certainly done his homework before delivering the lecture.* ● *The PhD-candidate hadn't done his homework before the defence.*

CHAPTER 2

Idioms in Text

Abstract|Idiomatic expressions are an under-researched topic within natural language processing (NLP), but have been widely studied in corpus linguistics. In this work, we are mostly concerned with the computational processing of idiomatic expressions, but first we establish the groundwork for further investigating of this phenomenon. We provide an overview of corpus-linguistic insights on idiom, focusing on their frequency in text, their distribution across genres, the occurrence of idioms' literal equivalents, and surface form variability.

We find that there is much disagreement on what qualifies as an idiomatic expression, but that at the same time, there is a consistent core of idiom characteristics which are widely agreed upon. As for their occurrence in text, the available evidence is consistent with the notion that 'idioms are rare individually, but frequent as a group'. Finally, we examine the amount of variation idioms display in their surface form and find various suggestions as to what enables an idiom to exhibit certain kinds of variations.

2.1 Introduction

Idiomatic expressions are a fascinating linguistic phenomenon, and have attracted much attention in linguistics. This stems mainly from their idiosyncratic nature: their meaning is partly fixed in the lexicon, but also partly compositional. This position on the edge between lexical meaning and phrasal syntax makes them a fruitful and challenging object for study.

In this work, we deal with the computational processing of idiomatic expressions, which builds on useful information from non-computational linguistic observations. For example, knowing about how much idioms vary in their form and how this relates to their meaning is very useful as an indicator for effective idiom disambiguation features (Chapter 7). Similarly, knowledge about the frequency and distribution of idioms in corpora will benefit the process of building an annotated corpus of idioms (Chapter 5).

In this chapter, we look at existing research, mainly from corpus linguistics. We aim to set up a consistent and concise working definition of what constitutes an idiomatic expression (Section 2.2). We will also explore the distribution of idiom in various kinds of language resources, specifically the effect of genre and text type (Section 2.3). Finally, the variability of idioms is a major topic of interest (Section 2.4).

2.2 Definition of Idiom

Many different definitions of what is and is not encompassed by the term ‘idiom’ are used by researchers. Here, we do not intend to arrive at a definitive, perfect definition, but rather to arrive at some concise, clear and usable set of defining characteristics. Whatever the definition, unclear borderline cases will inevitably exist. Based on definitions used in previous work, we hope to identify some properties which can be used to practically delineate the boundaries of idiom. Listed below are some (condensed) example definitions used in previous research:

Nunberg et al. (1994) *"Attempts to provide categorical, single-criterion definitions of idioms are always to some degree misleading and after the fact." "[..] idioms occupy a region in a multidimensional lexical space, characterized by a number of distinct properties: semantic, syntactic, poetical, discursive, and rhetorical." "[..] the meaning of an idiom cannot be predicted on the basis of a knowledge of the rules that determine the meaning or use of its parts when they occur in isolation from one another. For any given collocation, of course, conventionality is a matter of degree, and will depend among other things on how we interpret 'meaning' and 'predictability'."*

Fernando (1996) *"The three most frequently mentioned features of idioms: 1. Compositeness: idioms are commonly accepted as a type of multiword expression. 2. Institutionalization — idioms are conventionalized expressions, conventionalization being the end result of initially ad hoc, and in this sense, novel, expressions. 3. Semantic opacity — the meaning of an idiom is not the sum of its constituents."*

McCarthy (1998) *"[..] we used the word 'idiom' to mean strings of more than one word whose syntactic, lexical and phonological form is to a greater or lesser degree fixed and whose semantics and pragmatic functions are opaque and specialised, also to a greater or lesser degree."*

Moon (1998) *"[..] there is no unified phenomenon to describe but rather a complex of features that interact in various, often untidy, ways and represent a broad continuum between non-compositional (or idiomatic) and compositional groups of words."*

Simpson and Mendis (2003) *"The most prevalent description of an idiom is a group of words that occur in a more or less fixed phrase and*

> *whose overall meaning cannot be predicted by analyzing the meanings of its constituent parts. Starting from the premise that an idiom is a multiword expression, we used three criteria: compositeness or fixedness, institutionalization, and semantic opacity. Compositeness or fixedness means that the individual lexical units of these expressions are usually set and cannot easily be replaced or substituted for. Institutionalization refers to the conventionalization of what was initially an ad hoc, novel expression. Semantic opacity indicates that the meaning of such expressions is not transparent based on the sum of their constituent parts."*

Although these definitions emphasise different aspects, there is a lot of common ground between them. Firstly, an idiom is always a multiword expression (MWE), i.e. two or more words[1] which are in some way related, and often occur in a sequence. This is a seemingly trivial but not unimportant criterion, following Fernando (1996).

Secondly, there are the three main characteristics which we can use to distinguish idiomatic expressions from other MWEs. In order to be an idiom, the expression should be *conventionalised* (or institutionalised), *semantically non-compositional* (or figurative or opaque), and show *fixedness* (or be inflexible or composite).

There is general agreement on these criteria, but the crux of the matter is in how to define and delimit these characteristics. This sentiment is also expressed by McCarthy (1998): "*The cut-off point where fixed expressions become open, freshly synthesised lexico-grammatical configurations [..] and where opaque idiomatic meaning becomes transparent and more and more literal is problematic and ultimately impossible to pinpoint. [..] Ultimately, intuition also has to play a role, especially in borderline cases*".

[1] The term 'word' is used loosely here. In practice, we classify something as an MWE if it is written as more than one word in its dictionary form. However, in practice, the line between a single word and an MWE is not always clear. For example, many multiword idioms, like *tongue in cheek*, are sometimes written as a single word with dashes, as in 'Corbett loved the brilliant logic delivered so tongue-in-cheek [..]'.

Nevertheless, broadly speaking, the three criteria can be defined as follows. An idiom is conventionalised when it is recognisable as conventional and/or familiar by a large proportion of native speakers. Semantic non-compositionality means that the meaning of the idiom differs from the meaning arrived at by combining the meanings of its components in the regular way. Fixedness refers both to the lexical and syntactic aspects of the idiom, meaning that not all possible replacements of component words by synonyms still yield the same idiomatic expression and that not all syntactic transformations of the expression still allow an idiomatic reading.

Nunberg et al. (1994) identifies three more characteristics, which are more secondary in nature: informality, affect, and proverbiality. These are not necessarily useful to determine whether a given MWE is an idiom, but are common aspects of idioms. Following Nunberg et al. (1994), informality means that idioms are associated with relatively informal registers of language, affect means that idioms are usually used to express some kind of affect or emotion, and proverbiality means that idioms are typically used in reference to a recurrent situation of particular social interest, i.e. something non-mundane.

Thus, to summarise, an idiom is: *a conventionalised multiword expression, which is to some extent lexically fixed and semantically non-compositional.*

2.3 Distribution of Idioms

Following the question of what is and what is not an idiom, we are interested in another basic property of idioms: where, why, and how often are they used? We look into the frequency and distribution of idioms in text overall, and whether this varies by genre. Finally, we also consider how frequent literal equivalents of idiomatic phrases are, that is, the usage of *small potatoes* to refer to actual potatoes of small size, relative to the idiomatic usage of these phrases.

Assessing the distribution of idioms poses various challenges. First, idioms are relatively rare, meaning that one has to comb through a large corpus to get a large enough number of idiom instances to draw any conclusions about their distribution. This is emphasised by Minugh (2008), who states that "*It is also clear that, given the relative scarcity of individual idioms, unusually large samples are necessary [..]*". Second, there is no clearly delineated set of 'all idioms in the English language', so idiom extraction involves either selecting a subset of idioms to look at, or manually reading through the text and extracting anything that fits the definition of an idiom (also see Section 4.2). The first has the drawback of potentially introducing bias in the selection of idioms, while the latter is highly time-consuming and prone to disagreement between annotators. Finally, if one also wants to include literal uses of idioms, the workload and complexity of the task further increases.

Likely because of the amount of effort involved, there are only two examples of idiom extraction which do not rely on a subset of idiomatic expressions: Simpson and Mendis (2003) and Street et al. (2010). Simpson and Mendis studied the distribution of idioms in the MICASE corpus of American academic speech (1.7M tokens, Simpson et al., 1999). They started by manually annotating all idioms in one half of the corpus, and then extracting the same idioms from the other half of the corpus. In a similar approach, Street et al. manually annotated idioms in a 69K word subset of the American National Corpus (Reppen et al., 2005), limiting themselves to idioms of certain syntactic subtypes. However, a significant proportion of what they annotate as idioms are actually non-idiomatic multiword expressions, such as *work toward (something)* and *on the downside.*

The alternative approach, using a pre-selected list of idioms, has been used more often. Even though these include only a subset of idiomatic expressions, we can use the assumption that the average frequency for a well-chosen subset of idiom types approximates the average for the complete set, in the same corpus. This approach has been used by Cook

Study	Tokens	Types	Instances	ITM
Simpson and Mendis (2003)	1.7M	238	562	1.39
Street et al. (2010)	0.07M	135	154	16.3
Cook et al. (2008)	96.8M	53	2,984	0.58
Minugh (2008)	3.7M	3,485	5,439	0.42
Sporleder and Li (2009)	1,756.5M	17	3,964	0.13
Sporleder et al. (2010)	96.8M	52	3,703	0.73

Table 2.1: Statistics from various idiom extraction studies, and the average number of instances per idiom type per million words (ITM).

et al. (2008), Minugh (2008), Sporleder and Li (2009), and Sporleder et al. (2010). An overview of the counts and frequencies found in these studies is provided in Table 2.1.

There are large differences between the studies, both in idiom selection and corpus size, from the several thousand idioms and the modest 3.7M token COLL Corpus (Minugh, 2002) used by Minugh, to the 17 idiom types and 1,756.5M token Gigaword Corpus (Graff and Cieri, 2003) used by Sporleder and Li. Despite the differences in corpus size, idiom set, and extraction method, the average frequencies form a relatively consistent band: between 0.1 and 1 instances per idiom type per million words.

Similar figures are found by Moon (1998) and Liu (2003). They do similar work, but only report the distribution of idiomatic expressions across frequency bands, rather than exact frequencies. Still, Liu (2003) reports that only 3% of idiomatic expressions occur more than 2 times per million words, and Moon (1998) show that most idioms have a frequency of between 0.1 and 1 per million words, which implies an average in line with other findings.

The two unrestricted approaches do not fit in this frequency band, but there are some considerations to be made in those cases. Clearly, the number reported by Street et al. (2010) is inflated by both their broad definition of idiom, and the very small size of the corpus, and cannot be relied on because of that. However, the frequency found by Simpson and Mendis (2003), in a more robust study, is also higher. The explanation for this is straightforward: in the unrestricted approach, the 'set' of idioms used is, by definition, limited to idioms which occur at least once. In the idiom subset approach, the pre-defined idiom set can also contain idioms which do not occur in the corpus, which lowers the average frequency. To illustrate this, we see that the idiom frequency found by Minugh (2008) is much closer to that of Simpson and Mendis (2003) if we exclude the idioms which did not occur in the corpus (2,063 of 3,485 idioms). Then, the frequency increases from 0.42 to 1.03, much closer to Simpson and Mendis's 1.39.

In conclusion, these numbers paint a somewhat paradoxical picture, which can be summarised as '*idioms are rare individually, but frequent as a group*'. On the one hand, idioms are very rare, with an average idiom occurring less than once in a million-word corpus. On the other hand, idioms are surprisingly frequent. Assuming an idiom inventory of approximately 5,000 types and an ITM value of 0.50, in between Cook et al. (2008) and Minugh (2008), there would be 2,500 idiom instances in a million-word corpus, i.e. one idiom per 400 tokens. Moreover, assuming an average of three component words per idiom, approximately 1 in every 133 tokens is part of an idiomatic expression.

2.3.1 Distribution across Text Types

It is widely assumed that the distribution of idioms in different texts is highly variable. Several influences have been suggested: domain (Street et al., 2010), genre (Minugh, 1999, 2008), register (Minugh, 1999; Liu, 2003), language variety (Fernando, 1996; Minugh, 2008), discourse mode (Simpson and Mendis, 2003), age (Minugh, 2008), and authority of the writer (Minugh, 2008).

Here, we look in more detail at the evidence there is in existing work for each of these suggestions. Street et al. (2010) compare idiom frequencies in written fiction, written non-fiction, and spoken language. They find that idiom is more frequent in fiction than in the other two genres, but only for verb-noun constructions. For prepositional phrase-type idioms, the opposite is true. However, Street et al.'s study has significant drawbacks regarding sample size and the definition of idiom, so not too much stock can be put in these findings.

Minugh (2008) studies idiom in a corpus of college newspapers. He compares genres and language varieties, and finds no clear effect of language variety or geographical location on idiom frequency. For genre, however, he finds that there are clear differences, and that the genres with the highest frequency are those in which the writer has the most 'authority' (e.g. editorials), which he links to the idea of idioms being used to convey 'received wisdom'.

Simpson and Mendis (2003), in turn, look at the effect of discourse mode (monologue, interactive, or mixed) and domain (e.g. humanities or social sciences). For both factors, they found no clear effects, despite their expectations that idiom would be more frequent in interactive discussions and 'soft' sciences than in monologues and 'hard' sciences. Rather, they conclude that "*[..] the use of idioms seems to be a feature more of individual speakers' idiolects than of any linguistic or content-related categories.*".

In addition to the influence of text variation on the frequency of idioms overall, it is likely that there is as much of an influence, if not more, on the frequency of individual idiomatic expressions. However, given the scarcity of individual idioms, the amount of data needed to quantify such assumptions is often prohibitive.

On this aspect, Moon (1998) remarks that some expressions occur in OHPC, a corpus containing 'mannered, literary journalism', with surprisingly high frequencies. For example, she finds that *a leopard does not change its spots* and *the die is cast* are much more frequent there (0.55 instances per million tokens), than in the Bank of English, a corpus with a broader scope. There, they have clearly lower frequencies of 0.19 and 0.28 per million, and if they occur, they still tend to occur in written British journalism.

Moreover, Moon (1998) characterises some idioms, like *beg the question* to be more frequent in 'serious' journalism than fiction and non-fiction. She also suggests that horoscopes are highly frequent sources of idioms. Finally, she adds a counterpoint to the expectation that idioms are particularly common in spoken data. Rather, data shows that idioms are frequent in scripted 'spoken data', such as dialogue in fiction, film and television, and that this skews researchers' perceptions of idioms in spoken data overall.

Finally, McCarthy (1998) does not comment on text types or genres directly, but rather considers idiom usage from a discourse perspective. He states that "*Idioms are never just neutral alternatives to literal, transparent semantically equivalent expressions.*" and "*Idioms always comment on the world in some way, rather than simply describe it.*". This implies that idioms would be more likely to be found in texts which are non-neutral, 'commentary-like', such as editorials in newspapers (cf. Minugh (2008)'s observation), columns, or political language.

2.3.2 Distribution of Literal Usages of Idiom

Literal equivalents of idiomatic expressions, like *come out of the closet* being used in a situation where someone steps out of a wardrobe, pose an additional challenge for both corpus linguistic investigations of idiom and the automatic understanding of idioms alike. For the first, when investigating an idiomatic expression in a corpus by automatically searching for all occurrences, one has to manually filter out literal equivalents of the same phrase, which is time-consuming. For the latter, when, for example, automatically translating a sentence, it is crucial to know whether a seemingly idiomatic phrase is actually used idiomatically or literally in order to produce the correct translation.

However, the frequency of literal equivalents differs drastically between expressions; *cut off one's nose to spite one's face* is unlikely to ever be used literally, whereas the problem is much more significant for an expression like *see stars*. As such, in an attempt to quantify this phenomenon, we look at evidence from corpora annotated with both idioms and their literal equivalents.

There are four main corpora which can provide us with some insight regarding these distributions: by Cook et al. (2008), Sporleder and Li (2009), Sporleder et al. (2010), and Korkontzelos et al. (2013). Cook et al. (2008) present a dataset containing 53 different idiomatic expressions, of which they extract up to 100 instances from the BNC. They annotate these as either *idiomatic*, *literal* or *unclear*. It should be noted, however, that the authors explicitly selected for idioms which they expected to find a balanced sense distribution, which is also true for the other three corpora.

Sporleder and Li (2009) present a corpus of 17 idiom types, for which they extracted all instances from Gigaword. They annotated these potential idioms as either *literal* or *figurative*, excluding ambiguous instances. Sporleder et al. (2010) builds on this, by annotating a larger set of 52 idiom types, and extracting all occurrences from the BNC. They also use a larger tagset, distinguishing *literal*, *non-literal*, *both*, *meta-linguistic*, and *undecided* usages. Finally, Korkontzelos et al. (2013) created a dataset for the SemEval-2013 Shared Task on detecting semantic compositionality in context. They extract instances of 65 idiom types from ukWaC, and label them as *literal*, *idiomatic*, or *both*. For more detail on these corpora, see Section 3.3.

Across these datasets, the overall proportion of idiomatic expressions to literal equivalents varies significantly. The Cook et al. (2008) corpus has 78.54% idiomatic labels, the Sporleder and Li (2009) corpus has 78.25%, the Sporleder et al. (2010) corpus has 44.55%, and the Korkontzelos et al. (2013) corpus has 54.66%. The explanation for these differences is twofold. For one, the selection of idiom types to include has a strong influence, given that the label distributions of individual idiom types varies greatly. In the Cook et al. (2008) data for example, there are expressions used in a literal sense in over 90% of the cases, like *blow smoke*. Vice versa, there are expressions which are used (almost) exclusively in their idiomatic sense, like *keep tabs (on something)*, which is 98% idiomatic. Moreover, the manner of extraction has an influence. For example, when the extraction method allows for more morphosyntactic variation, it is likely to gather more literal equivalents. As such, we cannot draw conclusions about the category of idiomatic expressions as a whole based on these corpora. To get a clearer look at the true distribution of senses among potentially idiomatic expressions, a much larger, unbiased set of expressions would be required.

2.4 Form Variation of Idioms

Although fixedness is a part of what makes an idiom an idiom, this does not mean that all idioms only ever occur in the same form. This is true for some set phrases, like *by and large*, but most idioms allow some extent of morphological, syntactical, and lexical variation. On the other end of the spectrum is an expression like *don't give up the ship*, which allows for all kinds of variation: (examples from Glucksberg (2001))

Tense He will give up the ship; He gave up the ship.

Passivization The ship was given up by the city council.

Number Cowardly? You won't believe it: They gave up all the ships!

Adverbial modification He reluctantly gave up the ship.

Adverbial and adjectival modification After holding out as long as possible, he finally gave up the last ship.

Word substitution Give up the ship? Hell, he gave up the whole fleet!

This form variation of idioms is relevant for both corpus building and corpus analysis approaches, where morphosyntactic variation determines the difficulty of finding instances of an expression in a text corpus. Moreover, it affects the (automatic) semantic interpretation of idioms, since variation, and insertion and modification in particular, are frequently used to modify the meaning of idioms. For example, *the ice was well and truly broken* is a variant of *break the ice*, indicating a stronger version of its meaning of 'to initiate social conversation'. Here, we consider how much idioms can vary overall, how this differs between expressions, and which characteristics determine an idiom's variation potential.

Quantifying variation poses a challenge, due to the unclear nature of what constitutes variation, and because of the manual effort involved in categorising idiom instances in a corpus. However, research by Minugh (2007, 2008) provide a useful starting point. Minugh (2008) focuses exclusively on lexical variation, e.g. *collect dust* as a variant of *gather dust*. In a set of 4,951 idiom instances, he finds 250 of such lexical variants, approximately 5%. These lexical variants can be further classified into categories, including simple substitution, meaning reversal, idiom blending, punning, role reversal, and plain errors.

Minugh (2007) covers a different type of variation: anchoring. This is a type of insertion which connects the idiom to its context, as in 'These dangers are being swept *under the risk-factor rug*'. He finds that many idiom types, perhaps more than expected, allow for this kind of variation, but paradoxically, the number of instances actually containing anchorings is low, at 2.7%.

Another data point is provided by Riehemann (2001). She manually investigates the variation potential of a set of decomposable and non-decomposable, in addition to a set of non-idiomatic collocations, in a 350M word corpus of American English. Riehemann classifies every instance of an idiom occurring in its canonical form or canonical form with an inflectional variation of its head word as a non-variant. Based on this definition, she finds that non-decomposable idioms show variation 10% of the time, decomposable ones 25%, and collocations 84% of the time. In addition, out of a sample of V-NP idioms, 73% are decomposable, indicating that a small, but significant number of idiom instances show some kind of variation.

This is indicative of a more general observation, namely that almost all idioms allow for some kind of variation, but that, at the same time, the overwhelming majority of instances occur in a canonical form. However, rather than quantifying the frequency of non-standard form idioms, most research has focused on categorising the different kinds of variation and discern which idiom characteristics enable these variations.

Glucksberg (2001) points out the relation between different variation types and idiom characteristics as follows: "*[..] idioms are not simply long words. They consist of phrases and, more important, behave as do phrases, [..] If the idiom were simply a long word whose constituents had no meanings of their own, then the idiom should not be syntactically flexible, and one should not be able to replace one of its constituents with a pronoun.*". This gets at a central aspect of idiom, namely whether its component words are denoting or non-denoting (Villada Moirón, 2005). That is, whether a component word of the idiom can be interpreted to refer to some part of the idiomatic meaning, e.g. 'cat' in *let the **cat** out of the bag* clearly refers to the 'secret' part of its idiomatic meaning: 'to disclose a **secret**'. This is also often referred to as decomposability, i.e. whether the meaning of an idiom can be decomposed into component parts.

Usually, a high degree of decomposability is related to a high degree of variability, especially when it comes to allowing for anaphoric reference, syntactic modification, and internal modification. Grégoire (2009) examines the variation potential of 25 multiword expressions in a large corpus of Dutch. She finds that the picture from the data aligns with the hypothesis that decomposable idioms are more likely to show variation and allow for more types of variation. An exception to this is passivisation, which she finds to be unrelated to decomposability, but rather governed by other linguistic factors. It should be noted however, that this is not a 1-to-1 relation, as Glucksberg (2001) asserts that even completely non-decomposable idioms allow for some semantic variation, and even highly decomposable ones do not allow all kinds of variation.

Gustawsson (2006) looks at other factors of variability than decomposability. For one, she finds that having an easily recognised string in the idiom, ideally including rare words, makes it more clearly an idiom. This makes it more susceptible to variation, since even in variant form, the idiomatic meaning still comes through clearly. In addition, she finds that most variation occurs with reasonably semantically transparent idioms. Semantic transparency is a similar, but not identical concept to decomposability, suggesting that both characteristics taken together provide a better indication of variability. Nunberg et al. (1994) makes a similar point, when he identifies 'semantic analyzability' as a major factor in an idiom's variation potential.

CHAPTER 3

Computational Approaches to Idiom

Abstract|Idioms pose an interesting challenge to computational approaches to language, even if those approaches work well for non-idiomatic language. In this chapter, we attempt to provide an overview of what has been done so far, and where gaps in existing research remain. We clarify existing terminology and define a new term: Potentially Idiomatic Expression (PIE), a useful concept encompassing both literal and idiomatic usages of idiomatic expressions. Different datasets containing multiword expressions (MWEs) and PIEs are discussed, and we conclude that each has clear drawbacks, especially regarding size, which could be solved by the construction of a larger, broad-scope corpus (Chapter 5). We also review existing work for three different idiom-related tasks: idiom discovery, idiom extraction, and idiom disambiguation.

3.1 Idioms: A Pain in the Neck for NLP?

In principle, the goal of natural language processing (NLP) is to process all forms of natural language well, for whatever intended purpose this processing has. Not all forms of language have been treated equal in this respect, however. Most work originally focused on canonical, professionally written and edited text in a major language, such as English newspaper text, the most famous example of which is the Wall Street Journal corpus (Paul and Baker, 1992). The reason for this is obvious: natural language processing is very difficult, and English newswire is simply the easiest and most available thing to start with. More recently, great strides have been made within NLP, especially due to the 'deep learning tsunami' dramatically increasing performance and practical applicability (Manning, 2015). For example, NLP is of such reliably high quality that it can be used in both voice- and text-based interaction with virtual assistants. As such, this is the right moment to tackle more difficult types of language. This includes languages very different than English (e.g. morphologically rich languages), non-canonical text (e.g. transcriptions and noisy social media text), and more challenging language phenomena (e.g. metaphor, sarcasm, multimodality).

Idiomatic expressions are one of these phenomena, and in this chapter we provide an overview of (recent) research on idiomatic expressions within NLP. Idioms are challenging for multiple reasons: individual idiomatic expressions are rare, but idioms as a group are surprisingly frequent, they consist of multiple words and can take many different forms, and most crucially, their meaning is unpredictable from their form, i.e. it is non-compositional. In this chapter, we will attempt to define what the task of *idiom processing* consists of and how solving this task can best be approached (Section 3.2), which idiom-related datasets exist (Section 3.3), and which approaches to different idiom processing subtasks have been investigated (Section 3.4).

3.2 Definitions & Terminology

Within NLP, the main goal of idiom processing is to be able to interpret idiomatic expressions in text correctly. Given the sentence in Example 5 and the idiom *buy the farm*, this consists of three parts. One part is knowing that *buy the farm* is an idiom in the first place. This can be done by utilising a lexical resource, like an idiom dictionary, or automatically from text, e.g. using fixedness and collocation-based measures. In addition, one should be able to detect that the snippet 'bought the farm' in the sentence is a form of *buy the farm*. Finally, to interpret this snippet (and the sentence) correctly, one needs to decide whether 'bought the farm' in the sentence refers to a literal buying of a literal farm, or whether it is used idiomatically, in which case it means 'to die in a plane crash'. If all three subtasks have been completed, one can conclude that the original sentence contains 'bought the farm', which is a variant of the possibly idiomatic *buy the farm*, which is indeed used idiomatically in this case. As such, the sentence can be paraphrased as Example 6.

(5) If the engine quits, or even misses a couple of beats, they have **bought the farm**.

(6) If the engine quits, or even misses a couple of beats, they will die in a plane crash.

Although these three steps are necessary for idiom processing, it does not mean that they have to be tackled one by one, or conversely, all at once. For example, the first two steps could be done jointly, discovering instances in text based on fixedness and collocation-based measures rather than unifying or normalising them to types. Similarly, the last two steps can be done jointly, if one extracts only idiomatic usages of known idiomatic expressions from text.

This allows for flexibility in approaches towards the problem of idiom processing, but unfortunately it also causes terminological confusion. In existing idiom research, the task of discovering new idiomatic expressions is called *type-based idiom detection* and the task of figuring out the meaning of a potential idiom within context is called *token-based idiom detection* (cf. Sporleder et al., 2010; Gharbieh et al., 2016, for example), although this usage is not always consistent in the literature. Because these terms are very similar, they are potentially confusing. Other terminology comes from literature on multiword expressions, a broader category of expressions including collocations, particle verbs, and other types of set phrases. Here, the task of finding new MWE types is called *MWE discovery* and finding instances of known MWE types is called *MWE identification* (Constant et al., 2017). Note, however, that *MWE identification* generally consists of finding only the idiomatic usages of these types (e.g. Ramisch et al., 2018). This means that *MWE identification* consists of both the extraction and disambiguation tasks, performed jointly.

In order to clear up the terminology, we propose a new[1] term: *potentially idiomatic expressions*, or *PIEs* for short. The term *potentially idiomatic expression* refers to those expressions which can have an idiomatic meaning, regardless of whether they actually have that meaning in a given context.[2] We introduce this term because the ambiguity of phrases like *wake up and smell the coffee* poses a terminological problem. Usually, these phrases are called *idiomatic expressions*, which is suitable when they are used in an idiomatic sense, but not so much when they are used in a literal sense. So, *see the light* is a PIE in both Example 7 and 8,

[1] Note that Cook et al. (2008) came up with a similar term, *potentially-idiomatic combinations*.

[2] Ambiguity is not equally distributed across phrases. As with words, there are single sense phrases, with only an idiomatic sense, such as *piping hot*, which can only get the figurative interpretation 'very hot'. More commonly, there are phrases with exactly two senses, a literal and an idiomatic sense, such as *wake up and smell the coffee*, which can take the literal meaning of 'waking up and smelling coffee', and the idiomatic meaning of 'facing reality and stop deluding oneself'. Sometimes, phrases can have more than two senses, e.g. one literal sense and multiple idiomatic ones, as in *fall by the wayside*, which can take the literal meaning 'fall down by the side of the road', the idiomatic meaning 'fail to persist in an endeavour', and the alternative idiomatic meaning 'be left without help'.

while it is an idiomatic expression in the first context, and a literal phrase in the latter context.

(7) After another explanation, I finally **saw the light**.

(8) I **saw the light** of the sun through the trees.

Given the term PIE, the three subtasks of idiom processing can be easily distinguished and named, doing away with confusion. Here, we propose calling the discovery of (new) PIE types simply *PIE discovery*, analogous to *MWE discovery*, the extraction of instances of known PIE types in text *PIE extraction*, and the disambiguation of PIE instances in context *PIE disambiguation*. These terms can be easily extended from just PIEs to MWE in general as well, creating three tasks: *MWE discovery*, *MWE extraction*, and *MWE disambiguation*. However, since the existence of two distinct senses is less clear for all MWEs than it is for PIEs, it makes sense to join MWE extraction and disambiguation into *MWE identification*.

3.3 Idiom Datasets

There are many datasets containing idiom-related annotations, and they are discussed in this section. These are corpora containing both literal and idiomatic occurrences of idiomatic expressions, and they are labelled by their meaning. As such, we would call them sense-annotated PIE corpora; corpora containing potentially idiomatic expressions with labels indicating their meaning. The four biggest of these are discussed here in detail, while an overview of other, smaller datasets is provided in Section 3.3.5.

There are four sizeable corpora of idiom annotations for English: the Gigaword dataset (Sporleder and Li, 2009), the VNC-Tokens Dataset (Cook et al., 2008), the IDIX Corpus (Sporleder et al., 2010), and the SemEval-2013 Task 5 dataset (Korkontzelos et al., 2013). An overview of these corpora is presented in Table 3.1. The table includes the number

Name	Types	Instances	Senses	Base Corpus	Syntax Types
VNC-Tokens	53	2,984	3	BNC	V+NP
Gigaword	17	3,964	2	Gigaword	V+NP/PP
IDIX	52	4,022	6	BNC	V+NP/PP
SemEval-2013	65	4,350	4	ukWaC	unrestricted

Table 3.1: Overview of existing corpora of potentially idiomatic expressions and sense annotations for English. The syntax types column indicates the syntactic patterns of the idiom types included in the dataset. The base corpora are the British National Corpus (BNC, Burnard, 2007), ukWaC (Ferraresi et al., 2008), and Gigaword (Graff and Cieri, 2003).

of different idiom types in the corpora (i.e. different expressions, such as *sour grapes* and *speak of the devil*), the number of PIE instances, the number of different senses annotated (e.g. *idiomatic*, *literal*, and *unclear*), the corpus the data was extracted from, and the 'syntactic type' of the expressions covered. Syntactic type means that, in some cases, only idiom types following a certain syntactic pattern were included, e.g. only verb-(determiner)-noun combinations such as *hold your fire* and *see stars*.

3.3.1 VNC-Tokens

The VNC-Tokens dataset contains 53 different PIE types. Cook et al. (2008) extracted up to 100 instances from the British National Corpus for each type, for a total of 2,984 instances. These types are based on a pre-existing list of verb-noun combinations and were filtered for frequency and whether two idiom dictionaries both listed them. Instances were extracted automatically, by parsing the corpus and selecting all sentences with the right verb and noun in a direct-object relation. It is unclear whether the extracted sentences were manually checked, but no false extractions are mentioned in the paper or present in the dataset.

All extracted PIE instances were annotated for sense as either *idiomatic*, *literal* or *unclear*. This is a self-explanatory annotation scheme, but Cook et al. note that senses are not binary, but can form a continuum. For example, the idiomaticity of *have a word* in 'You have my word' is different from both the literal sense in Example 9 and the figurative sense in Example 10. They instructed annotators to choose *idiomatic* or *literal* even in ambiguous middle-of-the-continuum cases, and restrict the *unclear* label only to cases where there is not enough context to disambiguate the meaning of the PIE.

(9) The French **have a word** for this concept.

(10) My manager asked to **have a word** with me.

3.3.2 Gigaword

Sporleder and Li (2009) present a corpus of 17 PIE types, for which they extracted all instances from the Gigaword corpus (Graff and Cieri, 2003), yielding a total of 3,964 instances. Sporleder and Li extracted these instances semi-automatically by manually defining all inflectional variants of the verb in the PIE and matching these in the corpus. They did not allow for inflectional variations in non-verb words, nor did they allow intervening words. They annotated these potential idioms as either *literal* or *figurative*, excluding ambiguous and unclear instances from the dataset.

3.3.3 IDIX

Sporleder et al. (2010) build on the methodology of Sporleder and Li (2009), but annotate a larger set of idioms (52 types) and extract all occurrences from the BNC rather than the Gigaword corpus, for a total of 4,022 instances including false extractions.[3] Sporleder et al. use a more complex semi-automatic extraction method, which involves parsing the corpus, manually defining the dependency patterns that match the PIE,

[3] The corpus contains 52 types, rather than the 78/100 types mentioned in the paper, similarly, the actual number of instances in the corpus differs from that reported in the paper. (Caroline Sporleder, personal communication, October 9, 2016)

and extracting all sentences containing those patterns from the corpus. This allows for larger form variations, including intervening words and inflectional variation of all words. In some cases, this yields many non-PIE extractions, as for *recharge one's batteries* in Example 11. These were not filtered out before annotation, but rather filtered out as part of the annotation process, by having *false extraction* as an additional annotation label.

For sense annotation, they use the most extensive tagset of all existing corpora, distinguishing *literal*, *non-literal*, *both*, *meta-linguistic*, *embedded*, and *undecided* labels. Here, the *both* label (Example 12) is used for cases where both senses are present, often as a form of deliberate word play. The *meta-linguistic* label (Example 13) applies to cases where the PIE instance is used as a linguistic item to discuss, not as part of a sentence. The *embedded* label (Example 14) applies to cases where the PIE is embedded in a larger figurative context, which makes it impossible to say whether a literal or figurative sense is more applicable. The *undecided* label is used for unclear and undecidable cases. They take into account the fact that a PIE can have multiple figurative senses, and enumerate these separately as part of the annotation.

(11) These high-performance, rugged tools are claimed to offer the best value for money on the market for the enthusiastic d-i-yer and tradesman, and for the first time offer the possibility of a **battery recharging** time of just a quarter of an hour. (from IDIX corpus, ID #314)

(12) Left **holding the baby**, single mothers find it hard to fend for themselves. (from Sporleder et al., 2010, p.642)

(13) It has long been recognised that expressions such as to pull someone's leg, to have a bee in one's bonnet, to **kick the bucket**, to cook someone's goose, to be off one's rocker, round the bend, up the creek, etc. are semantically peculiar. (from Sporleder et al.,

2010, p.642)

(14) You're like a restless bird in a cage. When you get out of the cage, you'll **fly** very **high**. (from Sporleder et al., 2010, p.642)

The *both*, *meta-linguistic*, and *embedded* labels are useful and linguistically interesting distinctions, although they occur very rarely (0.69%, 0.15%, and an unknown percentage, respectively).

3.3.4 SemEval-2013 Task 5b

Korkontzelos et al. (2013) created a dataset for SemEval-2013 Task 5b, a task on detecting semantic compositionality in context. They selected 65 PIE types from Wiktionary, and extracted instances from the ukWaC corpus (Ferraresi et al., 2008), for a total of 4,350 instances. It is unclear how they extracted the instances, and how much variation was allowed for, although there is some inflectional variation in the dataset. An unspecified amount of manual filtering was done on the extracted instances.

The extracted PIE instances were labelled as *literal*, *idiomatic*, *both*, or *undecidable*. Interestingly, they crowdsourced the sense annotations using CrowdFlower, with high agreement (90%–94% pairwise). Undecidable cases and instances on which annotators disagreed were removed from the dataset.

3.3.5 Other Idiom-Related Datasets

In addition to the four datasets discussed so far, there is an array of other datasets containing English idioms. These datasets can be either smaller in scale, part of a wider category of expressions, or they can annotate idioms for a different purpose than disambiguation.

Street et al. (2010) present a pilot study in which they annotate 4,500 sentences, containing 69,000 tokens from the American National Corpus for idiomatic expressions, using multiple annotators. They annotate idiom spans, and type of idiom. These types are based on syntactic form, and they identify 3 types: prepositional phrase, verb-noun construction, and subordinate clause. Later, they suggest also adding verb-preposition construction. In this corpus, they find only 154 idiom tokens.

Another small-scale dataset is constructed by Gong et al. (2016), who extract instances for 104 English idioms and 64 Chinese ones. From Google Books, they extract and annotate 1 idiomatic and 1 literal example for each type, yielding a total of 336 instances.

Instead of creating their own dataset, Peng et al. (2015) extend the VNC dataset. They select 12 idiom types from the VNC and extract additional instances for those types from other corpora, in addition to the examples from the BNC already included in the VNC. They annotate these with binary sense labels, i.e. either literal or idiomatic. The final dataset consists of 2,072 instances, compared to the original number of 541 instances for the 12 selected types.

Other work focuses not just on idiomatic expressions, but on multiword expressions (MWEs) as a whole. As idioms are a subcategory of MWEs, these corpora also include some idioms. The most important of these are the PARSEME corpus (Savary et al., 2018) and the DiMSUM corpus (Schneider et al., 2016).

DiMSUM provides annotations of over 5,000 MWEs in approximately 90K tokens of English text, consisting of reviews, tweets and TED talks. However, they do not categorise the MWEs into subtypes, meaning we cannot easily quantify the number of idioms in the corpus. In contrast to the corpus-specific sense labels seen in other corpora, DiMSUM annotates MWEs with WordNet supersenses, which provide a broad category of meaning for each MWE.

Similarly, the PARSEME corpus consists of over 62K MWEs in almost 275K tokens of text across 18 different languages (with the notable exception of English). The main differences with DiMSUM, except for scale and multilingualism, are that it only includes verbal MWEs, and that subcategorisation is performed, including a specific category for idioms. Idioms make up almost a quarter of all verbal MWEs in the corpus, although the proportion varies wildly between languages. In both corpora, MWE annotation was done in an unrestricted manner, i.e. there was not a predefined set of expressions to which annotation was restricted.

Kato et al. (2018) also create a corpus of MWEs, and in their case only verbal MWEs in English. They extract all instances of a set of MWE types taken from Wiktionary from part of the OntoNotes corpus (Hovy et al., 2006). Since simple extraction based on words can yield a lot of noise, i.e. non-instances, they refine those extractions based on the gold-standard part-of-speech tags and parse trees that are present in the OntoNotes corpus. Most interesting, however, is their use of crowdsourcing for distinguishing between literal equivalents of MWE phrases like *get up* in 'He gets up early' and actual MWE instances like in 'He gets up a hill'. They frame the task as a sense annotation task, asking crowdworkers to label instances as either literal, non-literal, unclear, or 'none of the above'. Using this procedure, they create a corpus of 7,833 verbal MWE instances, of 1,608 different types.

Not all corpora just contain instances of idiom or MWE extracted from text, annotated with meaning. There exist corpora of other aspects of idiom, such as paraphrases and definitions. For example, Pershina et al. (2015) present a study on paraphrase detection for idiom definitions, for which they annotate a corpus of 1,400 idioms for paraphrases. That is, they intend to find idioms which have the same meaning, e.g. *seventh heaven* and *cloud nine*. They report that 460 out of 1,400 idioms can be considered as paraphrases of other idioms in the dataset. They used 3 annotators, and only kept idioms with (near-)unanimous agreement.

A different type of paraphrases are documented by Liu and Hwa (2016), who aim to replace idioms in context by non-idiomatic paraphrases of their idiomatic meaning, e.g. rephrase *work in harness* to *work together*. They present a corpus of tweets containing an idiom, a definition of that idiom, and 2 human-generated shortenings of that idiom definition, containing 172 samples in total.

Muzny and Zettlemoyer (2013) annotate idioms on the type-level, i.e. they annotate dictionary entries on whether they are actually idioms or just compositional expressions. They gather data from Wiktionary and annotate over 9,500 multi-word entries marked as *idiomatic* for whether they are actually idiomatic or literal. All entries were annotated by two annotators, with high agreement: 82% Kappa.

3.3.6 Overview

In sum, there is large variation in corpus creation methods, regarding PIE definition, extraction method, annotation schemes, base corpus, and PIE type inventory. Depending on the goal of the corpus, the amount of deviation that is allowed from the PIE's dictionary form to the instances can be very little (Sporleder and Li, 2009), to quite a lot (Sporleder et al., 2010). The number of PIE types covered by each corpus is limited, ranging from 17 to 65 types, often limited to one or more syntactic patterns. The extraction of PIE instances is usually done in a semi-automatic manner, by manually defining patterns in a text or parse tree, and doing some manual filtering afterwards. This works well, but an extension to a large number of PIE types (e.g. several hundreds) would also require a large increase in the amount of manual effort involved. Considering the sense annotations done on the PIE corpora, there is significant variation, with Cook et al. (2008) using only three tags, whereas Sporleder et al. (2010) use six. Outside of PIE-specific corpora there are MWE corpora, which provide a different perspective. A major difference there is that annotation is not restricted to a pre-specified set of expressions, which has not been done for PIEs specifically.

3.4 Approaches to Idiom Processing

As discussed in Section 3.2, idiom processing consists of three parts: PIE discovery, PIE extraction, and PIE disambiguation. In this section, different approaches to these tasks will be discussed, focusing on the differences between supervised and unsupervised methods, various ways of word and sentence representation, and the difficulty of consistent evaluation and comparison of different approaches.

3.4.1 PIE Discovery

PIE discovery is the task of distinguishing potentially idiomatic expressions from other multiword phrases, where the main purpose is to expand idiom inventories with rare or novel expressions (Fazly et al., 2009; Muzny and Zettlemoyer, 2013; Gong et al., 2016; Senaldi et al., 2016, among others). For example, its goal is to determine that of the two frequent verb-noun pairs *lose face* and *keep fish*, the first has an associated idiomatic expression, whereas the second does not.

One of the main lines of work towards solving this task is based on the notion that idiomatic expressions, like other multiword expressions, are less flexible syntactically and lexically than non-idiomatic, compositional expressions. For example, *lose face* will almost always be used without a determiner (? 'lose the face'[4]), with the noun in singular (? 'lose faces'), and without any internal modification (? 'lose a lot of face'). This category-specific property makes a good starting point for making an automatic distinction between phrases with and without an associated idiomatic meaning. This approach was popularised by Fazly and Stevenson (2006); Fazly et al. (2009), who relied on PMI for measuring lexical fixedness and the distribution over a set of syntactic patterns for syntactic fixedness. The lexical fixedness metric was further explored by Salton et al. (2017), significantly the method's performance. An alternative avenue is pursued by Williams (2017), who relies on a text partitioning algorithm to discover MWE types (as opposed to idioms specifically). Liebeskind and HaCohen-Kerner (2016) also work on MWE discovery, and combine fixedness and semantic features in a machine learning

[4] The question mark is used to indicate cases whose idiomaticity is unsure.

setup, where they find that the fixedness-related features do not provide much benefit in addition to features targeting semantic idiosyncrasy.

Semantic idiosyncrasy is the other defining feature of idiomatic expressions, namely the fact that their meaning is not directly derivable from the meaning of their component words. For example, 'face' in *lose face* refers to 'respect, dignity', rather than any actual face. Similarly, this can make idiomatic expressions stand out within the context they are used in. For example, the word 'pregnant' tends to occur in contexts related to pregnancy, but when used as part of the idiom *pregnant silence*, it can occur in all kinds of contexts one would not normally expect it to occur in. These two discrepancies, between idiom and component words and between context and component words, make for useful features to exploit in a computational approach to PIE discovery.

Most recent approaches to this task focus on this aspect, such as the work by Muzny and Zettlemoyer (2013). They aim to distinguish idiomatic and non-idiomatic multiword phrases in the online dictionary Wiktionary by exploiting the words in the idiom and its accompanying definition. Exploiting relations from the lexical database WordNet, they classify phrases as idiomatic if the discrepancy between the 'regular' meaning of its component words and its definition is large and vice versa. A similar approach is taken by Verma and Vuppuluri (2015), but they rely on the dictionary definitions of the idiom's component words rather than the words themselves. A third source of information is added by Salehi et al. (2014a,b), who make use of translations of MWEs in Wiktionary, on the assumption that a translation of an idiomatic phrase (e.g. *kick the bucket*-*sterven* (Dutch)) will have little string overlap with the translations of its component words (e.g. *kick*-*schoppen*, *bucket*-*emmer*).

In more recent years, these lexical and string-based approaches have been superseded by methods making use of distributional similarity, usually in the form of word embeddings. The underlying assumption is the same: that the meaning of the component words (represented as a vector) is different (distant) from the meaning of the context or the expression as a whole (another vector). Within this line of work, the main focus is on finding the optimal representations, be it count-based word vectors, neural word embeddings, or sense embeddings, tuning parameters, and examining methods of composing meaning representations for multiword phrases and contexts. Based on work on different languages, different datasets and both idioms in particular and MWEs in general, it seems that neural word embeddings are better than count-based vectors (Salehi et al., 2015; Weeds et al., 2017) and sense embeddings outperform sense-agnostic word embeddings (Köper and Schulte im Walde, 2017; Bott and Schulte im Walde, 2017).

A final way of exploiting the semantic non-compositionality is introduced by Villada Moirón and Tiedemann (2006), based on the assumption that most idiomatic phrases will not have direct (word-for-word) counterparts in other languages. As such, they are usually paraphrased in translation, in many different ways. Villada Moirón and Tiedemann exploit this by calculating the number of different translations of a phrases based on alignments, and classifying phrases with a large variance in translations as possibly idiomatic. Similar approaches have been taken by Cap (2017), who applies this method to noun-noun compounds in German and English, and Tsvetkov and Wintner (2010), who do this for Hebrew, but mostly rely on incorrect and complex (not 1:1) alignments to detect non-compositionality.

3.4.2 PIE Extraction

PIE extraction is the task of extracting all occurrences of a known idiomatic expression from a text, regardless of whether they are used literally, idiomatically, or otherwise. That is, given a known idiomatic expression like *lose face*, all instances of that phrase should be extracted, e.g. ‘face was lost’, ‘lose two faces’, ‘losing face’.

There is not much existing work that focuses on PIE extraction, with the main body of relevant work consisting of just two papers: by Flor and

Beigman Klebanov (2018) and by Iñurrieta et al. (2016). Flor and Beigman Klebanov (2018) is the most directly applicable of the two in that they focus on PIE extraction specifically. They set out to find and study idioms used in EFL essays, and aim to automate the identification of 'candidate idioms', which corresponds directly to what we term PIE. Their approach is dictionary-base, meaning they rely on a dictionary to come up with their inventory of idiom types. Flor and Beigman Klebanov extract all idioms from Wiktionary and filter out single words, verb-particle constructions, prepositional verbs, and dialogue expressions. These idioms are then inflected in all possible ways, and determiners, prepositions, and function words are marked as optional parts of the idiom. Their algorithm then extracts idioms by identifying any sentence which contains the mandatory parts of the idiom in order, with a pre-specified maximum number of intervening words.

Applying this method to a 1.1M word corpus of EFL essays yields 5,704 instances, of which 3,709 are non-PIE, 1,302 are idiomatic, and 693 are literal. The high proportion of non-PIEs in the output (65%) prompts additional experiments, show that a lower number of intervening words reduces false extractions dramatically, but also reduces the amount of true extractions not insignificantly. In addition, Flor and Beigman Klebanov find that making optional elements mandatory helps to reduce false extractions drastically at an acceptable loss in recall.

In addition, there is closely-related work by Iñurrieta et al. (2016), who present a system for the dictionary-based extraction of verb-noun combinations (VNCs) in English and Spanish. In their case, the VNCs can be any kind of multiword expression, which they subdivide into literal expressions, collocations, light verb constructions, metaphoric expressions, and idioms. They extract 173 English VNCs and 150 Spanish VNCs and annotate these with both their lexico-semantic MWE type and the amount of morphosyntactic variation they exhibit. Iñurrieta et al. then compare a word sequence-based method, a chunking-based method, and a parse-based method for VNC extraction. Each method relies on the morphosyntactic information in order to limit false extractions. Precision is evaluated manually on a sample of the extracted VNCs, and recall is estimated by calculating the overlap between the output of the three methods. Evaluation shows that the methods are highly complementary both in recall, since they extract different VNCs, and in precision, since combining the extractors yields fewer false extractions. Whereas Iñurrieta et al. (2016) focus on both idiomatic and literal uses of the set of expressions, Savary and Cordeiro (2017) tackle only half of that task, namely extracting only literal uses of a given set of verbal MWEs in Polish. This complicates the task, since it combines extracting all occurrences of the verbal MWEs and then distinguishing literal from idiomatic uses. Interestingly, they also experiment with models of varying complexity, i.e. just words, part-of-speech tags, and syntactic structures. Their results are hard to put into perspective however, since the frequency of literal verbal MWEs in their corpus is very rare, whereas corpora containing PIEs tend to show a more balanced distribution.

Other similar work also focuses on MWEs more generally, or on different subtypes of MWEs. In addition, these tend to combine both extraction and disambiguation in that they aim to extract only idiomatically used instances of the MWE, without extracting literally used instances or non-instances. Within this line of work, Baldwin (2005) focuses on verb-particle constructions, Boukobza and Rappoport (2009) on verbal MWEs (including idioms), and Pasquer et al. (2018) on verbal MWEs (especially non-canonical variants).

Both Boukobza and Rappoport (2009) and Pasquer et al. (2018) rely on a pre-defined set of expressions, whereas Baldwin (2005) also extracts unseen expressions, although based on a pre-defined set of particles and within the vary narrow syntactic frame of verb-particle constructions. The work of Baldwin is distinctive in that it builds an unsupervised system using existing NLP tools (PoS taggers, chunkers, parsers) and finds that a combination of systems using those tools performs best. Pasquer et al. and Boukobza and Rappoport, by contrast, use supervised classifiers which require training data, not just for the task in general, but specific to the set of expressions used in the task.

3.4.3 PIE Disambiguation

The third and final part of idiom processing is PIE disambiguation, which involves distinguishing between idiomatic, literal, and other uses of potentially idiomatic expressions. For example, given that we know 'buy the farm' in Example 15 is a form of the idiom *buy the farm*, the challenge is to determine whether it is used literally, idiomatically (as it is there), or in some other way that does not fit those two classes. Much previous work has tackled this task, and we discuss this along two major lines of research: unsupervised and supervised methods.

(15) They gambled with as much reckless abandon as they flew their airplanes. They knew they might **buy the farm** tomorrow.

Supervised methods for PIE disambiguation use training data to learn context-sense relations for either one PIE at a time or all at once. Unsupervised methods strive to exploit inherent properties of PIEs and their context to determine the right sense in-context. In addition, there are some that combine the two, e.g. acquiring training data for a supervised classifier in an unsupervised manner. In general, supervised learning yields better performance, especially when training a separate classifier for each expression, but as they rely on large amounts of training data for each PIE type, they cannot be easily extended to deal with new, unseen expressions. Unsupervised classifiers and supervised classifiers which are not expression-specific do not have this drawback, but their performance tends to be worse. As a result, there are also several approaches combining unsupervised and supervised classification.

Fazly et al. (2009) build an unsupervised system that labels a PIE instance as idiomatic when it occurs in its canonical form (i.e. dictionary form), and literal otherwise. In addition, they use the data labelled by this unsupervised system to train one classifier per expression, using context similarity. They compare this to the same classifier trained using gold labels. With 72.4% macro-accuracy, the canonical form-based method outperforms the distantly supervised method (65.8%), but is clearly outperformed by the supervised classifier trained on gold data (82.7%).

In a similar vein, Li and Sporleder (2009) experiment with combining unsupervised and supervised methods. They use Sporleder and Li (2009)'s lexical cohesion graph-based classifier. This classifier compares lexical cohesion in the sentence containing the PIE with and without the PIE's component words. If cohesion is higher with the PIE's component words, that indicates that the literal meaning of the words fits the context, and thus that the PIE is used in its literal sense. Conversely, if cohesion is lower, that is taken to indicate an idiomatic meaning. Li and Sporleder (2009) combine it with a supervised SVM using salience and word-relatedness features to iteratively generate training data for the SVM. They find that the combined classifier clearly outperforms the unsupervised classifier (87.03% vs. 78.38% micro-accuracy), but falls short of the same SVM trained on gold data (90.34%). In contrast to Fazly et al. (2009), the same classifier is used for all expressions, meaning it has higher generalisation potential.

Gharbieh et al. (2016), on the other hand, find that a single supervised classifier for all PIE types in one dataset performs poorly. They use an SVM comparing word embedding-based representations of the PIE and the context for classification, and find that a one-SVM-per-expression setup does clearly better (86.3%/88.3% macro-accuracy) than a one-SVM-for-all-expressions setup (68.9%/69.4%). In comparison, an unsupervised approach using k-means clustering over representations distantly labelled using a canonical form classifier performs clearly better (69.7%/ 78.1%). Other promising unsupervised approaches are by Gong et al. (2016), who use the distance between embedding-based representations of the PIE component words and the context to disambiguate PIE senses on a small dataset for English and Chinese, and by Ehren (2017) who experiments with variations on Sporleder & Li's lexical cohesion graph approach using different similarity measures, evaluating on a small corpus of German PIEs.

Another well-performing unsupervised methods is presented in the work by Liu and Hwa (2018), who, as in previous work, calculate the similarity between embedding-based representations of the PIE and its context. They improve this approach, however, by scaling the similarity threshold for classifying something as idiomatic or literal so that it maximally separates the two sets of labels. This means it is not completely unsupervised, as it requires multiple samples of that PIE type to find the threshold, but these samples need not be labelled, so they could be gathered automatically from a large corpus. Using the classifications generated by this heuristic in a Gibbs or Naive Bayes classifier (i.e., in a semi-supervised setting), they outperform other unsupervised approaches and come close to scores by supervised approaches. Unfortunately, Liu and Hwa evaluate on a subset of the VNC dataset, which makes their results incomparable to those of Gharbieh et al. (2016).

Although training a separate classifier for each idiomatic expression, requiring significant amounts of training data for each expression, has very little practical use, it is still a popular approach. This is likely because, given the limited scope (in types) of existing datasets, these approaches yield the highest performance.

Additional work in this vein comes from the participants in SemEval 2013 shared task 5b Korkontzelos et al. (2013). Participants used logistic regression with part-of-speech and tf-idf-based features Jimenez et al. (2013), noun-based word overlap Byrne et al. (2013), and similarities based on WordNet and word embeddings using a rule induction classifier Siblini and Kosseim (2013). In all cases, participants either participated only in setting where the test data contained only 'seen' expressions or tested on 'unseen' expressions as well and scored poorly.

More recent work focuses on using word embeddings as representations for both the context of the PIE and its component words and then exploiting the similarity between those representations as a feature. Peng et al. (2015) make use of both projections of PIE representations on to context representations and scatter matrices comparing contexts of literal and idiomatic uses, and report good scores using the Frobenius norm to measure distance between scatter matrices. However, they only evaluate on a small subset of the VNC Tokens dataset, which makes it hard to assess its true quality. Moving beyond single-sense word embeddings, Köper and Schulte im Walde (2017) experiment with different types of multi-sense embeddings for idiomaticity classification, not on English PIEs, but on German particle verbs. They find a clear benefit of multi-sense over single-sense embeddings, but do not find a single method that is clearly better than others. Moreover, the evaluation on German particle verbs makes it hard to value its applicability to English idiomatic expressions. King and Cook (2018) similarly experiment with different types of embedding representations to disambiguate PIEs in the English VNC dataset, and they find that regular word2vec embeddings outperform Siamese CBOW and skip-thought vectors. In combination with the embedding representations, incorporating a canonical form feature, capturing form variation, benefits performance as well.

Alternatively, a supervised classifier can be trained in such a way that it can also deal with unseen expressions. Ideally, this would combine the good performance of a per-expression supervised classifier with the broad applicability of an unsupervised classifier. In practice however, this has not always been the case Gharbieh et al. (2016); Jimenez et al. (2013). Nevertheless, promising all-expression supervised classifiers are being developed, mostly based on deep neural network architectures. For example, Salton et al. (2016) use bi-GRUs to create skip-thought vectors as sentence representations for a PIE and its context. Classifying these using SVMs trained per-expression yields good results, but the performance of a SVM trained on all expressions simultaneously is very close. Generally, deep neural networks have a lot of potential, as has also been shown for the distinct but similar task of verbal MWE identification (Ramisch et al., 2018; Waszczuk et al., 2019).

Additional advances come from combining PIE disambiguation with another task. Liu et al. (2017) integrate an idiomaticity detection component in their tree-LSTM based sentiment analysis system and find clear gains in sentiment analysis performance, although the size of the gain naturally depends on the amount of idioms in the texts to be analysed. Focusing on non-compositionality from a broader perspective, Do Dinh et al. (2018) apply a multitask learning approach to jointly detect metaphors and idioms in different corpora, in both English and German. They find that the tasks are mutually informative, but there are clear caveats as to which datasets are beneficial to each other and in which type of multi-task learning setup.

3.4.4 Overview

Based on the results of different approaches, although not directly comparable, we conclude that supervised approaches generally perform better than unsupervised ones, whereas semi-supervised or distantly supervised approaches are somewhere in between. Secondly, training a separate classifier for each PIE type seems to strongly benefit performance, but has the crucial disadvantage of not being extensible to unseen types for which there is no training data. At the same time, it is more difficult to achieve the same level of performance using a single supervised classifier for different PIE types, which would not have this drawback.

Ideally, we would be able to compare approaches from different papers directly, but this is often challenging. The lack of an established evaluation framework means that reported results for PIE disambiguation are often on different (splits of) datasets, obtained in different ways (cross-validation, leave-one-out) using a range of different metrics (micro- and macro-averaged accuracy and F1-score). For example, Sporleder and Li (2009) report micro-accuracy and micro-F1 scores on the Gigaword corpus, whereas Fazly et al. (2009) report macro-accuracy scores on the VNC-Tokens dataset. Peng et al. (2015) use only a handpicked subset of the VNC, but add additional examples of their own, while Gong et al. (2016) create their own dataset from scratch and do not use any established datasets. A potential solution to this problem was provided by SemEval-2013 Task 5b on PIE disambiguation Korkontzelos et al. (2013), as results from different participants could be directly compared. However, this dataset does not seem to have been used by the community since.

Furthermore, the quality of evaluation is hampered by the size and nature of the existing datasets for PIE disambiguation. Each corpus has somewhat different benefits and downsides: VNC-Tokens only contains verb-noun combinations (e.g. *hit the road*) and contains some types which we would consider light-verb constructions rather than idioms (e.g. *have a future*); the IDIX corpus covers various syntactic types and has a large number of instances per PIE type, but is partly single-annotated; the SemEval dataset is large and varied, but the base corpus, ukWaC, is noisy. An evaluation dataset would ideally measure the practical usability of an idiom disambiguation system, that is, test whether it shows robust disambiguation performance across different idiom types of different syntactic patterns, including many unseen types, and, if necessary, using only little training data per type. On top of this, such an evaluation dataset would have a limited number of instances per type, but a large number of instances in total, so that it allows for training of data-hungry supervised classifiers like deep neural networks.

PART II

Corpus Construction

work like a beaver *informal* work steadily and industriously. (The beaver is referred to here because of the industriousness with which it constructs the dams necessary for its aquatic dwellings.) ● *She has an important deadline coming up, so she's been working like a beaver.* ● *You need a vacation. You work like a beaver in that kitchen.*

CHAPTER 4

[*]Annotation and Extraction of Potentially Idiomatic Expressions

Abstract|In this chapter, we present work on the annotation and extraction of potentially idiomatic expressions (PIEs). Existing corpora of PIEs are small and have limited coverage of different PIE types, which hampers research. To further progress on the extraction and disambiguation of potentially idiomatic expressions, larger corpora of PIEs which enable better generalisation over different types are necessary. In addition, larger corpora are a potential source for valuable linguistic insights into idiomatic expressions and their variability. We propose automatic tools to facilitate the building of larger PIE corpora, by investigating the feasibility of using dictionary-based extraction of PIEs as a pre-extraction tool for English. We do this by assessing the reliability and coverage of idiom dictionaries, the annotation of a PIE corpus, and the automatic extraction of PIEs from a large corpus. Results show that combinations of dictionaries are a reliable source of idiomatic expressions, that PIEs can be annotated with a high reliability (0.74–0.91 Fleiss' Kappa), and that parse-based PIE extraction yields highly accurate performance (85% F1-score). Combining complementary PIE extraction methods increases reliability further, to over 90% F1-score.

[*] Chapter adapted from: Haagsma, H., Nissim, M., and Bos, J. (2019). Casting a Wide Net: Robust Extraction of Potentially Idiomatic Expressions. arXiv:1911.08829.

4.1 Introduction

Idioms show significant syntactic and morphological variability, which makes them hard to detect for machines. Moreover, their non-compositional nature makes idioms really hard to interpret, because their meaning is often very different from the meanings of the words that make them up. Hence, successful systems need not only be able to recognise idiomatic expressions in text or dialogue, but they also need to give a proper interpretation to them.

However, as described in Section 3.3, only small datasets for idiom interpretations are available. This is not surprising, as preparing and compiling such corpora involves a lot of manual extraction work, especially if one wants to allow for form variation in the idiomatic expressions (for example, extracting *cooking all the books* for *cook the books*). This work involves both the crafting of syntactic patterns to match potential idiomatic expressions and the filtering of false extractions, and increases with the amount of idiom types included in the corpus. Thus, building a large corpus of idioms, especially one that covers many types in many syntactic constructions, involves a lot of costly manual work. If a high-precision, high-recall system can be developed for the task of extracting the annotation candidates task, this cost will be greatly reduced, making the construction of a large corpus much more feasible.

As discussed in Section 2.4, idiomatic expressions mainly occur in their dictionary form, but there is a significant minority of idiom instances which occur in non-dictionary variations. This emphasises the need for corpora covering idiomatic expressions to include these variations, and for tools to be robust in dealing with variations. As such, the aim of this chapter is to describe methods and provide tools for constructing larger corpora annotated with a wide range of idiom types. In this way we hope to stimulate further research in this area. In addition, we expect that research will benefit from having larger corpora by improving evaluation quality, allowing for the training of better supervised systems, and providing additional linguistic insight into idiomatic expressions. Finally, a reliable method for detecting idiomatic expressions is not only needed for building an annotated corpus, but can also be used as part of an automatic idiom processing pipeline. In such a pipeline, extracting potentially idiomatic expressions can be seen as a first step before idiom disambiguation, and the combination of the two modules then functions as an complete idiom extraction system. These aims are summarised by the following questions, which we will answer at the end of the chapter:

1. What is the coverage of existing idiom dictionaries?

2. How can a wide-range idiom extraction system be evaluated?

3. Can a single extraction method cover a wide variety of idiom types with high accuracy?

The chapter consists of three main parts. First, we quantify the coverage and reliability of a set of idiom dictionaries, demonstrating that there is little overlap between resources (Section 4.2). Second, we develop and release an evaluation corpus for extracting potentially idiomatic expressions from text (Section 4.3)[2]. Finally, various extraction systems and combinations thereof are implemented and evaluated empirically (Section 4.4).[3]

4.2 Coverage of Idiom Inventories

Since our goal is developing a dictionary-based system for extracting potentially idiomatic expressions, we need to devise a proper method for evaluating such a system. This is not straightforward, even though the

[2] The corpus annotations are available at github.com/hslh/pie-annotation.

[3] The source code for the PIE extraction system is publicly available at github.com/hslh/pie-detection.

final goal of such a system is simple: it should extract all potentially idiomatic expressions from a corpus and nothing else, regardless of their sense and the form they are used in. The type of system proposed here hence has two aspects that can be evaluated: the dictionary that it is using as a resource for idiomatic expression, and the extractor component that identifies idioms in a corpus.

The difficulty here is that there is no 'complete' set of idiomatic expressions for English (or any other language). In theory, one could come up with a very elaborate definition of idiom and evaluate idiom dictionaries on their accuracy, but it is practically impossible to come up with a definition of idiom that leaves no room for ambiguity. This ambiguity, among others, creates a large grey area between clearly non-idiomatic phrases on the one hand (e.g. *buy a house*), and clear potentially idiomatic phrases on the other hand (e.g. *buy the farm*). As a consequence, we do not empirically evaluate the coverage of the dictionaries. Instead we quantify the divergence between various idiom dictionaries and corpora with regard to their idiom inventories. If they show large discrepancies, we take that to mean that there is either little agreement on definitions of idiom or the category is so broad that a single resource can only cover a small proportion. Conversely, if there is large agreement, we assume that idiom resources are largely reliable, and that there is consensus around what is, and what is not, an idiomatic expression, even though this consensus might not align with our goals or definition of idiom.

More concretely, we use different idiom resources and assume that the combined set of resources yields an approximation of the true set of idioms in English. A large divergence between the idiom inventories of these resources would then suggest a low recall for a single resource, since many other idioms are present in the other resources. Conversely, if the idiom inventories largely overlap, that indicates that a single resource can already yield decent coverage of idioms in the English language. The results of the dictionary comparisons are presented in Section 4.2.

4.2.1 Selected Idiom Resources

Before we report on the comparisons, we first describe why we select and how we prepare these resources. We investigate the following six idiom resources:

1. Wiktionary[4];
2. the Oxford Dictionary of English Idioms (ODEI, Ayto, 2009);
3. UsingEnglish.com (UE)[5];
4. the IDIX corpus (Sporleder et al., 2010);
5. the VNC dataset (Cook et al., 2008);
6. and the SemEval-2013 Task 5 dataset (Korkontzelos et al., 2013).

These dictionaries were mainly selected because they are available in digital format. Wiktionary and UsingEnglish have the added benefit of being freely available. However, they are both crowdsourced and lack professional editing. In contrast, ODEI is a traditional dictionary, created and edited by lexicographers, but it has the downside of not being freely available.

For Wiktionary, we extracted all idioms from the category ‘English Idioms’[6] from the English version of Wiktionary. We took the titles of all pages containing a dictionary entry and considered these idioms. Since we focus on multiword idiomatic expressions, we filtered out all single-word entries in this category. More specifically, since Wiktionary is a constantly changing resource, we used the 8,482 idioms retrieved on 10-03-2017, 15:30. We used a similar extraction method for UE, a web page containing freely available resources for ESL learners, including a list of

[4] en.wiktionary.org

[5] www.usingenglish.com/reference/idioms

[6] en.wiktionary.org/wiki/Category:English_idioms

idioms. We extracted all idioms which have publicly available definitions, which numbered 3,727 on 10-03-2017, 15:30. Again, single-word entries and duplicates were filtered out. Concerning ODEI, all idioms from the e-book version were extracted, amounting to 5,911 idioms scraped on 13-03-2017, 10:34. Here we performed an extra processing step to expand idioms containing content in parentheses, such as *a tough (or hard) nut (to crack)*. Using a set of simple expansion rules and some hand-crafted exceptions, we automatically generated all variants for this idiom, with good, but not perfect accuracy. For the example above, the generated variants are: *{a tough nut, a tough nut to crack, a hard nut, a hard nut to crack}*. The idioms in the VNC dataset are in the form `verb_noun`, e.g. `blow_top`, so they were expanded to a regular dictionary form, e.g. *blow one's top* before comparison.

4.2.2 Comparing Idiom Inventories

In many cases, using simple string-match to check overlap in idiom does not work, as exact comparison of idioms misses equivalent idioms that differ only slightly in dictionary form. Differences between resources are caused by, for example:

- inflectional variation (*crossing the Rubicon — cross the Rubicon*);
- variation in scope (*as easy as ABC — easy as ABC*);
- determiner variation (*put the damper on — put a damper on*);
- spelling variation (*mind your p's and q's — mind your ps and qs*);
- order variation (*call off the dogs — call the dogs off*);
- and different conventions for placeholder words (*recharge your batteries — recharge one's batteries*), where both *your* and *one's* can generalise to any possessive personal pronoun.

These minor variations do not fundamentally change the nature of the idiom, and we should count these types of variation as belonging to the same idiom. So, to get a good estimate of the true overlap between idiom resources, these variations need to be accounted for, which we do in our flexible matching approach.

There is one other case of variation not listed above, namely lexical variation (e.g. *rub someone up the wrong way* - *stroke someone the wrong way*). We do not abstract over this, since we consider lexical variation to be a more fundamental change to the nature of the idiom. That is, a lexical variant is an indicator of the coverage of the dictionary, where the other variations are due to different stylistic conventions and do not indicate actual coverage. In addition, it is easy to abstract over the other types of variation in an NLP application, but this is not the case for lexical variation.

The overlap counts are estimated by abstracting over all variations except lexical variation in a semi-automatic manner, using heuristics and manual checking. Potentially overlapping idioms are selected using the following set of heuristics: whether an idiom from one resource is a substring (including gaps) of an idiom in the other resource, whether the words of an idiom form a subset of the words of an idiom in the other resource, and whether there is an idiom in the other resource which has a Levenshtein ratio[7] of over 0.8. The Levenshtein ratio is an indicator of the Levenshtein distance between the two idioms relative to their length. These potential matches are then judged manually on whether they are really forms of the same idiom or not.

4.2.3 Results

The results of using exact string matching to quantify the overlap between the dictionaries is illustrated in Figure 4.1.

[7] As computed by `ratio()` from the `python-Levenshtein` package.

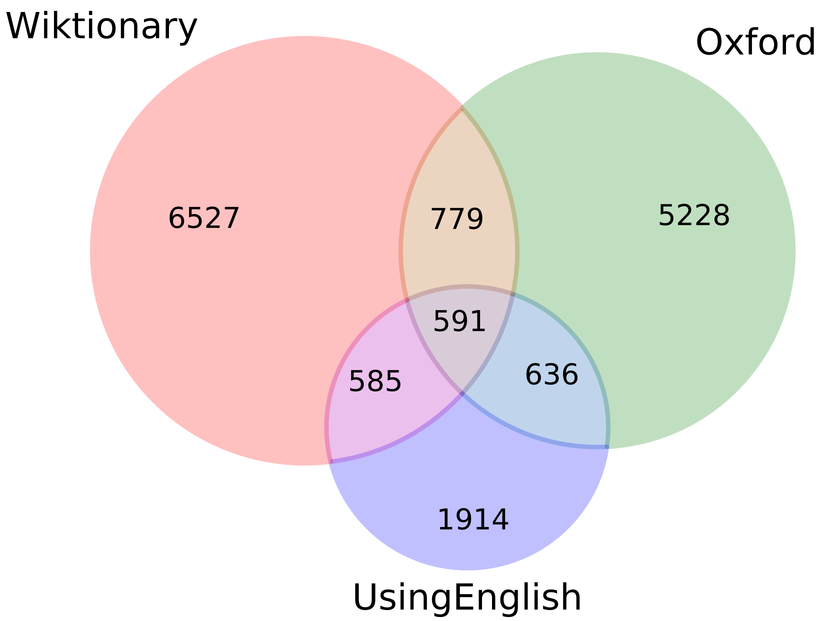


Figure 4.1: Venn diagram of case-insensitive exact string match overlap between the three idiom dictionaries.

Overlap between the three dictionaries is low. A possible explanation for this lies with the different nature of the dictionaries. Oxford is a traditional dictionary, created and edited by professional lexicographers, whereas Wiktionary is a crowdsourced dictionary open to everyone, and UsingEnglish is similar, but focused on ESL-learners. It is likely that these different origins result in different idiom inventories. Similarly, we would expect that the overlap between a pair of traditional dictionaries, such as the ODEI and the Penguin Dictionary of English Idioms (Gulland and Hinds-Howell, 2002) would be significantly higher. It should also be noted, however, that comparisons between more similar dictionaries also found relatively little overlap (Ide and Véronis, 1994; Seretan, 2008, p.99). A counterpoint is provided by Villavicencio (2005), who quantifies coverage of verb-particle constructions in three different dictionaries and finds large overlap – perhaps because verb-particle are a more restricted class.

As noted previously, using exact string matching is a very limited approach to calculating overlap. Therefore, we used heuristics and manual checking to get more precise numbers, as shown in Table 4.1, which also includes the three corpora in addition to the three dictionaries. As the manual checking only involved judging similar idioms found in pairs of resources, we cannot calculate three-way overlap as in Figure 4.1. The counts of the pair-wise overlap between dictionaries differ significantly between the two methods, which serves to illustrate the limitations of using only exact string matching and the necessity of using more advanced methods and manual effort.

Several insights can be gained from the data in Table 4.1. The relation between Wiktionary and the SemEval corpus is obvious (cf. Section 3.3.4), given the 96.92% coverage.[8] For the other dictionary-corpus pairs, the coverage increases proportionally with the size of the dictionary, except in the case of UsingEnglish and the Sporleder corpus. The proportional increase indicates no clear qualitative differences between the dictionaries, i.e. one does not have a significantly higher percentage of non-idioms than the other, when compared to the corpora.

Generally, overlap between dictionaries and corpora is low: the two biggest, ODEI and Wiktionary have only around 30% overlap, while the dictionaries also cover no more than approximately 70% of the idioms used in the various corpora. Overlap between the three corpora is also extremely low, at below 5%. This is unsurprising, since a new dataset is more interesting and useful when it covers a different set of idioms than used in an existing dataset, and thus is likely constructed with this goal in mind.

[8] One would expect 100% coverage here, but Wiktionary is an ever-changing resource and has changed since the creation of the SemEval corpus.

	Wikt.	ODEI	UE	IDIX	VNC	SemEval
Wikt.	100%	28.99%	20.60%	0.38%	0.40%	0.87%
ODEI	34.12%	100%	29.22%	0.46%	0.36%	0.69%
UE	44.73%	54.57%	100%	0.94%	0.54%	0.99%
IDIX	60.78%	60.78%	68.63%	100%	3.92%	3.92%
VNC	56.60%	45.28%	35.85%	3.77%	100%	1.89%
SemEval	96.92%	70.77%	52.31%	3.08%	1.54%	100%

Table 4.1: Percentage of overlapping idioms between different idiom resources, abstracting over minor variations. Value is the number of idioms in the intersection of two idiom sets, as a percentage of number of idioms in the resource in the left column. For example, 56.60% of idioms in the VNC occur in Wiktionary.

4.3 Corpus Annotation

We exhaustively annotate an evaluation corpus with all instances of a predefined set of PIEs, so we can use it to evaluate PIE extraction methods. As part of this, we come up with a workable definition of PIEs, and measure the reliability of PIE annotation by inter-annotator agreement.

Assuming that we have a set of idioms, the main problem of defining what is and what is not a potentially idiomatic expression is caused by variation. In principle, a potentially idiomatic expression is an instance of a phrase that, when seen without context, could have either an idiomatic or a literal meaning. This is clearest for the dictionary form of the idiom, as in Example 16. Literal uses generally allow all kinds of variation, but not all of these variations allow a figurative interpretation, e.g. Example 17. However, how much variation an idiom can undergo while retaining its figurative interpretation is different for each expression, and judgements of this might vary from one speaker to the other. An example of this is *spill the bean*, a variant of *spill the beans*, in Example 18 judged by Fazly et al. (2009, p.65) as being highly questionable. However, even here a corpus example can be found containing the same variant used in a figurative sense (Example 19). As such, we assume that we cannot know a priori which variants of an expression allow a figurative reading, and are thus a potentially idiomatic expression. Therefore we consider every possible morphosyntactic variation of an idiom a PIE, regardless of whether it actually allows a figurative reading. We believe the boundaries of this variation can only be determined based on corpus evidence, and even then they are likely variable.

(16) John **kicked the bucket** last night.

(17) *__The bucket__, John **kicked** last night.

(18) ??Azin **spilled the bean**. (from Fazly et al., 2009, p.65)

(19) Alba reveals Fantastic Four 2 details[.] The Invisible Woman actress **spills the bean** on super sequel. (from ukWaC)

4.3.1 Evaluating PIE Extraction

Evaluating the extraction methods is easier than evaluating dictionary coverage, since the goal of the extraction component is more clearly delimited: given a set of PIEs from one or more dictionaries, extract all occurrences of those PIEs from a corpus. Thus, rather than dealing with the undefined set of all PIEs, we can work with a clearly defined and finite set of PIEs from a dictionary.

Because we have a clearly defined set of PIEs, we can exhaustively annotate a corpus for PIEs, and use that annotated corpus for automatic evaluation of extraction methods using recall and precision. This allows us to facilitate and speed up annotation by pre-extracting sentences possibly containing a PIE. After the corpus is annotated, the precision and recall can be easily estimated by comparing the extracted PIE instances to those marked in the corpus. The details of the corpus selection, dictionary selection, extraction heuristic and annotation procedure are presented in Section 4.3.4, and the details and results of the various extraction methods are presented in Section 4.4.

4.3.2 Base Corpus and Idiom Selection

As a base corpus, we use the XML version of the British National Corpus (BNC Consortium, 2007), because of its size, variety, and wide availability.[9] The BNC is pre-segmented into *s-units*, which we take to be sentences, *w-units*, which we take to be words, and *c-units*, punctuation. We then extract the text of all w-units and c-units. We keep the sentence segmentation, resulting in a set of plain text sentences. All sentences are included, except for sentences containing `<gap>` elements, which are filtered out. These `<gap>` elements indicate places where material from the original has been left out, e.g. for anonymisation purposes. Since this can result in incomplete sentences that cannot be parsed correctly, we filter out sentences containing these gaps.

We use only the written part of the BNC. From this, we extract a set of documents with the aim of having as much genre variation as possible. To achieve this, we select the first document in each genre, as defined by the `classCode` attribute (e.g. `nonAc`, `commerce`, `letters`). The resulting set of 46 documents makes up our base corpus. Note that these documents vary greatly in size, which means the resulting corpus is varied, but not balanced in terms of size (Table 4.2). The documents are split across a development and test set, as specified at the end of Section 4.3.4. We exclude documents with IDs starting with `A0` from all annotation and evaluation procedures, as these were used during development of the extraction tool and annotation guidelines.

As for the set of potentially idiomatic expressions, we use the intersection of the three dictionaries, Wiktionary, Oxford, and UsingEnglish. Based on the assumption that, if all three resources include a certain id-

[9] The British National Corpus is freely available under the BNC User Licence at ota.ox.ac.uk/desc/2554.

	Documents	Tokens	Shortest Doc.	Longest Doc.
Dev.	22	832,083	1,815	228,230
Test	23	814,125	1,984	231,846
Total	45	1,646,208	1,815	231,846

Table 4.2: Statistics on the size of the BNC documents used for PIE annotation and the split in development and test set.

iom, it must unquestionably be an idiom, we choose the intersection (see Figure 4.1). This serves to exclude questionable entries, like *at all*, which is in Wiktionary, even though it is not a PIE. The final set of idioms used for these experiments consists of 591 different multiword expressions.

4.3.3 Extraction of PIE Candidates

To annotate the corpus completely manually would require annotators to read the whole corpus, and cross-reference each sentence to a list of almost 600 PIEs, to check whether one of those PIEs occurs in a sentence. Since this is not a realistic annotation setting, both in terms of difficulty and time cost, we use a pre-extraction step to present candidates for annotation to the human annotators.

Given the corpus and the set of PIEs, we heuristically extract the PIE candidates as follows: given an idiomatic expression, extract every sentence which contains all the defining words of the idiom, in any form. This ensures that all possibly matching sentences get extracted, while greatly pruning the amount of sentences for annotators to look at. In addition, it allows us to present the heuristically matched PIE type and corresponding words to the annotators, which makes it much easier to judge whether something is a PIE or not. This also means that annotators never have to go through the full list of PIEs during the annotation process.

Initially, the heuristic simply extracted any sentence containing all the required words, where a word is any of the inflectional variants of the words in the PIE, except for determiners and punctuation. This method produced large amounts of noise, that is, a set of PIE candidates with only a very low percentage of actual PIEs. This was caused by the presence of some highly frequent PIEs with very little defining lexical content, such as *on the make*, and *in the running*. For example, with the original method, every sentence containing the preposition *on*, and any inflectional form of the verb *make* was extracted, resulting in a huge number of non-PIE candidates.

To limit the amount of noise, two restrictions were imposed. The first restrictions disallows word order variation for PIEs which do not contain a verb. The rationale behind this is that word order variation is only possible with PIEs like *spill the beans* (e.g. *the beans were spilled*), and not with PIEs like *in the running* (**the running in??*). The second restriction is that we limit the number of words that can be inserted between the words of a PIE, but only for PIEs like *on the make*, and *in the running*, i.e. PIEs which only contain prepositions, determiners and a single noun. The number of intervening words was limited to three tokens, allowing for some variation, as in Example 20, but preventing sentences like Example 21 from being extracted. This restriction could result in the loss of some PIE candidates with a large number of intervening words. However, the savings in annotation time clearly outweigh the small loss in recall in this situation.

(20) Either at New Year or before July you can anticipate a change **in the** everyday **running** of your life. (*in the running* - BNC - document CBC - sentence 458)

(21) [..] if [he] hung around near the goal or **in the** box for that matter instead of **running** all over the show [..] (*in the running* - BNC - document J1C - sentence 1341)

4.3.4 Annotation Procedure

The manual annotation procedure consists of three different phases (pilot, double annotation, single annotation), followed by an adjudication step to resolve conflicting annotations.Two things are annotated: whether something is a PIE or not, and if it is a PIE, which sense the PIE is used in. In the first phase (`0-100-*`), we randomly select 100 of the 2208 PIE candidates which are then annotated by three annotators. All annotators have a good command of English, are computational linguists, and familiar with the subject.

The annotators were provided with a short set of guidelines, of which the main rule-of-thumb for labelling a phrase as a PIE is as follows: any phrase is a PIE when it contains all the words, with the same part-of-speech, and in the same grammatical relations as in the dictionary form of the PIE, ignoring determiners.[10]

For sense annotation, annotators were to mark a PIE as idiomatic if it had a sense listed in one of the idiom dictionaries, and as literal if it had a meaning that is a regular composition of its component words. For cases which were undecidable due to lack of context, the *?*-label was used. The *other*-label was used as a container label for all cases in which neither the literal or idiomatic sense was correct (e.g. meta-linguistic uses and embeddings in larger metaphorical frames, as in Examples 13 and 14).

The first phase of annotation serves to bring to light any inconsistencies between annotators and fill in any gaps in the annotation guidelines. The resulting annotations already show a reasonably high agreement of 0.74 Fleiss' Kappa. Table 4.3 shows annotation details and agreement statistics for all three phases.

In the second phase of annotation (`100-600-* & 600-1100-*`), another 1000 of the 2208 PIE candidates are selected to be annotated by two pairs of annotators.annotators each. This shows very high agree-

[10] Note that, while not exactly the same relation, we do allow for passivisation, e.g. 'The trick was done by using a new approach' for *do the trick*.

Annotation Task	Annotators	Candidates	% Agr.	Fleiss' κ
0-100-PIE	3	100	0.87	0.74
100-600-PIE	2	500	0.96	0.91
600-1100-PIE	2	500	0.94	0.88
1100-2208-PIE	1	1108	n/a	n/a
0-100-sense	3	38	0.82	0.65
100-600-sense	2	160	0.92	0.83
600-1100-sense	2	259	0.79	0.63
1100-2208-sense	1	527	n/a	n/a

Table 4.3: Details of the annotation phases and inter-annotator agreement statistics. The number of candidates for sense annotation is the number on which all annotators initially agreed that it was a PIE, i.e. pre-adjudication.The annotation tasks suffixed by '-PIE' indicate agreement on PIE/non-PIE annotation and the tasks suffixed by '-sense' indicate agreement on sense annotation for PIEs. Note that sense and PIE annotation are split here for clarity of presentation; in practice they were annotated as a joint task.

ment, as shown in Table 4.3. This is probably due to the improvement in guidelines and the discussion following the pilot round of annotation. The exception to this are the somewhat lower scores for the `600-1100-sense` annotation task. Adjudication revealed that this is due almost exclusively because of a different interpretation of the literal and idiomatic senses of a single PIE type: *on the ground*. Excluding this PIE type, Fleiss' Kappa increases from 0.63 to 0.77.

Because of the high agreement on PIE annotation, we deem it sufficient for the remainder (1108 candidates) to be annotated by only the primary annotator in the third phase of annotation (`1100-2208-*`). The reliability of the single annotation can be checked by comparing the distribution of labels to the multi-annotated parts. This shows that it falls clearly within the ranges of the other parts, both in the proportion of PIEs and idiomatic senses. The single-annotated part has 49.0% PIEs, which is only 4 percentage points above the 44.7% PIEs in the multi-annotated parts, while the proportion of idioms is just 2 percentage points higher, with 55.9% versus 53.9.%

Part	Cands.	PIE (y/n)	% PIE	Sense (y/n/o)	% Idiom
0-100	100	47/53	47.0	23/24/0	48.9
100-600	500	169/331	33.4	112/54/3	66.3
600-1100	500	276/224	55.2	130/132/14	47.1
1100-2208	1108	527/581	47.6	280/230/17	53.1
Total	2208	1019/1189	46.2	545/440/34	53.5

Table 4.4: Distributional statistics on the annotated PIE corpus, post-adjudication. Adjudication resolved all instances which were disagreed upon and all *?*-sense-labels, hence the presence of only 3 sense labels: i(diomatic), l(iteral), and o(ther).

Although inter-annotator agreement was high, there was still a significant number of cases in the triple and double annotated PIE candidate sets where not all annotators agreed. These cases were adjudicated through discussion by all annotators, until they were in agreement. In addition, all PIE candidates which initially received the *?*-label (unclear or undecidable) for sense or PIE were resolved in the same manner. In the adjudication procedure, annotators were provided with additional context on each side of the idiom, in contrast to the single sentence provided during the initial annotation. The main reason to do adjudication, rather than simply discarding all candidates for which there was disagreement, was that we expected exactly those cases for which there are conflicting annotations to be the most interesting ones, since having non-standard properties would cause the annotations to diverge. Examples of such interesting non-standard cases are *at sea* as part of a larger satirical frame in Example 22 and *cut the mustard* in Example 23 where it is used in a headline as wordplay on a Cluedo character.

(22) The bovine heroine has connections with Cowpeace International, and deals with a huge treacle slick **at sea**. (*at sea* - BNC - docu-

ment CBC - sentence 13550)

(23) Why not **cut the Mustard**? [..] WADDINGTON Games's proposal to axe Reverend Green from the board game Cluedo is a bad one. (*cut the mustard* - BNC - document CBC - sentence 14548)

We split the corpus at the document level. The corpus consists of 45 documents from the BNC, and we split it in such a way that both the development and test set have 1104 candidates, spread across 22 or 23 different documents. During development it became apparent that for four PIE types (*no go, have a go, by and large, by the by*), the pre-extraction had failed. As such, no instances of these PIEs were extracted or annotated. Rather than excluding these PIEs from consideration, we decided to correct the extraction mistake and annotate the instance of these types post-hoc. 31 instances were extracted and annotated by the primary annotator, yielding a total of 2239 candidates, of which 1112 from 22 documents form the development set, and the other 1127 from 23 documents make up the test set.

4.4 Dictionary-based PIE Extraction

We propose and implement four different extraction methods, of differing complexities: exact string match, fuzzy string match, inflectional string match, and parser-based extraction. Because of the absence of existing work on this task, we compare these methods to each other, where the more basic methods function as baselines. Below, each of the extraction methods is presented and discussed in detail.

4.4.1 String-based Extraction Methods

Exact String Match This is, very simply, extracting all instances of the exact dictionary form of the PIE, from the tokenised text of the corpus. Word boundaries are taken into account, so *at sea* does not match 'th**at**

seawater'. As a result, all inflectional and other variants of the PIE are ignored.

Fuzzy String Match Fuzzy string match is a rough way of dealing with morphological inflection of the words in a PIE. We match all words in the PIE, taking into account word boundaries, and allow for up to 3 additional letters at the end of each word. These 3 additional characters serve to cover inflectional suffixes.

Inflectional String Match In inflectional string match, we aim to match all inflected variations of a PIE. This is done by generating all morphological variants of the words in a PIE, generating all combinations of those words, and then using exact string match as described earlier.

Generating morphological variations consists of three steps: part-of-speech tagging, morphological analysis, and morphological re-inflection. Since inflectional variation only applies to verbs and nouns, we use the Spacy[11] part-of-speech tagger to detect the verbs and nouns. Then, we apply the morphological analyser `morpha` to get the base, uninflected form of the word, and then use the morphological generation tool `morphg` to get all possible inflections of the word. Both tools are part of the Morph morphological processing suite (Minnen et al., 2001). Note that the Morph tools depend on the part-of-speech tag in the input, so that a wrong PoS may lead to an incorrect set of morphological variants.

For a PIE like *spill the beans*, this results in the following set of variants: {*spill the bean, spills the bean, spilled the bean, spilling the bean, spill the beans, spills the beans, spilled the beans, spilling the beans*}. Since we generate up to 2 variants for each noun, and up to 4 variants for each verb, the number of variants for PIEs containing multiple verbs and nouns can get quite large. On average, 8 additional variants are generated for each potentially idiomatic expression.

[11] spacy.io

Additional Steps For all string match-based methods, ways to improve performance are implemented, to make them as competitive as possible. Rather than doing exact string matching, we also allow words to be separated by something other than spaces, e.g. *nuts-and-bolts* for *nuts and bolts.* Additionally, there is an option to take into account case distinctions. With the *case-sensitive* option, case is preserved in the idiom lists, e.g. *coals to Newcastle*, and the string matching is done in a case-sensitive manner. This increases precision, e.g. by avoiding PIEs as part of proper names, but also comes at a cost of recall, e.g. for sentence-initial PIEs. Thirdly, there is the option to allow for a certain number of intervening words between each pair of words in the PIE. This should improve recall, at the cost of precision. For example, this would yield the true positive *make a huge mountain out of a molehill* for *make a mountain out of a molehill*, but also false positives like *have a smoke and go* for *have a go.*

A third shared property of the string-based methods is the processing of placeholders in PIEs. PIEs containing possessive pronoun placeholders, such as *one's* and *someone's* are expanded. That is, we remove the original PIE, and add copies of the PIE where the placeholder is replaced by one of the possessive personal pronouns. For example, *a thorn in someone's side* is replaced by *a thorn in {my, your, his, ...} side.* In the case of *someone's*, we also add a wildcard for any possessively used word, i.e. *a thorn in —'s side*, to match e.g. *a thorn in Google's side.* Similarly, we make sure that PIE entries containing —, such as *the mother of all* —, will match any word for — during extraction. We do the same for *someone*, for which we substitute objective pronouns. For *one*, this is not possible, since it is too hard to distinguish from the *one* used as a number.

4.4.2 Parser-Based Extraction Methods

Parser-based extraction is potentially the extraction method with the widest coverage, since it has the capacity to extract both morphological and syntactic variants of the PIE. This should be robust against the most common modifications of the PIE, e.g. through word insertions (*spill all the beans*), passivisation (*the beans were spilled*), and abstract over articles (*spill beans*).

In this method, PIEs are extracted using the assumption that any sentence which contains the lemmata of the words in the PIE, in the same dependency relations as in the PIE, contains an instance of the PIE type in question. More concretely, this means that the parse of the sentence should contain the parse tree of the PIE as a subtree. This is illustrated in Figure 4.2, which shows the parse tree for the PIE *lose the plot*, parsed without context. Note that this is a subtree of the parse tree for the sentence 'you might just lose the plot completely', which is shown in Figure 4.3. Since the sentence parse contains the parse of the PIE, we can conclude that the sentence contains an instance of that PIE and extract the span of the PIE instance.

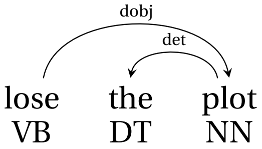


Figure 4.2: Automatic dependency parse of the PIE *lose the plot*.

All PIEs are parsed in isolation, based on the assumption that all PIEs can be parsed, since they are almost always well-formed phrases. However, not all PIEs will be parsed correctly, especially since there is no context to resolve ambiguity. Errors tend to occur at the part-of-speech level, where, for example, verb-object combinations like *jump ship* and *touch wood* are erroneously tagged as noun-noun compounds. An analysis of the impact of parser error on PIE extraction performance is presented in Section 4.4.4. Initially, we use the Spacy parser for parsing both the PIEs and the sentences.

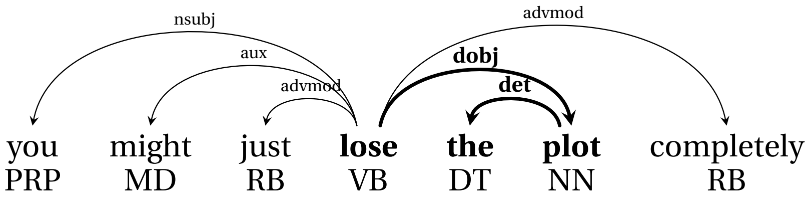


Figure 4.3: Automatic dependency parse of the sentence 'you might just lose the plot completely', which contains the PIE *lose the plot.* From BNC document CH1, sentence 829. Sentence shortened for display convenience.

Next, the sentence is parsed, and the lemma of the top node of the parsed PIE is matched against the lemmata of the sentence parse. If a match is found, the parse tree of the PIE is matched against the subtree of the matching sentence parse node. If the whole PIE parse tree matches, the span ranging from the first PIE token to the last is extracted. This span can thus include words that are not directly part of the PIE's dictionary form, in order to account for insertions like *ships **were** jumped* for *jump ship*, or *have a **big** heart* for *have a heart.*

During the matching, articles (*a*/*an*/*the*) are ignored[12], and passivisation is accounted for. In addition, a number of special cases are dealt with. These are PIEs containing *someone('s), something('s), one's*, or —. These words are used in PIEs as placeholders for a generic possessor (*someone's/something's/one's*), generic object (*someone/something*), or any word of the right PoS (—).

For *someone's*, and *something's*, we match any possessive pronoun, or (proper) noun + possessive marker. For *one's*, only possessive pronouns are matched, since this is a placeholder for reflexive possessors.

[12] As articles can be inherent parts of idiomatic expressions, we have also tested our method taking articles into account during matching. However, results were lower overall than when ignoring articles. When matching articles, the regular parsing-based method achieves an F1-score of 84.56%, and the in-context parsing-based method achieves an F1-score of 86.43%.

For *someone* and *something*, any non-possessive pronoun or (proper) noun is matched.

For — wildcards, any word can be matched, as long as it has the right relation to the right head. An additional challenge with these wildcards is that PIEs containing them cannot be parsed, e.g. *too — for words* is not parseable. This is dealt with by substituting the — by a PoS-ambiguous word, such as *fine*, or *back*.

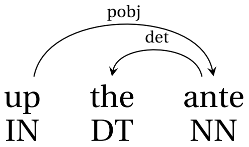


Figure 4.4: Automatic dependency parse of the PIE *up the ante*.

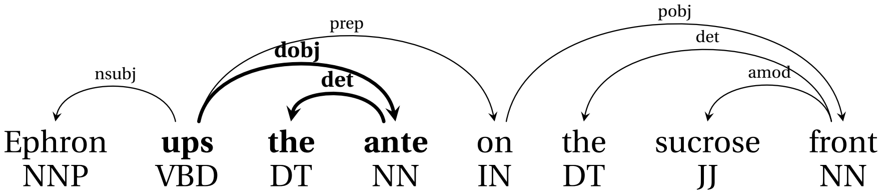


Figure 4.5: Automatic dependency parse of the sentence ‘Ephron ups the ante on the sucrose front’, which contains the PIE *up the ante*. From BNC document CBC, sentence 7022. Sentence shortened for display convenience.

Two optional features are added to the parser-based method with the goal of making it more robust to parser errors: generalising over dependency relation labels, and generalising over dependency relation direction. We expect this to increase recall at the cost of precision. In the first *no labels* setting, we match parts of the parse tree which have the same head lemma and the same dependent lemma, regardless of the relation label. An example of this is Figure 4.4, which has the wrong relation label between *up* and *ante*. If labels are ignored, however, we can still extract the PIE instance in Figure 4.5, which has the correct label. In the *no directionality* setting, relation labels are also ignored, and in addition the directionality of the relation is ignored, that is, we allow for the reversal of heads and dependents. This benefits performance in a case like Figure 4.7, which has *stock* as the head of *laughing* in a *compound* relation, whereas the parse of the PIE (Figure 4.6) has *laughing* as the head of *stock* in a *dobj* relation.

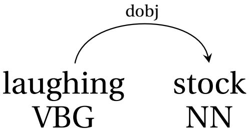


Figure 4.6: Automatic dependency parse of the PIE *laughing stock*.

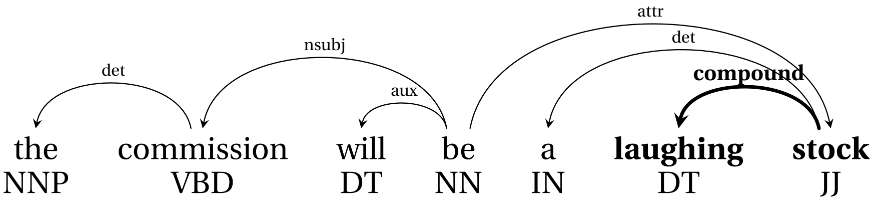


Figure 4.7: Automatic dependency parse of the sentence ‘the commission will be a laughing stock’, which contains the PIE *laughing stock*. From BNC document A69, sentence 487. Sentence shortened for display convenience.

Since the parser-based method parses PIEs without any context, it often finds an incorrect parse, as for *jump ship* in Figure 4.8. As such, we add an option to the method that aims to increase the number of correct parses by parsing the PIE within context, that is, within a sentence. This can greatly help to disambiguate the parse, as in Figure 4.9. If the number of correct parses goes up, the recall of the extraction method should also increase. Naturally, it can also be the case that a PIE is parsed correctly without context, and incorrectly with context. However, we expect the gains to outweigh the losses.

The challenge here is thus to collect example sentences containing the PIE. Since the whole point of this work is to extract PIEs from raw text, this provides a catch-22-like situation: we need to extract a sentence containing a PIE in order to extract sentences containing a PIE.

The workaround for this problem is to use the exact string matching method with the dictionary form of the PIE and a very large plain text corpus to gather example sentences. By only considering the exact dictionary form we both simplify the finding of example sentences and the extraction of the PIE's parse from the sentence parse.

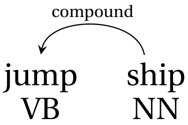


Figure 4.8: Automatic dependency parse of the PIE *jump ship*.

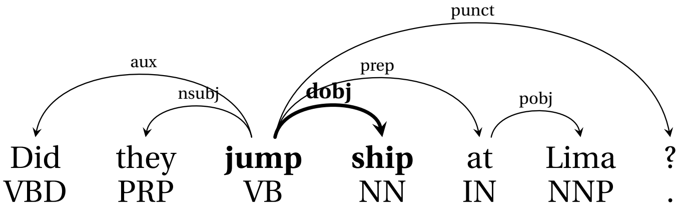


Figure 4.9: Automatic dependency parse of the extracted sentence 'Did they jump ship at Lima?' containg the PIE *jump ship*.

In case multiple example sentences are found, the shortest sentence is selected, since we assume it is easiest to parse. This is also the reason we make use of very large corpora, to increase the likelihood of finding a short, simple sentence. The example sentence extraction method is modified in such a way that sentences where the PIE is used metalinguistically in quotes, e.g. “the well-known English idiom ‘to spill the beans’ has no equivalents in other languages”, are excluded, since they do not provide a natural context for parsing. When no example sentence can be found in the corpus, we back-off to parsing the PIE without context. After a parse has been found for each PIE (i.e. with or without context), the method proceeds identically to the regular parser-based method.

We make use of the combination of two large corpora for the extraction of example sentences: the English Wikipedia[13], and ukWaC (Ferraresi et al., 2008). For the Wikipedia corpus, we use a dump (13-01-2016) of the English-language Wikipedia, and remove all Wikipedia markup. This is done using WikiExtractor[14]. The resulting files still contain some mark-up, which is removed heuristically. The resulting corpus contains mostly clean, raw, untokenised text, numbering approximately 1.78 billion tokens.

As for ukWaC, all XML-markup was removed, and the corpus is converted to a one-sentence-per-line format. UkWaC is tokenised, which makes it difficult for a simple string match method to find PIEs containing punctuation, for example *day in, day out.* Therefore, all spaces before commas, apostrophes, and sentence-final punctuation are removed. The resulting corpus contains approximately 2.05 billion tokens, making for a total of 3.83 billion tokens in the combined ukWaC and Wikipedia corpus.

4.4.3 Results

In order to determine which of the methods described previously produces the highest quality extraction of potentially idiomatic expressions,

[13] dumps.wikimedia.org/enwiki/20160113/

[14] medialab.di.unipi.it/wiki/Wikipedia_Extractor

we evaluate them, in various settings, on the corpus described in Section 4.3.

For parser-based extraction, systems with and without in-context parsing, ignoring labels, and ignoring directionality are tested. For the three string-based extraction methods, varying numbers of intervening words and case sensitivity are evaluated. Evaluation is done using the development set, consisting of 22 documents and 1112 PIE candidates, and the test set, which consists of 23 documents and 1127 PIE candidates. For each method the best set of parameters and/or options is determined using the development set, after which the best variant by F1-score of each method is evaluated on the test set.

Since these documents in the corpus are exhaustively annotated for PIEs, we can calculate true and false positives, and false negatives, and thus precision, recall and F1-score. The exact spans are ignored, because slight differences there can grossly distort the results of the evaluation. Rather, we count an extraction as a true positive if it finds the correct PIE type in the correct sentence.

Note that we judge the system with the highest F1-score to be the best-performing system, since it is a clear and objective criterion. However, when using the system in practice, the best performance depends on the goal. When used as a preprocessing step for PIE disambiguation, the system with the highest F1-score is perhaps the most suitable, but as a corpus building tool, one might want to sacrifice some precision for an increase in recall. This helps to get the most comprehensive annotation of PIEs possible, without overloading the annotators with false extractions (i.e. non-PIEs).

The results for each system on the development set are presented in Tables 4.5 and 4.6. Generally, results are in line with expectations: (the best) parse-based methods are better than (the best) string-based methods, and within string-based methods, inflectional matching works best. The same goes for the different settings: case-sensitivity increases precision at the cost of recall, allowing intervening words increases recall at

	0 words			1 word			2 words			3 words		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Exact	92.80	59.19	72.28	90.73	66.54	**76.78**	83.48	67.83	74.85	77.29	**68.20**	72.46
Exact-CS	**97.35**	54.04	69.50	94.83	60.66	73.99	87.73	61.76	72.49	81.25	62.13	70.42
Fuzzy	64.26	68.75	66.43	37.53	76.65	50.39	21.50	77.39	33.65	14.42	**77.02**	24.29
Fuzzy-CS	**75.33**	62.87	68.54	69.51	70.40	**69.95**	59.06	71.88	64.84	51.17	72.24	59.91
Inflect	89.79	71.14	79.38	87.10	80.70	**83.78**	80.11	82.90	81.48	73.66	**83.27**	78.17
Inflect-CS	**93.90**	65.07	76.87	90.74	73.90	81.46	83.94	75.92	79.73	77.57	76.29	76.92

Table 4.5: PIE extraction performance (precision/recall/F1-score) of the three string-based systems, with different options, on the development set. The number of words indicates the number of intervening words allowed between the parts of the PIE for matching to occur. *CS* indicates case-sensitive string matching. The best score for each metric and system is in **bold**.

	Regular			No Labels			No Directionality		
	P	R	F1	P	R	F1	P	R	F1
Parsing-In-Isolation	**90.83**	80.15	85.16	80.00	84.56	82.22	51.40	87.68	64.81
Parsing-In-Context	89.79	87.32	**88.54**	55.29	89.34	68.31	39.61	**90.44**	55.10

Table 4.6: PIE extraction performance (precision/recall/F1-score) of the parser-based system, with different options, on the development set. The best score for each metric is in **bold**.

the cost of precision, and the same goes for the *no labels* and *no directionality* options for parser-based extraction. Overall, in-context parser-based extraction works best, with an F1 of 88.54%, whereas fuzzy matching does very poorly.

Within string-based methods, exact matching has the highest precision, but low recall. Fuzzy matching increases recall at a disproportionately large precision cost, whereas inflectional matching combines the best of both worlds and has high recall at a small loss in precision. For the parser-based system, it is notable that parsing idioms within context yields a clear overall improvement by greatly improving recall at a small cost in precision, likely due to a reduction in parsing error.

	Precision	Recall	F1-score
Exact-1Word	**92.66**	59.88	72.75
Fuzzy-CS-1Word	60.19	61.86	61.01
Inflect-1Word	87.76	73.72	80.13
Parsing-In-Context	90.78	**87.55**	**89.13**

Table 4.7: PIE extraction performance of the best variant by F1-score of each of the four systems, on the test set. The best score for each metric is in **bold**.

We evaluate the best variant of each system, as determined by F1-score, on the test set. This gives us an indication of whether the system is robust enough, or was overfitted on the development data. Results on the test set are shown in Table 4.7. On average, the results are lower than the results on the development set. The string-based methods perform clearly worse, with drops of about 4% F1-score for exact and inflectional match, and a large drop of almost 9% F1-score for fuzzy matching. The parser-based method, on the other hand, is more robust, with a small 0.59% increase in F1-score on the test set.

4.4.4 Analysis

Broadly speaking, the PIE extraction systems presented above perform in line with expectations. It is nevertheless useful to see where the best-performing system misses out, and where improvements like in-context parsing help performance. We analyse the shortcomings of the in-context parser-based system by looking at the false positives and false negatives on the development set. We consider the output of the system with best overall performance, since it will provide the clearest picture.

The system extracts 529 PIEs in total, of which 54 are false extractions (false positives), and it misses 69 annotated PIE instances (false negatives). Most false positives stem from the system's failure to capture nuances of PIE annotation. This includes cases where PIEs contain, or are part of, proper nouns (Example 24), PIEs that are part of coordination constructions (Example 25), and incorrect attachments (Example 26). Among these errors, sentences containing proper nouns are an especially frequent problem.

(24) Drama series include [..] airline security thrills **in Cleared** For Takeoff and Head Over Heels [..] (*in the clear* - BNC - document CBC - sentence 5177)

(25) They prefer silk, satin or lace underwear **in tasteful black** or ivory. (*in the black* - BNC - document CBC - sentence 14673)

(26) [..] 'I saw this chap make something **out of an ordinary piece of wood** — he fashioned it into an exquisite work of art.' (*out of the woods* - BNC - document ABV - sentence 1300)

The main cause of false negatives are errors made by the parser. In order to correctly extract a PIE from a sentence, both the PIE and the sentence have to be parsed correctly, or at least parsed in the same way. This means a missed extraction can be caused by a wrong parse for the PIE or a wrong parse for the sentence. These two error types form the largest class of false negatives. Since some PIE types are rather frequent, a wrong parse for a single PIE type can potentially lead to a large number of missed extractions.

It is not surprising that the parser makes many mistakes, since idioms often have unusual syntactic constructions (e.g. *come a cropper*) and contain words where default part-of-speech tags lead to the wrong interpretation (e.g. *round* is a preposition in *round the bend*, not a noun or adjective). This is especially true when idioms are parsed without context, and hence, where in-context parsing provides the largest benefit: the number of PIEs which are parsed incorrectly drops, which leads to F1-scores on those types going from 0% to almost 100% (e.g. *in light of* and *ring a bell*). Since parser errors are the main contributor to false negatives, hurting recall, we can observe that parsing idioms in context serves to benefit only recall, by 7 percentage points, at only a small loss in precision.

We find that adding context mainly helps for parsing expressions which are structurally relatively simple, but still ambiguous, such as *rub shoulders*, *laughing stock*, and *round the bend*. Compare, for example, the parse trees for *laughing stock* in isolation and within the extracted context sentence in Figures 4.10 and 4.11. When parsed in isolation, the relation between the two words is incorrectly labelled as a compound relation, whereas in context it is correctly labelled as a direct object relation. Note however, that for the most difficult PIEs, embedding them in a context does solve the parsing problem: a syntactically odd phrase is hard to phrase (e.g. *for the time being*), and a syntactically odd phrase in a sentence makes for a syntactically odd sentence that is still hard to parse (e.g. ‘London for the time being had been abandoned.’). Finding example sentences turned out not to be a problem, since appropriate sentences were found for 559 of 591 PIE types.

An alternative method for reducing parser error is to use a different, better parser. The Spacy parser was mainly chosen for implementation convenience and speed, and there are parsers which have better per-

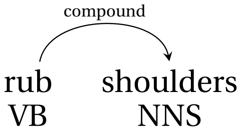


Figure 4.10: Automatic dependency parse of the PIE *rub shoulders*.

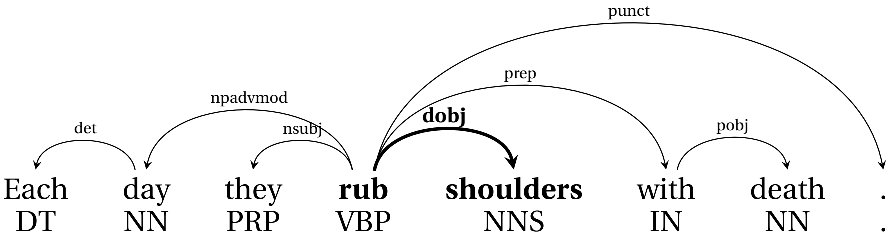


Figure 4.11: Automatic dependency parse of the extracted sentence 'Each day they rub shoulders with death.' containing the PIE *rub shoulders*.

formance, as measured on established parsing benchmarks. To investigate the effectiveness of this method, we used the Stanford Neural Dependency Parser (Chen and Manning, 2014) to extract PIEs in the regular parsing, in-context parsing and the *no labels* settings. In all cases, using the Stanford parser yielded worse extraction performance than the Spacy parser. A possible explanation for why a supposedly better parser performs worse here is that parsers are optimised and trained to do well on established benchmarks, which consist of complete sentences, often from news texts. This does not necessarily correlate with parsing performance on short (sentences containing) idiomatic phrases. As such, we cannot assume that better overall parsing performance implies PIE extraction performance.

It should be noted that, when assessing the quality of PIE extraction performance, the parser-based methods are sensitive to specific PIE types. That is, if a single PIE type is parsed incorrectly, then it is highly probable that all instances of that type are missed. If this type is also highly frequent, this means that a small change in actual performance yields a large change in evaluation scores. Our goal is to have a PIE extraction system that is robust across all PIE types, and thus the current evaluation setting does not align exactly with our aim.

Splitting out performance per PIE type reveals whether there is indeed a large variance in performance across types. Table 4.8 shows the 25 most frequent PIE types in the corpus, and the performance of the in-context-parsing-based system on each. Except two cases (*in the black* and *round the bend*), we see that the performance is in the 80–100% range, even showing perfect performance on the majority of types.

For none of the types do we see low precision paired with high recall, which indicates that the parser never matches a highly frequent non-PIE phrase. For the system with the *no labels* and *no-directionality* options (per-type numbers not shown), however, this does occur. For example, ignoring the labels for the parse of the PIE *have a go* leads to the erroneous matching of many sentences containing a form of *have to go*, which is highly frequent, thus leading to a large drop in precision.

Although performance is stable across the most frequent types, it is more spotty among the less frequent types. This hurts overall performance, and there are potential gains in mitigating the poor performance on these types, such as *for the time being*. At the same time, the string matching methods show much more stable performance across types, and some of them do so with very high precision. As such, a combination of two such methods could boost performance significantly. If we use a high-precision string match-based method, such as the exact string match variant with a precision of 97.35%, recall could be improved for the wrongly parsed PIE types, without a significant loss of precision.

We experiment with two such combinations, by simply taking the union of the sets of extracted idioms of both systems, and filtering out duplicates. Results are shown in Table 4.9. Both combinations show the expected effect: a clear gain in recall at a minimal loss in precision. Com-

PIE Type	Count	Precision	Recall	F1-score
on the ground	48	96.00	100.00	97.96
on board	24	100.00	83.33	90.91
on the cards	18	94.74	100.00	97.30
at sea	15	93.33	93.33	93.33
in someone's pocket	13	90.91	76.92	83.33
in the hole	9	100.00	100.00	100.00
all along	9	100.00	100.00	100.00
all over the place	9	100.00	100.00	100.00
under fire	8	100.00	87.50	93.33
in light of	8	100.00	87.50	93.33
on the level	8	100.00	100.00	100.00
over the top	7	100.00	100.00	100.00
on edge	7	100.00	100.00	100.00
at the end of the day	7	100.00	100.00	100.00
ring a bell	6	100.00	66.67	80.00
in the bag	6	85.71	100.00	92.31
in the running	6	100.00	83.33	90.91
up for grabs	6	100.00	100.00	100.00
on the rocks	5	100.00	100.00	100.00
in the black	5	40.00	40.00	40.00
out of the blue	5	100.00	100.00	100.00
round the bend	5	100.00	40.00	57.14
behind bars	5	100.00	100.00	100.00
have a go	5	71.43	100.00	83.33
turn the corner	4	100.00	100.00	100.00

Table 4.8: Extraction performance of the in-context-parsing-based system on each of the 25 most frequent PIE types in the corpus.

pared to the in-context-parsing-based system, the combination with exact string matching yields a gain in recall of over 6%, and the combination with inflectional string matching yields an even bigger gain of almost 8%, at precision losses of 0.6% and 0.8%, respectively. This indicates that the systems are very much complementary in the PIEs they extract. It also means that, when used in practice, combining inflectional string matching and parse-based extraction is the most reliable configuration.

	Precision	Recall	F1-score
Parsing-Context ∪ Exact-CS-0Words	**89.18**	93.93	91.50
Parsing-Context ∪ Inflect-CS-0Words	89.00	**95.22**	**92.01**

Table 4.9: PIE extraction performance of the combined output (union) of a string-based and a parser-based system, on the development set. *CS* indicates case-sensitive string matching. The best score for each metric is in **bold**.

4.5 Conclusions

We have presented an in-depth study on the automatic extraction of potentially idiomatic expressions based on dictionaries. The purpose of automatic dictionary-based extraction is, on the one hand, to function as a pre-extraction step in the building of a large idiom-annotated corpus. On the other hand, it can function as part of an idiom extraction system when combined with a disambiguation component. In both cases, the ultimate goal is to improve the processing of idiomatic expressions within NLP. Given the relative novelty of this task, we do not just explore how to develop PIE extraction methods, but also cover more fundamental questions on the coverage of idiom dictionaries and the evaluation of PIE extraction.

Based on the work presented in this chapter, we can now answer the research questions posed in Section 4.1. Our first question concerns the coverage of idiom dictionaries. Based on the comparison of dictionaries to each other, we estimate that the overlap between them is limited, varying from 20% to 55%, which indicates a large divergence between the dictionaries. This can be explained by the fact that idioms vary widely by register, genre, language variety, and time period. In our case, it is also likely that the divergence is caused partly by the gap between crowdsourced dictionaries on the one hand, and a dictionary compiled by professional lexicographers on the other. Given these factors, we conclude that a single dictionary cannot provide even reasonably complete coverage of English idioms, but that by combining dictionaries from various sources, matters can be significantly improved.

In Section 4.3, we experiment with the exhaustive annotation of PIEs in a corpus of documents from the BNC. Using a set of 591 PIE types, much larger and more varied than in existing resources, we show that it is feasible to establish a working definition of PIE that allows for a large amount of variation, while still being useful for reliable annotation. This resulted in high inter-annotator agreement, ranging from 0.74 to 0.91 Fleiss' Kappa. The final corpus of PIEs with sense annotations is publicly available and consists of 2,239 PIE candidates, of which 1,050 actual PIEs instances, and contains 278 different PIE types. This corpus is the answer to our second question, about how we can evaluate PIE extraction; it shows that exhaustive and reliable annotation of PIEs at sufficiently large scale is possible for an acceptable cost in time and effort.

Finally, several methods for the automatic extraction of PIE instances were developed and evaluated on the annotated PIE corpus. We tested methods of differing complexity, from simple string match to dependency parse-based extraction. Comparison of these methods revealed that the more computationally complex method, parser-based extraction, works best. Parser-based extraction is especially effective in capturing a larger amount of variation, but is less precise than string-based methods, mostly because of parser error. The best overall setting of this method, which parses idioms within context, yielded an F1-score of 88.54% on the development set. Parser error can be partly compensated for by combining the parse-based method and the inflectional string match method, which yields an F1-score of 92.01%. This aligns well with the findings by Baldwin (2005), who found that combining simpler and more complex methods improves over just using a simple method case for extracting verb-particle constructions. It also provides the answer to our final research question: yes, a single extraction method can cover a wide variety of idiom types with high accuracy, but a combination of methods works even better. Furthermore, this level of performance means that the tool can be used in corpus building by reducing the amount of manual extraction effort involved, which we do in Chapter 5.

As for future applications, we note that, although used here in the context of PIE extraction, our methods are equally applicable to other phrase extraction tasks, for example the extraction of light-verb constructions, metaphoric constructions, collocations, or any other type of multiword expression (cf. Baldwin, 2005; Iñurrieta et al., 2016; Savary and Cordeiro, 2017). Similarly, our method can be conceived as a blueprint and extended to languages other than English. For this to be possible, for any given new language one would need a list of target expressions and, in the case of the parser-based method, a reliable syntactic parser. If this is not the case, the inflectional matching method can be used, which requires only a morphological analyser and generator. Obviously, for languages that are morphologically richer than English, one would need to develop strategies aimed at controlling non-exact matches, so as to enhance recall without sacrificing precision. Previous work on Italian, for example, has shown the feasibility of achieving such balance through controlled pattern matching (Nissim and Zaninello, 2013). Languages that are typologically very different from English would obviously require a dedicated approach for the matching of PIEs in corpora, but the overall principles of extraction, using language-specific tools, could stay the same.

CHAPTER 5

[*]Crowdsourcing a Large Idiom Corpus

Abstract|In the previous chapter, we have shown the potential of automatic pre-extraction methods as a highly accurate tool for corpus building. As such, in this chapter, we focus on using these methods together with a crowdsourced annotation approach to build a large corpus of sense-annotated PIEs. Given the limited size of existing corpora, we aim to enable progress in idiom disambiguation and evaluation by creating a corpus of larger scale. We find that, using a fixed idiom list, automatic pre-extraction, and a strictly controlled crowdsourced annotation procedure, it is feasible to build a high-quality corpus containing a total of 56,622 instances. The most crucial parts of crowdsourcing were the selection of crowdworkers, clear and comprehensive instructions, and an interface that breaks down the task in small, manageable steps. Analysis of the resulting corpus revealed strong effects of both genre and form variation on idiom distribution, providing new evidence for existing theories on what influences idiom usage.

[*] Chapter published as: Haagsma, H., Nissim, M., and Bos, J. (2020). MAGPIE: A Large Corpus of Potentially Idiomatic Expressions. In *Proceedings of LREC 2020.* (to appear)

5.1 Introduction

Idiomatic expressions are an established research topic within NLP, but progress has been hampered by a lack of large annotated corpora. Existing corpora cover less than 5,000 instances of less than 100 different idiom types, which means they do not provide a good training and testing ground for methods aimed at capturing the phenomenon of idiom as a whole. Idioms are a very large, constantly changing, fuzzy-boundaried category of expressions, which cannot be completely captured in a corpus by its very nature. However, the bigger and more varied the corpus, both in number of different idiom types and the number of idiom instances, the more likely it is that conclusions drawn from this corpus are valid for 'idioms' as a whole.

Larger corpora have benefits from both the descriptive and computational linguistic perspectives. For the first, it allows for better evaluation of assumptions about idiomatic expressions regarding their distribution, frequency, variation, and more. For natural language processing purposes, it allows for more reliable, more fine-grained, and more representative evaluation of idiom disambiguation tools, and for better training of such tools, given the possibility of using more data-greedy machine learning tools.

In this chapter, we aim to build a corpus that satisfies the criteria mentioned earlier. This corpus should be significantly larger than existing ones. If we take that to mean at least 10 times bigger, it should contain over 1,000 different idiom types and 50,000 instances. Due to its size, we do not rely on the tried and tested method of using expert annotators to label the potentially idiomatic expressions. Instead, we make use of the automatic tool developed in Chapter 4 for pre-extraction and use crowdsourcing to get the annotations from non-experts. Of course, crowdsourcing is an established and reliable method for large-scale annotation, but it has not been applied to idioms specifically, likely because of the significant difficulty of the task and the resulting difficulty of instructing crowdworkers to perform the task reliably. However, there is closely related work by Kato et al. (2018), who use crowdsourcing for a similar annotation task for verbal MWEs, which includes some idioms. Although they do not provide details on inter-annotator agreement, they generally get good results, with all 3 annotators agreeing in approximately 67% of cases.

Here, we will describe the selection of idiom types and data to annotate, the crowdsourcing setup, procedure and difficulties, and a detailed analysis of the resulting corpus. Ultimately, based on this work, our aim is to answer the following research questions:

1. Is crowdsourcing a suitable method for large-scale, high-quality annotation of a large variety of potentially idiomatic expressions?

2. To what extent can existing assumptions and theories about the distribution, variability, and frequency of idioms be verified using a corpus of this size?

3. How do genre and form variation interact with sense distributions and annotation difficulty?

5.2 Idiom and Corpus Selection

Overall, the procedure for building the corpus is as follows: we select a set of idioms from a dictionary, and use the PIE extraction system to extract all forms of this idiom type from a base corpus. The resulting PIEs are annotated using crowdsourcing, and the resulting annotations are aggregated and converted into a corpus of sense-annotated PIEs.

We use three dictionaries, the Oxford Dictionary of English Idioms, UsingEnglish, and Wiktionary (see Section 4.2 for details). Taking the intersection of all three yields the highest precision, but only a limited number of 591 expressions. Therefore, we compromise a little bit on precision to get a higher number of expressions. We use all expressions which occur in ODEI and in either UE or Wiktionary, i.e. the following set: $(Wiktionary \cap ODEI) \cup (UE \cap ODEI)$. This yields a list of 2,007 idiom types. Because these contain some undesirable instances (e.g. *to go*), we manually refine the list. Filtering these instances out leaves 1,967 types, based on the following criteria:

- It should not yield an overwhelming number of false extractions. These are cases with little lexico-semantic content like *to go, at it,* and *have a go.*
- It should be an idiom according to the definition of idiom specified in Section 2.2.

We do not take into account our intuitions about potential sense distributions, i.e. we do not exclude types which we expect to only be used literally or idiomatically. We avoid this because it is difficult to reliably estimate sense distributions, and because this is something that will become clear as a result of the annotation. The same thing goes for idiom types for which we do not see a clear distinction between literal and idiomatic senses. If this is indeed the case, it will be reflected in (a lack of) inter-annotator agreement.

As for the base corpus for our idiom annotation, we select the British National Corpus (BNC Consortium, 2007). This corpus has many benefits, such as its size (large enough to get sufficient idiom instances, while easy to work with), its variety (many different genres are included, which hopefully leads to more varied idiom types and forms), and its standardised format (no noisy data, even in the transcribed spoken part). The main disadvantage is that it is not openly available, meaning the corpus can only be released as offset annotations. However, the corpus is available easily and for free on the web[2], which means that anyone can get their own copy without significant cost or effort. Other disadvantages

[2] `ota.ox.ac.uk/desc/2554`

are its limited geographical scope (i.e. it only has the British variant of English, meaning some idiom variants might not occur), and its limited time period (1960–1993), which makes it somewhat dated. However, the fact that it only contains British English also means that it aligns well with the main source of idioms, ODEI, which is focused on British idioms.

We use the pre-extraction system described in Chapter 4 to select candidate phrases for annotation. To achieve both high-precision and high-recall extraction, the combination of the regular parser-based system and the case-sensitive no-intervening-words inflection-based system is used, which achieved an F1-score on PIE extraction of almost 91% (cf. Table 4.9). We value a balance between precision and recall over maximising recall, since a lower precision would lead to annotators being overloaded with non-PIEs, complicating the annotation task. Applying this combined system to the whole of the BNC yields just over 200,000 instances, as shown in Table 5.1.

Some heuristics are applied to increase the quality of this set of instances and to whittle down its size. Instead of using the original idiom list, we use the manually filtered idiom list, which greatly decreases the number of instances. By excluding 40 idiom types, the number of annotation candidates drops by over 80,000. This means that the excluded types were highly frequent, due to an extreme number of false extractions (e.g. *to go*).

Finally, we consider downsampling highly frequent idiom types. The reason for this is that these types tend to be frequent either because they have a large number of false extractions, or because they occur very frequently, but only in their literal sense. Moreover, we want our corpus to be useful for generalised idiom disambiguation, i.e. idiom disambiguation that works for all idiom types. This means that we prioritise having more idiom types over having a huge number of instances per type, which would only benefit per-type classifiers. Limiting the number of instances per type allows us to annotate more different types with the same annotation budget. We settle on a maximum of 200 instances per type as a compromise between type coverage and overall corpus size, leaving us with 72,713 instances for annotation. The type coverage of this set is excellent, with 1,781 of the 1,967 types occurring at least once (90.54%) and 1,153 types occurring at least 10 times (58.62%).

Downsampling	**Instances**	**+ filtered list**	**# spoken**
n/a	201,602	117,387	11,476
5,000	172,015	117,387	11,476
2,000	152,115	113,489	11,234
1,000	132,965	105,693	10,493
500	109,280	92,720	8,956
200	80,398	72,713	6,992
100	61,273	57,063	5,428
50	43,996	41,803	3,871
20	24,656	23,706	2,155
10	14,675	14,173	1,353

Table 5.1: The number of annotation candidates extracted from the British National Corpus by the combined PIE extraction system, with various constraints. Downsampling specifies the maximum number of instances per idiom type. Filtered list is the exclusion of certain types, as described in Section 5.2. The rightmost column indicates the number of instances part of the spoken-language part of the BNC.

In addition to the BNC data, we use a smaller additional corpus: the Parallel Meaning Bank (PMB; Abzianidze et al., 2017). The PMB is a multilingual corpus with many annotation layers, resulting in a fine-grained meaning representation for each text in the corpus. We include this corpus for two reasons. For one, its multilingual and parallel nature allows for a future extension of the idiom detection and sense annotations to translated texts. Secondly, detecting and annotating idioms in this corpus allows for research on how to represent the meaning of idioms in a deep semantic framework.

Pre-extraction from the PMB is done in a separate step from the BNC extraction, but the annotation procedure is the same. Pre-extraction was done using the same selection of idiom types as earlier, but with one additional filtering step. Because the PMB contains many very short documents, there might not be enough context to disambiguate the PIEs. Therefore, we extract only PIEs with at least 50 characters of context, i.e. the document should be at least 50 characters longer than the span of the PIE. The resulting set of PIEs contains 2,560 instances across 598 types. No downsampling was applied.

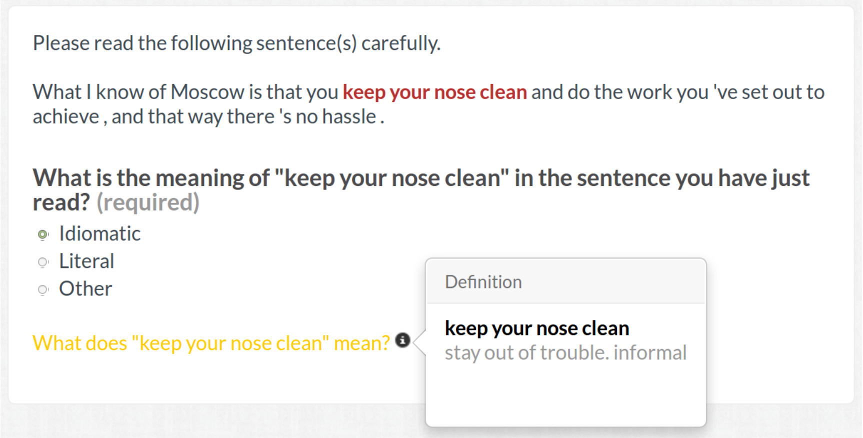
5.3 Annotation Procedure

We make use of the FigureEight[3] crowdsourcing platform for annotation. We see crowdsourcing as the only feasible method for creating a corpus of this size, in contrast to the existing corpora, which were all annotated by researchers and/or their students. Crowdsourcing does pose some challenges, especially for a relatively hard task like this one. Given the set of candidate instances from the pre-extraction system and the FigureEight platform, the challenge is to get high-quality sense annotations at manageable costs of time and money. Therefore, we strive to make the task as easy as possible for crowdworkers.

The basic setup of the task is three-tiered (see Figure 5.1). Given a highlighted PIE instance in context, the annotators are asked to make a three-way decision: whether the instance is used idiomatically, literally, or in a way that does not fit the binary distinction. By only asking about non-binary senses in a subquestion, the initial decision is kept as simple as possible. The dictionary definition of the idiom, extracted from the ODEI, is available on mouseover of the highlighted text below the question. This is only displayed on-demand, in order to prevent information overload.[4] This constitutes the first tier of the annotation.

[3] `www.figure-eight.com`

[4] During one round of pilot annotations, crowdworkers were asked to indicate whether they were unfamiliar with the idiom type in question. This was almost never indicated, but annotation quality increased dramatically after adding in definitions nonetheless.



Please read the following sentence(s) carefully.

What I know of Moscow is that you **keep your nose clean** and do the work you 've set out to achieve , and that way there 's no hassle .

What is the meaning of "keep your nose clean" in the sentence you have just read? (required)

● Idiomatic
○ Literal
○ Other

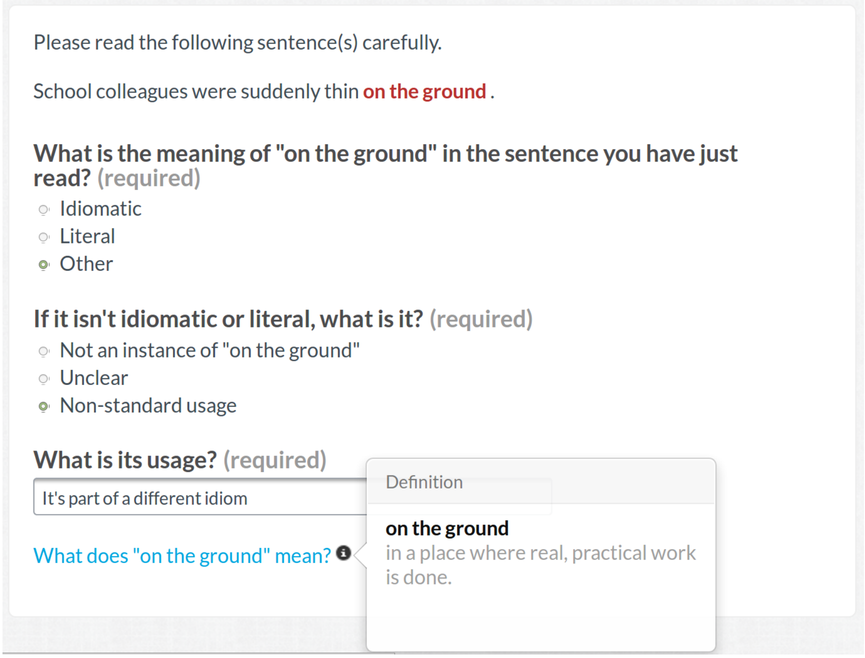
What does "keep your nose clean" mean? ℹ

Definition

keep your nose clean
stay out of trouble. informal

Figure 5.1: Screenshot of the annotation interface presented to crowdworkers, with definition tooltip displayed.

The second tier of annotation is only triggered when the 'Other'-sense is selected in the first tier (see Figure 5.2). In this case, annotators are asked whether the instance is a false extraction ('Not an instance of ...'), whether the given context is insufficient to interpret the instance ('Unclear') or whether it can be interpreted but simply does not fit the binary sense distinction ('Non-standard usage'). Since this latter category is a 'miscellaneous'-category that can contain anything, perhaps even things we have not thought of beforehand, it triggers the third tier. The third tier is a plain text input asking annotators to describe the instance's usage or meaning in the context, i.e. why they selected the non-standard usage option. For example, this category contains PIE instances used meta-linguistically ('The origin of the expression *bite the bullet* is...') and instances occurring as part of different idioms ('We *saw the light* at the end of the tunnel').



Please read the following sentence(s) carefully.

School colleagues were suddenly thin **on the ground** .

What is the meaning of "on the ground" in the sentence you have just read? (required)

○ Idiomatic
○ Literal
● Other

If it isn't idiomatic or literal, what is it? (required)

○ Not an instance of "on the ground"
○ Unclear
● Non-standard usage

What is its usage? (required)

It's part of a different idiom

What does "on the ground" mean?

Definition

on the ground
in a place where real, practical work is done.

Figure 5.2: Screenshot of the annotation interface presented to crowdworkers, with definition tooltip displayed and all subquestions triggered.

5.4 Selection of Crowdworkers

The annotation procedure itself stayed constant throughout the annotation of the corpus, but the way crowdworkers were selected and tested was subject to experimentation and variation. Based on pilot rounds of annotation, our initial setup was as follows.

Annotators are presented with instructions (see Appendix A). The instructions describe the nature of the task, which steps to go through at each annotation question, an overview of the possible answers, examples for each possible answer, and a section highlighting difficult cases (e.g. PIEs as part of proper names). After reading the instructions, annotators are presented with a set of 6 test questions, in the first phase of work (called quiz mode). If they answer at least 70% of these questions correctly, they can start the actual annotation phase (called work mode). Here, annotators are presented with 6 instances at a time, of which one is a test question. During work mode, their accuracy on test questions should remain above 70% for them to be able to continue working. The test questions have gold standard annotations, we take these from the PIE Corpus described in Section 4.3. Additional test questions were added later by taking items with high-agreement crowdsourced annotations and manually checking them for validity and suitability.

Although the use of test questions in general is a useful quality assurance method, we find that it is very much dependent on the selection of questions. The questions should be a good representation of the possible cases and answers in the data, balancing simple and difficult questions. Including too many hard questions excludes quality crowdworkers, while having only easy questions does not expose crowdworkers to the more challenging cases. We do this by enforcing a label distribution[5] among test questions and excluding test questions which are ambiguous. That is, we leave in difficult cases (e.g. idiom as part of a different idiom), but only if they are unambiguously so and clearly covered by the instructions. In addition, we provide reasons for why the test questions have a certain label. This ensures that, if crowdworkers get a question wrong, they get clear feedback on why that is the case and they can learn from this for the remainder of the task. If crowdworkers provide the wrong answer, they can also provide feedback on why they think their initial answer should be correct. Based on this feedback, the set of test questions is updated and problematic questions are filtered after every batch of annotations.

We limit the pool of crowdworkers to largely monolingual English-speaking countries, so we include only the United States, United Kingdom, Canada, Ireland, Australia, and New Zealand. Crowdworkers on

[5] The final label distribution is 25% false extractions, 32% idiomatic, 37% literal, and 6% other, but small variations on this have also been used.

the FigureEight platform are assigned level 1, 2, or 3 status, indicating the quality of their work, level 3 being the highest. We experimented with allowing different levels of crowdworkers to participate in the task and initially settled on using only level 2 and level 3 crowdworkers, since this greatly reduced the number of nonsensical contentions on test questions and the failure rate in quiz mode. Initially, no maximum number of annotations per annotator was set, but to prevent concentration loss and over-reliance on a single annotator, we settled on a limit of 500 per batch. Annotators were paid 4 cents (USD) per annotation (3, 3.5, and 5 cents per annotation were also tried).

Pilot annotations showed that, for many cases, having 3 annotations is sufficient and leads to 100% agreement. Therefore, we collect only 3 annotations per instance. However, we also found that for a minority of challenging cases, agreement is very low, usually $< 50\%$. That is, we see a clear two-way split between straightforward cases (100% agreement) and ambiguous cases ($< 50\%$ agreement). As such, we collect more annotations for these low-agreement cases, until agreement is over 70%, up to a maximum of 7 or 9 total annotations.

This setup yielded good results initially, getting a large number of annotations quickly, with acceptably high accuracy, especially after fine-tuning of the test question selection and guidelines. However, after a certain number of annotation batches, annotation quality degraded quickly and dramatically. We noticed a large influx of US-based crowdworkers achieving 100% accuracy on test questions, annotating the maximum number of instances unrealistically quickly, with implausible answer distributions (e.g. exactly 33.3% literal, 33.3% idiomatic, 33.3% other, where we expect approx. 70% idiomatic, 25% literal, 5% other) and very low agreement on those annotations. Manual inspection of the annotations showed that labels were assigned randomly, free text entries were nonsensical and the annotations were unusable. As such, we stopped annotation and concluded that the quality controls were being circumvented by (a group of) untrustworthy crowdworkers.

Multiple attempts were made to prevent these workers from participating in the task. An additional constraint was placed on annotators by enforcing a specific answer distribution. We experimented with both strict limits, which erroneously excluded too many honest crowdworkers, and lax limits, e.g. *idiomatic* labels between 20% and 80%, *literal* labels between 10% and 60%, and *other* labels between 0% and 40%, which did not sufficiently filter out untrustworthy workers. We also tried not using the same test questions across multiple batches, adding noise to the test questions (machine-readable noise, invisible to humans), excluding USA-based workers, and excluding known untrustworthy workers from working on our tasks. However, none of these approaches yielded lasting results and annotation quality did not increase sufficiently.

Ultimately, the use of an open pool of crowdworkers was abandoned. The initial preference for using an unrestrained set of workers was based on maximising the number of different annotators for the data in order to ensure a diverse set of annotators (and thus, annotations) and to increase annotation throughput. However, this was outweighed by the problems described earlier, and we settled on using a manually selected set of known, reliable, trusted crowdworkers.

These workers were selected by going through the list of workers with high accuracy scores on already-annotated batches, manually inspecting their annotations (both labels and text entries) and classifying them as either trusted, sincere but inaccurate, or untrustworthy crowdworkers. Only crowdworkers classified as ‘trusted’ were allowed to work on subsequent batches. As for untrustworthy crowdworkers, they were excluded from working on this task and their previous annotations were discarded as unreliable. This meant that part of the already-annotated data did not have sufficient annotations. To compensate for this, additional annotations on this data were done by the set of trusted crowdworkers, in order to not let the remaining good annotations on this data go to waste.

Using a set of 54 known and trusted crowdworkers, annotation quality increased drastically and remained stable throughout, as did the complexity of the free text explanations. Although annotation throughput was clearly lower with a small set instead of a large pool of annotators, several thousands of instances could still be annotated per day. Note that trusted crowdworkers still followed the same setup as previously, i.e. they still had to pass quiz mode and maintain a minimum level of accuracy throughout work mode. Pay was increased to 5 cents per annotation, the maximum of 500 annotations per crowdworker was maintained, and up to 7 annotations were gathered for difficult instances. In order to maintain a high standard within the set of trusted crowdworkers, their annotations were compared to the aggregated (majority agreement) label for the annotated instances after each batch. A high overlap indicates agreement with other workers, whereas low overlap reveals outliers. Generally, agreement between individual annotators and the majority label was at least 80%, indicating high reliability of annotations, but avoiding complete uniformity.

With the funds available[6], after a set of experiments aimed at defining and refining the task, and with the combination of funds and rows available, almost 78% of the data was annotated. That is, of the 72,713 extracted instances, 56,622 were annotated. This constitutes a random subset of the full pre-extracted data, so we assume conclusions drawn based on the annotated data hold for the full dataset as well.

5.5 Results

The annotated data is converted into a practically usable corpus by excluding those instances on which the majority of annotators agree that it is a false extraction from the data.[7] Aggregation of annotations, i.e. se-

[6] The annotation funding was provided by FigureEight as part of winning their AI for Everyone Challenge.

[7] The corpus is released as the MAGPIE corpus, and is available at `github.com/hslh/magpie-corpus`.

Listing 5.1: Example of a data point in the corpus for the PIE *larger than life*.

```
{"confidence": 0.8245609268758899,
  "context": [
    "I am undone —— ‘",
    "It was a deep voice of great beauty even when as now , she was over – emphasizing .",
    "Dear Vicky —— larger than life ( too large for little life ... )",
    "She had sat up and was pulling her fingers through the tangled forest .",
    "’ Oh dear God —— I must take your name in vain ."
    ],
  "document_id": "FPH",
  "genre": "W fict prose",
  "id": 316,
  "idiom": "larger than life",
  "judgment_count": 9,
  "label": "i",
  "label_distribution": {
    "?": 0.0,
    "f": 0.0,
    "i": 0.8245609268758899,
    "l": 0.17543907312411014,
    "o": 0.0},
  "non_standard_usage_explanations": [],
  "offsets": [[13, 19], [20, 24], [25, 29]],
  "sentence_no": "1078",
  "split": "training"}
```

lecting the majority label and assigning a confidence score, is done on the annotation platform, so those labels and scores are used.

As for the format of the corpus, each PIE is presented within its context sentence and two additional sentences on either side, if available. Context is extracted directly from the BNC, in tokenised form. Metadata consists of a unique ID for each instance, a document identifier and sentence number, the document genre, the offsets of the component words within the context[8], the label, the confidence score of that label, the distribution of annotations over the labels, the number of annotations, whether it is part of the training, development or test set, and any free text entered as explanation for a non-standard usage. We do not normalise or select these non-standard usage explanations; they are preserved ‘as is’. An example data point is presented in Listing 5.1.

Perhaps the most important factor in the composition of the corpus is the choice for a confidence threshold value. The confidence value is based on the reliability scores of, and agreement between, annotators. The higher the threshold value, the smaller the corpus and the more likely it is to include clear and canonical cases, whereas a lower threshold value results in a bigger corpus containing more difficult and ambiguous cases. In Table 5.2, an overview of corpus size at different threshold values is presented.

Overall, we see that even when we include only instances with perfect annotation agreement, almost 4/5th of the annotated instances are included. This indicates that, although annotators found the task quite difficult, this difficulty originates from just a small portion of cases. Besides this general trend, the influence of specifics of the annotation procedure is significant. Given that there are only five labels, a confidence level of under 0.2 practically never occurs. Above that level, the number of instances declines gradually up to a threshold value of 0.7, after which it declines steeply. This is caused by the fact that more judgements were gathered in addition to the initial 3 while confidence was under 0.7. As such, many instances end up with a confidence level between 0.7 and 0.8, since collection of annotations was stopped once confidence reached that level. For data analysis we will make use of the full corpus, including

[8] Since offsets were generated automatically during pre-extraction, they are not always correct. During the corpus preparation, heuristics are applied to correct the most frequent incorrect and missing offsets and to make them more consistent.

Threshold (≥)	# Instances	% Instances
0.00	56,622	100.00
0.10	56,622	100.00
0.20	56,618	99.99
0.30	56,491	99.77
0.40	56,422	99.65
0.50	55,742	98.45
0.60	53,859	95.12
0.70	52,866	93.37
0.75	48,502	85.66
0.80	45,007	79.49
0.85	44,563	78.70
0.90	44,488	78.57
0.95	44,488	78.57
1.00	44,488	78.57

Table 5.2: An overview of corpus size using different confidence threshold values.

low-confidence annotations.

5.6 Analysis

Given that this is an annotated corpus of PIEs of unprecedented size, in addition to being an excellent resource for training disambiguation models, we are curious about what insight we can gather from it. In this analysis, we look into various aspects: annotation (dis)agreement, distribution of idiom types, sense distributions across types, composition of the ‘other’-category, influence of genre, and influence of variation in PIE form.

5.6.1 Sense Distributions

Given the full corpus of 56,622 annotated instances, our first point of interest is the overall distribution of sense labels. Table 5.3 provides an overview of label distributions for the corpus and its subcorpora. Most noteworthy is that the distribution is far from balanced. The 'other' and 'unclear' labels are rare, as expected, with only 436 'other' instances and 7 'unclear' instances in the whole corpus. The two major labels, 'idiomatic' and 'literal', are not equally common either, with idiomatic instances being around 2.5 times as frequent as literal instances. This is not wholly surprising, given that we include many idiom types which are unlikely to ever be used literally. Moreover, we excluded types which yielded an overwhelming amount of only literal instances and we downsampled the most frequent types, which were likely mostly literal as well. Nevertheless, there is a significant amount of literal instances, making the corpus suitable for PIE disambiguation experiments.

Section	Instances	% Idiom	% Literal	% Other	% Unclear
BNC-written	47,766	71.96	27.29	0.75	0.01
BNC-spoken	6,498	63.50	35.58	0.88	0.05
PMB	2,358	64.12	34.90	0.98	0.00
Total	56,622	70.66	28.55	0.77	0.01

Table 5.3: Basic statistics of the unfiltered annotated corpus and its subcorpora, including label distributions.

In addition to the overall picture, we see a clear difference between the written part of the BNC on the one hand and the other two subcorpora on the other. In the spoken and PMB data, literal instances are more frequent than in the written data, at the cost of idiomatic instances. Moreover, unclear instances occur more in the spoken data. This can be explained by the fact that transcribed spoken data is inherently more noisy than edited, written data.

Having 70% idiomatic instances and 30% literal instances by itself does not make a challenging PIE disambiguation task. If it were the case, for example, that 70% of idiom types are always idiomatic, and 30% are always literal, without any types having both usages, the disambiguation task would be trivial. As such, we need to look at the interaction between the label distributions and idiom types. How many types always have one sense? How many are relatively balanced?

The annotated data contains 1,756 different idiom types, with an average of 32.24 instances for each type. Table 5.4 shows the distribution of instances per type, i.e. how many types have only a single instance, and how many have over a 100 instances. Note that the number of instances per type was limited to a maximum of 200. The table shows that the vast majority of idiom types, 1,430 of 1,756, occur less than 50 times. This fits with the general idea that idioms are individually rare, but frequent as a group (cf. Section 2.3).

Frequency	# Types	Example
1	126	*in apple-pie order*
2–5	372	*greasy spoon*
6–10	264	*hit the right note*
11–20	288	*big hitter*
21–50	380	*true to form*
51–100	148	*for Africa*
100–200	178	*at the crossroads*

Table 5.4: Overview of number of idiom types in each frequency band, with examples.

There is a large amount of variation in the label distributions of the 1,756 types in the data, ranging from all idiomatic to all literal. Figure 5.3 shows the percentage of idiomatic labels each type has, in relation to its frequency. Given the overall predominance of idiomatic senses, it is no surprise that most of the weight in the graph is on the right-hand side, with many types occurring only in their idiomatic sense. In fact, 1,035 out of 1,755 idiom types (58.94%) are purely idiomatic. Of the remaining 721 types, only 17 (0.97%) occur only in their literal sense, meaning that 704 types (40.09%) are ambiguous to some extent.

However, truly ambiguous idiom types, i.e. those with a label distri-

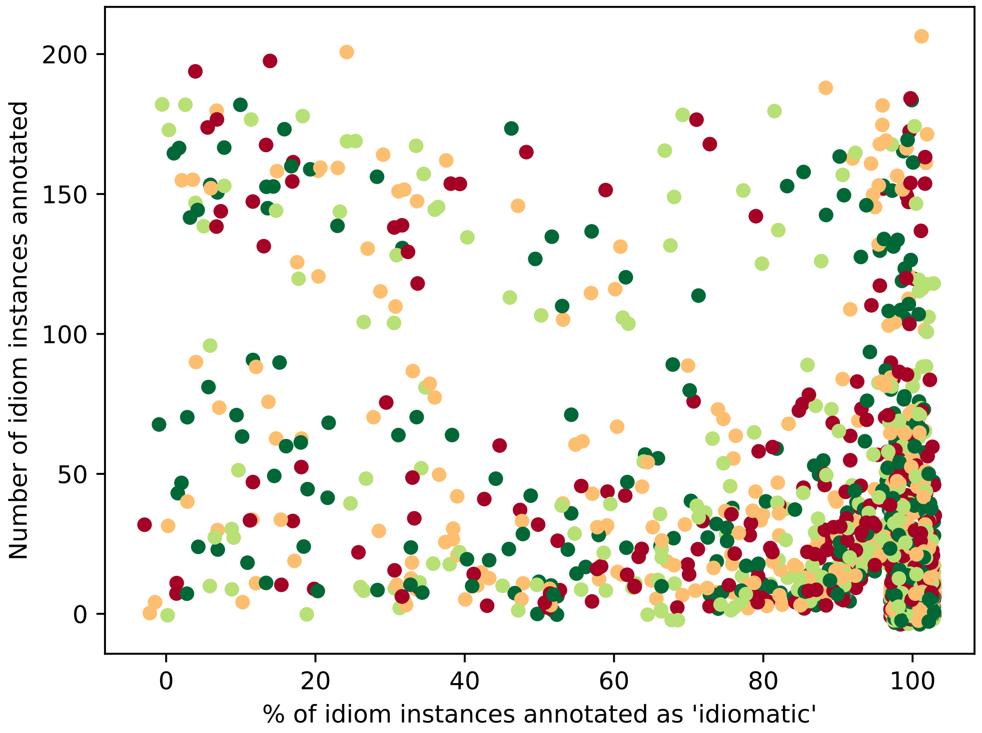


Figure 5.3: Scatterplot of the relation between idiom types’ frequencies and their label distributions. Each data point indicates one idiom type present in the corpus. The colour of the data point carries no information, but enhances visibility. Note that noise has been added to the plot to improve visibility, so some types seem to occur less than 0 times, or are over a 100% idiomatic.

bution close to 50/50, are rare: only 81 types (4.61%) fall in the 40–60% idiomatic range. Looking at high-frequency types (>100 instances) only, it becomes even clearer that, even among ambiguous types, most are clearly leaning towards the literal or idiomatic side: 14 are in the 40–60% range, while 38 are in the 10–30% range. Looking at specific types, the most 'balanced' ones with a significant number of instances (>10) are *cold feet* with 28 instances, 50% idiomatic, and *go all the way* with 168 instances, 50.60% idiomatic.

This is in line with previous findings, such as those by Cook et al. (2008). They select 53 idiom types, explicitly choosing those for which they deem both a literal and an idiomatic interpretation as 'possible'. Still, after this pre-selection they define a 'skewed' subset of the data containing 25 expressions which were too imbalanced for their purposes. Even then, the remaining 28 expressions are generally strongly imbalanced, with only 4 falling in the 40–60% range.

5.6.2 Inter-Annotator Agreement

One of the causes for the high number of idiom-only types is the existence of types like *piping hot* and *turn a blind eye*, which only have a (realistic) idiomatic interpretation. For the same reason, these types tend to have very high confidence scores for their labels; the chances of annotators interpreting them as having a literal meaning are very small. The average confidence score of idiom-only types is 0.9745, while the average confidence score of other types is 0.9244. In line with this, on average, the more balanced a type is in terms of its label distribution, the lower its confidence score is. Simply put, the more ambiguous an idiom type, the harder it is to interpret in context.

Although overall confidence is high at 0.9334, there are large differences between types. For example, *strike a chord* has a high average confidence score of 0.9611, whereas *with a will*, of similar frequency, has a much lower average score of *0.7446*. Of the idiom types occurring more than 10 times in the corpus, the lowest confidence values are for *give head* (0.6286), *have words* (0.6879), *keep your head* (0.7132), and *get a life* (0.7156). In the case of *have words*, for example, there are multiple causes. For one, the perceived rigidity of this expression (any variation like *have the words* loses its idiomaticity immediately), causes annotators to annotate it as a 'false extraction', i.e. a non-PIE, even though the guidelines say otherwise. More importantly, there is a high proportion of annotations with the 'other'-label, due to there being a similar idiomatic expression, *have a word with someone*. In general, instances with this label have much lower confidence, since it is more difficult to recognise than the simple binary distinction between 'idiomatic' and 'literal'. The same reasons apply to the other low-confidence idiom types. For example, *get a life* is also perceived as not allowing any variation, and thus gets marked as a false extraction. This is understandable, since its idiomatic usage only occurs in exclamations ('Get a life, loser!'), and does not extend to in-context usages.

A different view on disagreement comes from a qualitative perspective: considering which annotations co-occur with different labels. That is, given the majority label 'idiomatic', how often are there conflicting minority annotations and if so, which labels get confused most often? Based on this data, the 'idiomatic' label is the least ambiguous and the easiest to annotate. In over 85% of instances, there is complete agreement on this label. In 10% of instances, some annotators chose the 'literal'-label, and in only 1.94% of cases, more than 2 different labels were chosen. By contrast, the 'other'-label has complete agreement in only 8% of instances, with 13% also marked 'idiomatic', 14% marked 'false extraction', and 60% of instances having more than 2 different annotations, clearly indicating the difficulty of identifying these cases for annotators. The main reason for this is that it is easy to overlook, for example, a PIE being part of a different idiom type and mistake it for being idiomatic. Moreover, the 'other'-label covers a very mixed category, combining different things like idioms as part of different idiom, intentional wordplay, and novel usages of existing idioms.

5.6.3 The ‘Other’-Label

For ‘other’-labels, we also collected annotators’ explanations for selecting that label. A closer inspection of these reveals what makes a PIE fall out of the binary split between ‘idiomatic’ and ‘literal’, and which subcategories make up the instances with this label. There are 436 instances with ‘other’ as the majority label. Although these instances have a low average confidence score of 0.5748, manual inspection reveals that the labels are generally correct and reliable.

By applying some simple heuristics, we can get an overview of the frequency of different subgroups in this category. For example, we feel it is safe to assume that any instance with free text explanations containing the word ‘linguistic’ are used meta-linguistically. Based on such heuristics, we find that, out of a total of 436, at least 41 (9.40%) are (part of) titles and names, which we consider non-PIEs and should have been annotated as such (Example 27), at least 332 (76.15%) are PIEs part of bigger and/or different idiom types (Example 28), and at least 21 (4.82%) are meta-linguistic usages (Example 29).

(27) George was recording ‘**Under Lock and Key**’, Steve was doing ‘Eat ’Em And Smile’ in David Lee Roth’s band [..] and the list went on from that. (*under lock and key* - BNC - document C9J - sentence 1504)

(28) GRAHAM Taylor was down **to the bare bones** today when only 14 of his England squad took part in his first Bisham training session [..] (*to the bone* - BNC - document K2D - sentence 288)

(29) When gentle-folk rented horse-drawn transport from Mr Hobson, they were never allowed to select their own horse and carriage, [..] hence the expression ‘**Hobson’s choice**’, meaning no

choice at all (*Hobson's choice* - BNC - document B11 - sentence 697)

Among the remaining 42 instances, we find instances falling into these 3 categories but not captured by heuristics, but also puns and other jokes (Examples 30 and 31), which tend to evoke both meanings simultaneously, as part of a simile (Example 32), and in altogether surrealistic context (Example 33). Note that, technically, Examples 31 and 33 are not PIEs, being a name and having a different part-of-speech respectively, but the question of their interpretation is still interesting.

(30) **IN THE PINK** : Being a snooker star has given Joe a luxury lifestyle, but he still practises six hours a day at his local club (*in the pink* - BNC - document HAE - sentence 1134)

(31) Dear Old Fishfinger, can you recommend some dwarfs for my four foot tank? – Grumpy, Sneezy, Doc, Dopey, Bashful, Happy, Sleepy, Sid Little, **Tired and Emotional**, Colin Moynihan, Dave Dee [..] (*tired and emotional* - BNC - document FBN - sentence 332)

(32) 'After he got back I felt like I was **in a black hole**', groaned a disconsolate Wilkinson after his two hour defeat. (*in the hole* - BNC - document K3H - sentence 1383)

(33) The crying man has no weapon. The baby is the weapon. That's not how things stand **in the black** room, with its groping carbon, its stilled figures. I just know this. In here, the baby is not a weapon. (*in the black* - BNC - document FYV - sentence 1209)

5.6.4 Influence of Genre

A benefit of using the BNC is that it has genre information. For example, document FYV has the genre label `W fict prose`, indicating that it is (part of) a written work of fictional prose. An additional advantage of these labels is that they are not atomic, but structured: every part of the label forms a category of labels. For example, we could select all documents with genres starting with `W` to get all written documents, all `W fict` documents to get all kinds of written fiction, and so on. For an overview of genre encoding in the BNC, see Lee (2000).

We are interested in two main things related to genre: the distribution of PIE labels across genres and the overall frequency of PIEs and idioms in each genre. At the highest level of genre labels, distinguishing written from spoken language, there are only small differences. PIEs and idioms are almost equally frequent, while written PIEs are somewhat more likely to be idiomatic (72%) than those in spoken language (63%).

At a more fine-grained level, there are 34 different genres, displayed in Table 5.5. Here, we see much larger differences appear. For example, in transcribed parliamentary language PIEs are idiomatic 87% of the time, whereas only 38% of PIEs in the also transcribed 'live' demonstration genre are idiomatic. Generally, we find technical and instructional language to have the lowest percentage of idiomatic PIEs, whereas the highest percentages are found with speech and writing which is persuasive in nature, such as political discussions and debates. The likely underlying reason is that the genres with more literal PIEs talk about concrete, physical things (as in a demonstration or instruction), whereas the genres with more idiomatic PIEs focus on rhetoric and abstract ideas, leading to the invocation of PIEs in their non-literal sense. The high frequency of idioms in persuasive and rhetorical language corroborates the statements by McCarthy (1998) that idioms are used for commenting on the world, rather than describing it, and Minugh (2008), who finds that idioms are used most often by those with some authority, especially when conveying 'received wisdom'.

However, this genre distinction is still quite crude. For example, all academic texts are grouped together under `W ac`, with no distinction between different fields of science. When this genre is split out further, a new picture emerges (Table 5.6). A clear split emerges between the texts

Genre	% Idiom.	PIEs	K Tokens	fPIE	fIdiom
W hansard	87.24	478	1,167	409	357
S parliament	87.23	47	95	491	429
S pub	84.40	109	281	387	326
W commerce	81.75	1,452	3,789	383	313
W newsp	80.70	7,711	9,288	830	670
W religion	78.31	627	1,125	557	436
S tutorial	77.42	62	139	446	345
S meeting	76.60	748	1,342	557	427
W news	75.86	1,077	1,202	896	680
W ac	74.44	3,681	16,022	230	171
S brdcast	73.96	699	1,033	676	500
W nonAc	73.60	5,860	16,152	363	267
W pop	73.59	5,797	7,356	788	580
W institut	73.47	98	550	178	131
S speech	73.22	463	639	724	530
W biography	72.52	2,107	3,523	598	434
S unclassified	69.89	279	406	686	479
S lect	69.61	181	292	619	431
W email	68.86	273	210	1,299	894
S consult	66.67	66	132	499	333
W misc	66.03	4,416	9,190	480	317
W advert	65.52	261	549	475	311
W fict	65.47	13,532	15,928	850	556
W admin	65.38	26	221	117	77
W letters	64.71	51	119	428	277
S courtroom	64.29	28	126	221	142
S sermon	61.73	81	79	1,014	626
W essay	59.46	74	202	366	218
S conv	56.30	2,849	3,979	716	403
S interview	55.42	590	921	640	355
S sportslive	53.57	56	32	1,739	932
S classroom	52.05	219	413	530	276
S demonstratn	38.10	21	30	686	261
W instructional	37.14	245	439	558	207

Table 5.5: Distribution of (idiomatic) PIEs across genres. *K Tokens* is the number of tokens in each genre, in thousands. *fPIE* is the frequency of PIEs per million tokens. *fIdiom* is the frequency of idiomatic PIEs per million tokens.

Genre	% Idiom.	PIEs	K Tokens	fPIE	fIdiom
W ac: polit law edu	79.00	1,081	4,671	231	183
W ac: humanities arts	78.78	1,046	3,338	313	247
W ac: soc science	75.78	1,160	4,765	243	184
W ac: medicine	59.18	147	1,433	103	61
W ac: tech engin	43.68	87	690	126	55
W ac: nat science	36.25	160	1,122	142	52

Table 5.6: Distribution of (idiomatic) PIEs across genres. *K Tokens* is the number of tokens in each genre, in thousands. *fPIE* is the frequency of PIEs per million tokens. *fIdiom* is the frequency of idiomatic PIEs per million tokens.

on exact sciences on the one hand, which rarely use idiomatic PIEs, and humanities, law and social sciences on the other hand, which use idioms much more frequently, with medicine being somewhere in between. Perhaps, this difference too is caused by the focus on concrete, physical things in one area and a focus on rhetoric and abstract ideas in another.

Our findings are complementary to those by Simpson and Mendis (2003), who also look at the frequency of idioms in different academic disciplines. Their initial expectation is that 'soft sciences' use more idioms than 'hard sciences', but they find no significant difference in their data. However, they use transcribed spoken data of different academic contexts, whereas we use written papers. As such, a possible explanation for their incorrect expectation is that it was influenced by their impressions from written language from the different disciplines.

The balance between idiomatic and other usages of PIEs does not mean much if the overall number of PIEs and/or tokens in the genre is very small. Therefore we also look at the frequency of PIEs relative to the number of tokens. The average frequency across the corpus is 560 PIEs per million tokens, of which 70.95% are idiomatic, so 397 idiomatic PIEs per million tokens.[9] These frequencies differ greatly between genres, with `W ac` averaging only 230 PIEs per million, and `W newsp` averaging 830 PIEs per million. More extreme values are found, but only with small genres such as `W admin` and `S sermon`.

If we exclude such genres, looking only at those having at least a million tokens, we get the overview in Table 5.7. Note that this uses more fine-grained genres than Table 5.5, so we can look at academic writing in more detail, for example. Now, `W news script`, which contains newsreaders' autocue data, contains the most PIEs per million tokens, 896, almost nine times as frequent as its opposite, `W ac: medicine`, academic texts on medicine, which has only 103 PIEs per million tokens. For idiomatically used PIEs, the proportions are similar.

Finding `W news script` and `W newsp other` with the highest idiom frequencies aligns nicely with previous work on idiom frequencies. For example, Moon (1998) notes that the frequency of idioms in spoken language has been overestimated relative to written language. She suggest this may be caused by the high frequency of idioms in scripted speech, such as in fiction, film, and television, a category which also covers `W news script`. As for `W newsp other` (and `W newsp brdsht`, which has the third-highest fIdiom), it has been noted that journalistic writing is a particularly rich source of idioms (Moon, 1998; Fazly et al., 2009; Grégoire, 2009).

More generally, we see that academic texts (`W ac`) have the lowest frequency of PIEs, followed by non-academic non-fiction (`W nonAc`), i.e. texts whose main purpose is instruction, information, and education. PIEs are most frequent in news, prose fiction, conversations, and popular magazines (`pop lore`), i.e. texts whose main purpose is entertainment. However, spoken conversations (`S conv`) do not fit this category neatly, even if they have a similar PIE frequency. We have no clear explanation for its high PIE frequency, but we do note that it stands out of

[9] Note that only 75% of the PIE candidates were annotated, so these numbers cannot be directly compared to statistics from other datasets.

Genre	% Idiom.	PIEs	K Tokens	fPIE	fIdiom
W news script	75.86	1,077	1,202	896	680
W newsp other	80.94	4,712	5,559	848	686
W fict prose	65.40	13,250	15,662	846	553
W pop lore	73.59	5,797	7,356	788	580
S conv	56.30	2,849	3,979	716	403
W newsp brdsht	79.54	2,116	3,010	703	559
W biography	72.52	2,107	3,523	598	434
S meeting	76.60	748	1,342	557	427
W religion	78.31	627	1,125	557	436
W nonAc: tech engin	87.07	642	1,212	530	461
W misc	66.03	4,416	9,190	480	317
W hansard	87.24	478	1,167	409	357
W nonAc: nat science	63.89	1,022	2,527	404	258
W nonAc: soc science	73.46	1,458	3,683	396	291
W commerce	81.75	1,452	3,789	383	313
W nonAc: humanities arts	71.65	1,330	3,724	357	256
W ac: humanities arts	78.78	1,046	3,338	313	247
W nonAc: polit law edu	78.18	1,219	4,502	271	212
W ac: soc science	75.78	1,160	4,765	243	184
W ac: polit law edu	79.00	1,081	4,671	231	183
W ac: nat science	36.25	160	1,122	142	52
W ac: medicine	59.18	147	1,433	103	61

Table 5.7: Distribution of (idiomatic) PIEs across genres. *K Tokens* is the number of tokens in each genre, in thousands. *fPIE* is the frequency of PIEs per million tokens. *fIdiom* is the frequency of idiomatically used PIEs per million tokens.

the group of news, prose, and magazines if we look at the frequency of idiomatically used PIEs rather than PIEs overall.

5.6.5 Form Variation

As a final area of interest in this corpus we look into PIE form variation. That is, the deviation of PIEs from the idiom's canonical or dictionary form, as in Example 34, and its relation to inter-annotator agreement, genre, and label distribution.

(34) However, with attendances falling and United facing a potentially damaging FA Cup third - round draw [..] **Edwards's hand may yet be forced**. (*force someone's hand* - BNC - document AA7 - sentence 206)

First, however, we use heuristics to identify which PIEs exhibit variation and try to classify this variation into categories. Table 5.8 presents an overview of variation categories and their frequency.

PIEs appearing in their dictionary form are classed as Identical. Those exhibiting only orthographic differences are in the classes Case for case differences and Dashes for PIEs written as one, e.g. 'never-say-die'. Inflection covers PIEs showing only inflectional variation, as in Example 35. The Insertion category covers PIEs which have an additional word inserted at any place in the PIE. This includes both major insertions, such as in Example 34, or minor insertions like the determiner in Example 36. Deletion contains the inverse, although Deletion is limited to determiner deletions only, as in Example 37. Although determiner insertion and deletion are covered by these two categories, determiner variation falls under the two Determiner categories. This contains PIEs where the determiner is replaced by another determiner (narrow, Example 38), or replaced by another word, such as a quantifier (broad, Example 39). The Placeholder category cover PIEs where a placeholder word like 'someone' was replaced by a filler word (Example 40). Finally, the Combined category contains PIEs which exhibit a combination of multiple types of variation. This includes the largest deviations from the dictionary form, as in Examples 41 and 42. The remaining 2.50% are cases which are not captured by heuristics or do not fit into any of the mentioned categories.

Variation Category	Count (abs)	Count (%)
Identical	26,375	46.58
Case	2,049	3.62
Dashes	3,580	6.32
Inflection	9,305	16.43
Insertion	2,287	4.04
Deletion	777	1.37
Determiner (narrow)	592	1.05
Determiner (broad)	971	1.71
Placeholder	828	1.46
Combined	8,442	14.92
Unclassified	1,416	2.50

Table 5.8: Overview of different types of variation exhibited by PIEs and their frequency in the corpus.

(35) On second thoughts, it might be better to say that it **goes without saying** that linguistic communication is a matter of conveying ideas or thoughts. (*go without saying* - BNC - document CK1 - sentence 782)

(36) It was a little winter scene – the Thames frozen from bank to bank, people skating, and children playing round a bonfire **on the ice**. (*on ice* - BNC - document EDJ - sentence 2500)

(37) There was a place [..] – Burkett had pointed it out to him when they were fishing on Derwent Water – a sheer cliff coming suddenly **out of woods** and fronting the valley. (*out of the woods* - BNC - document FP1 - sentence 1456)

(38) We could go downstairs to the sixty eight foot level was er er fog

horn and lighting system for **in the fog**. (*in a fog* - BNC - document HEE - sentence 18)

(39) There is no man in a red jacket standing next to a yellow kayak **on some rocks** by the ocean. (*on the rocks* - PMB - document 61/0053 - sentence 1)

(40) Da da da da da that 's love and I can't believe it's true da da you better stop before you go and **break my heart**. (*break someone's heart* - BNC - document KPE - sentence 73)

(41) According to Kamichika, Japan **has eyes only for** its technology while it continues to ignore the increasingly urgent problems posed by unbridled economic growth [..] (*have an eye for* - BNC - document EBT - sentence 251)

(42) High above, in abrupt contrast, sit the comparatively recent buildings of the town, their roofs and turrets **catching the early morning sun**. (*catch the sun* - BNC - document ARB - sentence 2051)

The data shows that over half of PIEs show no variation at all (Identical) or negligible variation (Case, Dashes). The main categories of actual variation are inflectional variation, insertion, and combined variation. Since most idiom types only allow a limited amount of variation without losing the possibility of an idiomatic interpretation, we expect the type of variation to strongly influence the proportion of idiomatic PIEs. Similarly, we are interested in seeing whether the amount of variation influences human interpretability, e.g. whether certain types of variation make PIEs harder to interpret, and thus lower annotators' agreement. Statistics on these two factors are presented in Table 5.9.

As for confidence scores, there are only small differences between the different types of variation. Overall, confidence is high, with Identical, Dashes, Inflection, and Placeholder categories showing somewhat higher confidence, and the Unclassified a somewhat lower confidence.

Variation Category	Count (%)	Idiomatic (%)	Confidence
Identical	46.58	80.08	0.95
Case	3.62	87.70	0.91
Dashes	6.32	91.40	0.95
Inflection	16.43	77.84	0.94
Insertion	4.04	33.97	0.90
Deletion	1.37	30.50	0.90
Determiner (narrow)	1.05	17.74	0.90
Determiner (broad)	1.71	16.07	0.90
Placeholder	1.46	67.03	0.94
Combined	14.91	45.70	0.89
Unclassified	2.50	62.92	0.87

Table 5.9: Overview of different types of variation exhibited by PIEs and their label distribution and label confidence scores. Idiomatic (%) indicates the percentage of PIEs labelled as idiomatic.

The latter is unsurprising, since this grab bag category contains mostly difficult and singular cases, such as PIEs which are part of different expressions (*hands down* vs. *hand something down*), or which should have been marked false extractions ('low-lying' for *lie low*). On the other end of the spectrum, it makes sense that PIEs identical to the dictionary form are easier to classify, and the same goes for those joined with dashes, which we take to indicate a high degree of fixedness. Inflection is similar to Identical; a change in inflection only when compared to the dictionary form is unlikely to throw annotators off. We would expect the Case category to be equally easy, but this is made more difficult by the fact that it contains relatively many headlines, which are hard to interpret.

The label distribution, represented by the percentage of idiomatic labels, shows much more extreme variation. Broadly speaking, the results are in line with expectation: more variation leads to fewer idiomatic senses. The 'Identical' category provides a baseline level: in its dictionary form, a PIE is 80% likely to have an idiomatic interpretation. This is topped only by 'Dashes' and 'Case'. Since dashes indicate a very high level of fixedness for the PIE, it is essentially written as a single word, it is unsurprising that it is also most likely to have an idiomatic interpretation. For the ‘Case’ category there is no clear explanation. The only other category close to ‘Identical’ is ‘Inflection’, at 77.84%. This indicates that most PIEs retain their idiomatic interpretation under inflectional variance, e.g. ‘fell on deaf ears’ for *fall on deaf ears*.

When subject to more variation, idiomatic interpretations are a lot less common, down to a low of 16.07% for broad determiner variation. Clearly, determiners are crucial in enabling PIEs’ idiomatic interpretations, given that the three categories with lowest idiomaticity are the two determiner variation categories and deletion, which covers determiner deletion. Insertion also reduces an expression’s idiomaticity strongly, likely because only a limited set of expressions allows modification of its component words (through insertion), namely those with a more transparent semantic nature.

Finally, we look at the relation between genre and idiom form variation: do certain genres allow more creativity in the use of idiomatic expressions than others? Since we already know that idiomaticity is strongly related to the amount of variation, we only look at PIEs with the label ‘idiomatic’. If we did not do this, the genres with the highest percentage of literal PIEs would also automatically show the highest variation, not telling us anything about idiom variability. Table 5.10 shows the amount of variation per genre in 3 classes: none (identical), minor (case, dashes, inflection), and major (all other variation). Overall, differences are small, but two things stand out. The genre with most variety is prose fiction writing, which is also likely the most creatively free genre in the list, i.e. there seems to be a relation between idiom variation and creativity in writing. On the other side of the spectrum, news writing stands out for its lack of idiom variation – possibly a reflection of its need to balance entertaining and informational writing.

Genre	Idiom Count	None	Minor	Major
W news script	817	61.44	28.15	10.40
W commerce	1,187	53.75	34.20	12.05
W newsp tabloid	726	48.90	37.60	13.50
W newsp brdsht	1,683	51.69	34.46	13.84
W pop lore	4,266	52.95	32.30	14.74
W nonAc: soc science	1,071	53.50	31.65	14.85
W newsp other	3,814	50.55	34.58	14.87
W misc	2,916	51.99	32.82	15.19
S meeting	573	66.14	18.50	15.36
W nonAc: humanities arts	953	47.64	36.20	16.16
W nonAc: polit law edu	953	51.94	31.48	16.58
W ac: soc science	879	53.24	30.15	16.61
W biography	1,528	52.36	30.69	16.95
W ac: polit law edu	854	52.69	30.33	16.98
S conv	1,604	63.09	19.83	17.08
W ac: humanities arts	824	51.33	31.31	17.35
W nonAc: nat science	653	49.16	33.23	17.61
W nonAc: tech engin	559	49.91	32.20	17.89
W fict prose	8,665	46.34	31.67	22.00

Table 5.10: Overview of amount of variation in idioms across genres. Variation types are binned into three categories: ‘None’ (identical), ‘Minor’ (case, dashes, inflection), and ‘Major’ (all other variation). Only genres with at least 500 idiomatic PIEs are included.

5.7 Conclusions

In this chapter, we have built a large corpus of sense-annotated PIEs using a crowdsourced annotation approach. Given the limited size of previously existing corpora, our aim was to create a corpus of significantly larger scale to enable progress in idiom disambiguation and evaluation. Based on a high-accuracy list of idioms created by combining idiom dictionaries, and by using a strictly controlled crowdsourced annotation procedure, we created a high-quality corpus containing a total of 56,622 PIE instances.

Based on the lessons learned during the creation of the corpus and the findings resulting from analysing its contents, we can now attempt to answer the research questions posed in Section 5.1. First, we consider whether crowdsourcing is a suitable method for large-scale, high-quality annotation of a large variety of potentially idiomatic expressions. The answer to this is a qualified yes: crowdsourcing is suitable, but the procedure has to be set up carefully to yield reliable results. We found that the most important characteristics were the procedure for selecting crowdworkers, the writing of instructions which are both clear and comprehensive, and the development of an interface that breaks down the task in small, manageable steps. The fact that, taking care of these requirements, yielded a corpus which is an order of magnitude bigger than previous ones, both in number of types and instances, with a high-level of inter-annotator agreement, confirms the potential of crowdsourcing for complex linguistic tasks.

As for the second question, which covers the relation between a corpus of this size and its potential for insight in idioms' behaviour and distribution, the answer is varied. On the one hand, the fact that the BNC is the base corpus means it has rich genre information, which provides insight into the matters of idiom usage in spoken vs. written language, and differences between academic disciplines. On the other hand, what we have done here in terms of analysis is limited to a relatively high-level view of idiom distributions and frequencies. As such, the true potential for linguistic insight of the corpus remains for further investigation, which likely includes more in-depth inspection of specific questions and manual encoding of specific idiom characteristics, such as a more fine-grained classification of variation types than we have used here.

Finally, we look at previously under-researched matters regarding idiom: the influence of genre and form variability on the proportion of literal vs. idiomatic PIEs and the difficulty of annotating those PIEs. Overall, we have found that almost all types are strongly skewed towards either idiomatic or literal usage, and that truly balanced types are rare. This implies that a corpus properly representative of idiom as a linguistic phenomenon will necessarily include many skewed types. This contrasts strongly with previous corpus-building efforts, which all explicitly aimed to include mostly balanced idioms.

As for the influence of form variation on sense distributions, we find that, in line with previous findings, more variation leads to a lower proportion of idiomatic usages. For annotation difficulty we find a similar, albeit weaker effect: the closer the PIE is to its dictionary form, the higher the inter-annotator agreement. The strongest effect, however, is that of genre on the proportion of literal and idiomatic PIEs. The percentage of idiomatic PIEs ranges from over 85% in some genres to less than 40% in others, showing a clear pattern: technical, instructional language discussing concrete, physical topics have more literal PIEs, while argumentative, rhetorically rich language touching on more abstract concepts contains relatively more idiomatic PIEs.

PART III

PIE Disambiguation

there's no two ways about it *folksy* no choice about it; no other interpretation of it. (Note the form *there's* rather than *there are.*) ● *You have to go to the doctor whether you like it or not. There's no two ways about it.* ● *This letter means you're in trouble with the tax people. There's no two ways about it.*

CHAPTER 6

*Unsupervised Disambiguation of Potentially Idiomatic Expressions

Abstract|In this chapter, we present various unsupervised approaches for solving the challenge of PIE disambiguation. The main advantage of unsupervised approaches is their ability to generalise to unseen expressions without relying on training data. We make use of the existing cohesion-graph-based approach by Sporleder and Li (2009), optimise it by tuning its parameters, and expand it using a novel information source: literal representations of figurative sense definitions. The main goal of this work is twofold: (1) are literalisations of idiom senses a useful signal for disambiguation?; (2) can an optimised unsupervised method achieve performance competitive with supervised methods? Experimental results show that, while literalisations carry novel information, this is not sufficient to bridge the gap between unsupervised and per-expression supervised approaches. Closer analysis points out the potential for improvement by combining multiple information sources and the need for using multiple datasets and metrics for evaluation.

[*] Chapter adapted from: Haagsma, H., Nissim, M., and Bos, J. (2018). The Other Side of the Coin: Unsupervised Disambiguation of Potentially Idiomatic Expressions by Contrasting Senses. In *Proceedings of the Joint Workshop on Linguistic Annotation, Multiword Expressions and Constructions (LAW-MWE-CxG-2018)*, pages 178–184, Santa Fe, New Mexico, USA. Association for Computational Linguistics

6.1 Introduction

Disambiguating potentially idiomatic expressions (PIEs, for short) is the task of determining the meaning of potentially idiomatic expressions in context. In its most basic form, this consists of distinguishing between the figurative and literal usages of a given expression, as illustrated by Examples 43 and 44, respectively.

(43) Melanie **hit the wall** so familiar to British youth: not successful enough to manage, but too successful for help. (*hit the wall* - British National Corpus - doc. ACP - sent. 1209)

(44) There was still a dark blob, where it might have **hit the wall**. (*hit the wall* - British National Corpus - doc. B2E - sent. 1531)

In this work, we will focus on this basic distinction with the binary version of the task, where PIEs take either a literal or a figurative sense. This is sufficient for almost all uses of potentially idiomatic expressions, but more complicated cases exist, such as meta-linguistic usages, deliberate wordplay invoking both senses simultaneously, and idiomatic expressions with multiple figurative senses.

Distinguishing literal and figurative uses is a crucial step towards being able to automatically interpret the meaning of a text containing idiomatic expressions. This, in turn, is a requirement for high-quality natural language processing that deals with meaning in one way or another. It has been shown that idiomatic expressions pose a challenge for various NLP applications (Sag et al., 2002), including machine translation (Salton et al., 2014a; Isabelle et al., 2017; Fadaee et al., 2018), and sentiment analysis (Williams et al., 2015; Liu et al., 2017; Spasić et al., 2017; Hwang and Hidey, 2019). However, being able to interpret the figurative meaning of idioms mitigates this problem, as has been shown for machine translation (Salton et al., 2014b).

In this chapter, we use a method for unsupervised disambiguation that uses semantic cohesion between the PIE and its context, based on the lexical cohesion approach pioneered by Sporleder and Li (2009). We improve and extend this method, mainly by adding literal representations of idioms' figurative senses, which we call literalisations. Finally, we evaluate the extended and improved methods in a comprehensive evaluation framework to answer the following research questions:

1. Do contexts enriched with literalisations of idioms provide a useful new signal for disambiguation?

2. To what extent can an unsupervised method that optimally uses the available information approach the performance of supervised methods?

6.2 Unsupervised vs. Supervised Methods

PIE disambiguation research consists of three major lines: unsupervised methods, type-general supervised methods, and type-specific supervised methods. The supervised methods for PIE disambiguation use training data to learn context-sense relations for either one PIE at a time or all at once. Unsupervised methods strive to exploit inherent properties of PIEs and their context to determine the right sense-in-context. In addition, there are a few that combine the two, e.g. acquiring training data for a supervised classifier in an unsupervised manner. An overview of these methods is presented in Section 3.4.3. Here, we discuss the considerations to be made for each type of method.

Type-specific supervised methods generally achieve better performance, since they train a separate classifier for each expression. However, they rely on large amounts of training data for each PIE type, so they cannot be easily extended to deal with new, unseen expressions. This crucially hampers their practical applicability. To be practically useful, such methods would need a large amount of training data for each possible PIE type, which is an unrealistic requirement. Type-general supervised methods could potentially be the solution, combining good performance with applicability to unseen types. However, in practice such supervised methods have yielded mostly unsatisfactory results.

Because of such drawbacks, we choose to focus on unsupervised approaches. They do not rely on annotated PIE sense data, and can be expected to work equally well for all types of idiomatic expressions. Of course, unsupervised approaches have their own drawbacks, the main one of which is that the signal that can be extracted from raw data is much weaker than one learned from annotated data. In addition, this signal has to carry information that is general enough to disambiguate any type of PIE. Clearly, it is easier to learn that the word *grass* in the context of *make hay* is an indicator of literal sense than defining a feature that is indicative for both the figurative sense of *face the music* and *small potatoes* simultaneously. The challenge is thus to extract all available signals, and exploit them as effectively as possible.

6.3 Data

In order to provide a comprehensive evaluation dataset, we make use of four[2] sizeable corpora containing sense-annotated PIEs: the VNC-Tokens Dataset (Cook et al., 2008), the IDIX Corpus (Sporleder et al., 2010), the SemEval-2013 Task 5 dataset (Korkontzelos et al., 2013), and the PIE Corpus.[3] An overview of these datasets is presented in Table 6.1. Each corpus has somewhat different benefits and downsides: VNC-Tokens only contains verb-noun combinations (e.g. *hit the road*) and contains some types which we would not consider idioms, but a more general type of MWE (e.g. *catch one's attention*); the IDIX corpus covers various syntactic types and has a large number of instances per PIE type, but is only partly double-annotated; the SemEval dataset is large and varied, but the

[2] We do not include the MAGPIE corpus because the research conducted in this chapter was carried out before it was ready for exploration.

[3] Corpus available at `github.com/hslh/pie-annotation`.

base corpus, ukWaC, is noisy; the PIE Corpus covers a very wide range of PIE types, but has only few instances per type and is also partly single-annotated. Therefore, we combine these four datasets in order to get a dataset that is not skewed by any particular corpus property.

Dataset	Types	Instances	Labels	Base Corpus
VNC-Tokens	53	2,984	3	BNC
IDIX	52	4,022	6	BNC
SemEval	65	4,350	4	ukWaC
PIE Corpus	278	1,050	3	BNC
Combined (dev.)	299	8,235	2	BNC & ukWaC
Combined (test)	146	3,073	2	BNC & ukWaC

Table 6.1: Overview of existing corpora of sense-annotated potentially idiomatic expressions. The source corpus indicates the corpora from which the PIE instances were selected, either the British National Corpus (BNC; Burnard, 2007) or ukWaC (Ferraresi et al., 2008).

6.3.1 Preprocessing

The datasets are combined by converting them into a consistent format. We read in all datasets into a format where each PIE instance has a type (*raise one's eyebrows*), a sense label from the original corpus, a normalised binary sense label (*idiomatic* or *literal*), a 20-sentence tokenised context window, and offsets of the PIE's component words. For PIEs with senses which do not fit the binary split, such as *meta-linguistic*, no binary sense label is defined, and we discard those instances. The same goes for false extractions, i.e. sentences included in the corpus not containing a PIE at all.

For all datasets, the contexts are re-extracted from the base corpus and offsets are added or corrected to match the PIE component words. This allows for the use of large contexts, and assures that PIE component words can be reliably identified in the sentence. For the VNC-Tokens dataset, IDIX, and the PIE Corpus we can re-extract contexts from the BNC easily, using the document IDs and sentence numbers. For VNC-Tokens, we add character offsets for all instances using a combination of automatic pattern matching and manual entry. For three instances, no context could be extracted, since the document ID and sentence number combination did not exist in the BNC. For the IDIX corpus, we do the same as for VNC-Tokens, and in addition discard all double-annotated instances where annotators disagreed. In the PIE Corpus and the SemEval dataset, offsets were already provided. However, for the SemEval data, the context window was provided as is, without any metadata linking it to ukWaC and with many encoding errors. As such, we used (approximate) string matching to find the corresponding documents and sentence numbers in ukWaC and re-extract the contexts. Here too, for three instances contexts were not found in the base corpus, and these were discarded.

6.3.2 Experimental Split

The newly combined dataset is split into a development set and a test set, based on the existing splits of the original datasets. The VNC-Tokens corpus is split into a `development` and `test` set, and a `skewed` set, which contains PIE types with highly skewed sense distributions. The SemEval-2013 corpus is split in two settings, each of which has its own three-way split. In the `known phrases` setting, the `train`, `development`, and `test` sets contain the same set of PIE types. In the `unknown phrases` setting, the `train`, `development`, and `test` sets contain different sets of PIE types, with no overlap. The IDIX corpus is split into `single` annotated and a `double` annotated set. The PIE Corpus is split into `training`, `development`, and `test` sets. We use the test sets of the original corpora to build the combined test set, which thus consists of: `VNC-test`, `IDIX-double`, `SemEval-*-test`, and `PIE-test`. The remaining subsets make up the combined development set. Statistics on the combined dataset are presented in Table 6.1.

6.4 Methods

The disambiguation systems presented here are based on the original lexical cohesion graph classifier developed by Sporleder and Li (2009).[4] Their classifier relies on the notion of lexical cohesion, based on the idea that the words in a PIE will be more cohesive with the words in the context when used in a literal sense, than when used in a figurative sense. For example, *make hay* is likely used literally when the context contains the word *grass*, since *hay* and *grass* are strongly related.

Using this assumption, Sporleder and Li built a fully unsupervised classifier. This classifier works by building cohesion graphs, i.e. graphs of content word tokens in the PIE and its context, where each pair of words is connected by an edge, which in turn is weighted by the semantic similarity between the two words. Then, by comparing the average similarity of the complete graph to the average similarity of only the edges between content words, a classification can be made. If the average similarity of the complete graph is higher, this means the PIE component words add to overall cohesiveness, and thus implying a literal sense for the PIE. Vice versa, if average similarity of the complete graph is lower than within the context, this means the PIE component words make overall cohesiveness worse, meaning the PIE is used in a figurative sense. An example of these graphs is shown in Figure 6.1, for the sentence “That coding exercise was **a piece of cake**”. The full graph on the left has higher similarity than the pruned graph on the right, erroneously producing the classification of *literal*. By contrast, Figure 6.2 shows a different example, for which the same method produces the correct classification of *idiomatic*, since the average similarity of pruned graph is higher than that of the full graph.

[4] The code implementing these systems is available at `github.com/hslh/pie-disambiguation`.

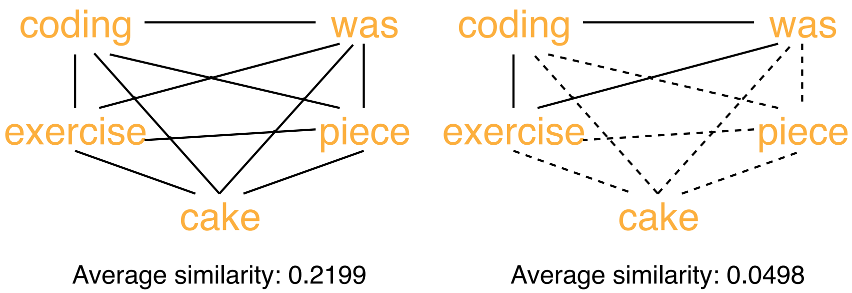


Figure 6.1: Two lexical cohesion graphs for the sentence "That coding exercise was a piece of cake", with their average similarity score. The graph on the left represents the full graph, the graph on the right the pruned graph.

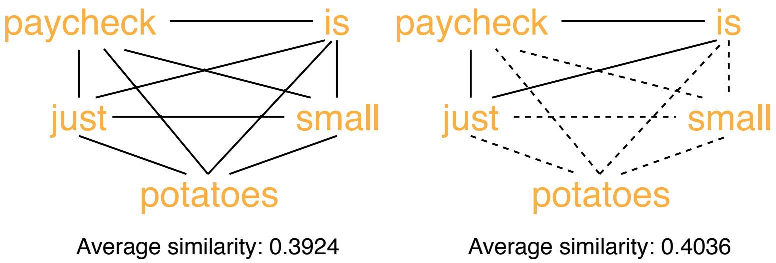


Figure 6.2: Two lexical cohesion graphs for the sentence "This paycheck is just small potatoes", with their average similarity score. The graph on the left represents the full graph, the graph on the right the pruned graph.

We reimplement the original lexical cohesion graph method, with a few modifications. Rather than Normalized Google Distance, we use word similarity based on the GloVe pre-trained word embeddings (Pennington et al., 2014). In addition, Sporleder and Li build the graph using 'content words', which we define as being verbs and nouns, where the part-of-speech of words is determined automatically using the Spacy PoS-tagger.[5] As a context window, we use only the sentence containing

[5] `spacy.io`

the PIE initially.

6.4.1 Optimised Lexical Cohesion Graph

To maximise the cohesion graph's performance, we implement several potential optimisations. We add the possibility to use contexts consisting of various part-of-speech, and of different lengths, either *n* sentences or *n* words in a symmetric window around the PIE. An option to use lemmatisation is also added, to reduce the number of out-of-vocabulary words and get more accurate similarity measures. For the same reason, we experiment with higher-dimensional embeddings.

As for classification, an option is implemented to remove edges connecting 2 PIE component words, since those are not informative of cohesion between PIE and context, but rather add noise to the graph. Secondly, while selecting context words based on part-of-speech is intuitively sensible, it does not always make sense to remove PIE component words from the graph. As such, there is an option to include all PIE component words in the graph. Thirdly, we experiment with adding a classification threshold. Rather than classifying a PIE as idiomatic if average similarity of the whole graph is higher than of the pruned graph, similarity should be at least *n* higher or lower to produce an *idiomatic* label. This serves to compensate for any strong bias in the classifier towards a certain sense.

6.4.2 Idiom Literalisation

The biggest change made to the cohesion graph method is the introduction of idiom literalisations. Idiom literalisations are literal representations of the PIE's figurative sense, similar to dictionary definitions of an idiom's meaning. For example, a possible literalisation of *a piece of cake* is 'a very easy task'. This provides the possibility of building two graphs: one with the original PIE component words, and one with the original PIE replaced with the literalisation of its idiomatic sense. In this way, we can contrast lexical cohesion with a representation of the literal sense to lexical cohesion with a representation of the figurative sense. If the latter is more cohesive, the classifier will label the PIE as idiomatic, and vice versa. Figure 6.3 illustrates this process; the rightmost graph containing the literalisation has higher cohesion than the original graph, leading to the correct classification of *idiomatic*.

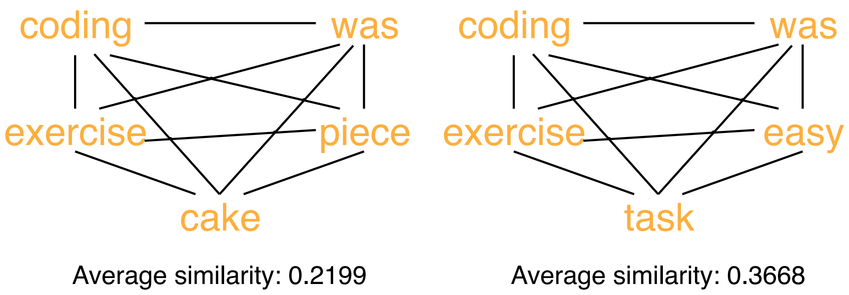


Figure 6.3: Two lexical cohesion graphs for the sentence "That coding exercise was a piece of cake", with their average similarity score. The graph on the left represents the full graph for the original sentence, the graph on the right the full graph for the literalised sentence.

Generally, the change in average similarity will be small, since the context words (which stay the same) greatly outnumber the changed PIE component words. However, since we compare the original and the literalised graph directly, only the direction of the similarity change determines the classification.

We rely on definitions extracted from idiom dictionaries which were manually refined in order to make them more concise. For example, the definition 'Permanently fixed or firmly established; not subject to any amendment or alteration.' for the idiom *etched in stone* is refined to 'permanently fixed or established', in order to represent the figurative meaning of the idiom more concisely.

6.4.3 Evaluation

To put the cohesion graph-based classifier into perspective, we compare its performance to a simple baseline which calculates the most frequent sense in the development set, and classifies all PIEs as such. Performance is judged by three evaluation measures, macro-averaged accuracy, micro-averaged accuracy, and the harmonic mean of the two. Micro-accuracy simply reflects how good the disambiguation system is doing overall; out of all PIE instances, how many were labelled correctly? Macro-accuracy serves to ensure that we do not just optimise on the most frequent types, since some PIE types are much more frequent than others. For example, the most frequent type in our development data (*ring a/the bell*) is more common than the 180 least frequent types together. By using the harmonic mean of the two, we can rely on a single value to indicate balanced performance, both overall and across types. In addition to the harmonic mean, we report micro-accuracy to shed light on overall performance, and macro-accuracy to specifically indicate balanced performance across types.

6.5 Results & Analysis

Our first aim is to explore the potential of the original cohesion graph method, without making use of idiom literalisations. We test the optimisations described in Section 6.4 and report their effect on performance on the combined development set in Table 6.2.

Results show that the fully optimised classifier clearly outperforms the default classifier, with an approximate 6 percentage point increase in all metrics. In terms of individual optimisations, the largest gains are made by restricting the cohesion graph to only use nouns, and to move from 50-dimensional to 300-dimensional word embeddings. Surprisingly, including verbs in the cohesion graph is clearly detrimental to performance, and the same goes for any other parts-of-speech. In line with this, an option to always keep the PIE's component words in the

Setting	Macro	Micro	HM
original	66.40	57.00	61.34
only nouns	68.69	59.15	**63.56**
only verbs	62.70	49.82	55.52
nouns + proper nouns	66.44	57.95	61.91
all except determiners	61.29	54.21	57.53
1-sentence window	68.38	60.55	64.23
2-sentence window	68.85	60.44	**64.37**
3-sentence window	69.26	60.02	64.31
6-word window	66.10	59.80	62.79
lemmatisation	67.56	59.51	63.28
keep component words	55.24	51.66	53.39
no intra-PIE edges	68.91	60.54	**64.45**
100-dimensional	69.16	60.67	64.64
200-dimensional	71.19	61.60	66.05
300-dimensional	71.92	63.38	**67.38**
diff+0.01	71.82	60.84	65.88
diff+0.001	72.67	63.67	67.87
diff+0.0005	72.74	63.78	**67.97**

Table 6.2: Results of the original and optimised cohesion graph classifier on the combined development set. Accuracy scores are micro- and macro-averages over PIE types and the harmonic mean (HM) of the two. Note that each optimisation builds upon the best previous one highlighted in **bold**, e.g. context window length was optimised using a context of only nouns.

graph, regardless of PoS, dramatically lowers performance. Clearly, only nouns provide a useful signal in this setup.

The size of the context window has only little effect on performance, although it is remarkable that windows defined by numbers of sentences are clearly better than word-length windows, even though sentences can be quite variable in length, and thus lead to large variations in graph size. Lemmatisation does not yield any improvements, likely because word embeddings provide good coverage of inflected forms as well, and because inflectional variation is relatively limited in nouns. Removing edges between PIE component words yields a slight but consistent performance increase. This is expected, since these edges do not reflect any specifics of the PIE instance, but instead add noise to the graphs and lexical cohesion scores. Furthermore, high-dimensional word embeddings yield a benefit, indicating the importance of a good similarity measure, and the potential for further improvement in that area. Finally, the classifier tends to under-predict idiomatic senses, which is compensated by setting a very small threshold value.

6.5.1 Added Value of Literalisation

Our first research question asks whether literalisations of figurative senses are a useful source of information for improved disambiguation of PIEs. We test this method on the same dataset as the original cohesion graph method. In addition to a basic implementation, the same optimisations as for the original method are tested and applied, to ensure a fair comparison between the two methods.

Results are displayed in Table 6.3. Note that the optimal settings for the literalisation method differ somewhat from those for the original method, as the optimal context window is five words on each side, instead of two sentences, and the optimal threshold value is $+0.005$, rather than $+0.0005$. We also compare our findings to the most frequent sense baseline performance.

While slightly worse in the default configuration, the graph including literalisations achieves an almost identical score to the original method when fully optimised. It has higher micro-accuracy, but lower macro-accuracy, indicating that it performs better on frequent types than the original classifier. It under-predicts the idiomatic sense quite severely, which is compensated for by a higher threshold value.

Classifier	Macro	Micro	HM
most frequent sense	73.25	57.89	64.67
original	66.40	57.00	61.34
original (optimised)	72.74	63.78	67.97
literalisation	64.56	55.85	59.89
literalisation (optimised)	71.21	64.94	67.93

Table 6.3: Results of the original and literalisation-extended cohesion graph classifiers on the combined development set, with default and optimised settings. Accuracy scores are micro- and macro-averages over PIE types and the harmonic mean (HM) of the two.

Although both classifiers show similar performance scores, the make the same judgement in only 5,737 ($\sim 70\%$) of 8,235 instances in the dataset. Additionally, in only $\sim 49\%$ of cases both classifiers are correct, while in $\sim 15\%$ of cases only the original classifier gets it right, and in $\sim 16\%$ of cases, the literalisation classifier predicts the right label. This indicates that the classifiers have at least some partially complementary performances, as they use different information sources, yielding potential for combination; in $\sim 79\%$ of PIE instances, at least one of the classifiers is right. In a practical setting, such an optimal combination of classifiers is of course not possible, but a significant performance increase could be gained by either combining the features of the two classifiers, or by using average similarity differences as confidence values to pick one classifier over the other for a particular instance.

A closer inspection of the data can reveal whether there are patterns in the classifiers' differing performance. Splitting out performance by subcorpus shows that the original classifier does better on the `VNC-skewed` dataset with a 13% higher score. Conversely, the classifier using literalisations performs about 10 percentage points better on `PIE-train` and `SemEval-allwords-dev`. The difference on the `VNC-skewed` dataset is likely caused by the fact that it contains several frequent types which we would not consider PIEs, such as *catch (someone's) attention* and *have (a) future*. For these items, there is no clear idiomatic sense, so adding literalisations here hurts, rather than helps, performance.

To evaluate whether the syntactic form of the PIE is a significant factor in the difficulty of disambiguating its senses, we also consider performance on different syntactic types of PIE. Examples of such syntactic types are *verb-(determiner)-noun*, e.g. *kick the habit*, or *preposition-adjective-(determiner)-noun*, e.g. *in the fast lane*. The syntactic structure is determined by automatically tagging the PIE types using the Spacy tagger. Looking at the 5 most common frequent syntactic types shows that performance is consistent, and differences between the two classifiers are small. This is unsurprising, given that only nouns are used to build the cohesion graph, and as such, the rest of the PIE's syntactic form is irrelevant.

Since the graph-based classifiers are optimised on the development set, and we report results on that same set, the risk of overfitting exists. Therefore, we evaluate on the unseen combined test set as well. Results in Table 6.4 indicate that our models generalise well. In absolute terms, performance is very similar to that on the development set. Relative to the most frequent sense baseline, the models do better on the test set than on the development set, in particular the model using figurative sense definitions.

	Macro	Micro	HM
most frequent sense	70.22	55.47	61.98
original (optimised)	69.68	65.66	67.61
literalisation (optimised)	69.80	69.18	69.49

Table 6.4: Results of the optimised original and literalisation-extended classifiers on the combined test set, compared to the most frequent sense baseline. Accuracy scores are micro- and macro-averages over PIE types and the harmonic mean (HM) of the two.

6.5.2 Comparison to Previous Work

Our second research question considers the comparative performance of unsupervised and supervised methods. To answer this question, we evaluate our two best models on datasets used in previous work. A comprehensive overview of scores is presented in Table 6.5.

On the `VNC-test` dataset, our classifiers are clearly outperformed by existing systems, both supervised and unsupervised. Although they are competitive with the semi-supervised Context classifier of Fazly et al. (2009) and the all-expression supervised classifier of Gharbieh et al. (2016), the completely unsupervised approaches of the same authors do better. That supervised classifiers trained in a leave-one-token-out per-expression setup achieve better performance is unsurprising, but other unsupervised approaches also outperform our methods. A possible explanation here is that our classifiers are optimised on harmonic mean accuracy, whereas previous work focuses only on either micro- or macro-averaged accuracy.

On the SemEval datasets, the picture is very different. Here, our classifiers perform much better, the literalisation classifier in particular. In the 'known phrases' setting, where the same PIE types made up both training and test set, the best supervised classifiers still outperform our unsupervised approaches, but by much smaller margins than on the `VNC-test` dataset. Again, per-expression classifiers are better than all-expression classifiers. This fits in with the more general picture that supervised classifiers can achieve good performance, but only in per-expression settings. In the 'unknown phrases' setting, where training and test sets were made up of different PIE types, the literalisation classifier actually outperforms all other approaches. This implies that, in a realistic setting where we do not have training data for every PIE type, unsupervised approaches can be competitive with supervised approaches.

VNC-test

Model	Unsup.	Unseen	Macro	Micro
This work (original)	✓	✓	66.11	63.13
This work (literalisation)	✓	✓	64.67	67.37
C-Form (Fazly et al., 2009)	✓	✓	72.4	–
Context (Fazly et al., 2009)	✓/✗	✗	65.8	–
Sup (Fazly et al., 2009)	✗	✗	82.7	–
K-means (Gharbieh et al., 2016)	✓	✓	76.5	–
SVM-all (Gharbieh et al., 2016)	✗	✓	69.4	–
SVM-per (Gharbieh et al., 2016)	✗	✗	88.3	–

SemEval-lexsample-test (known phrases)

Model	Unsup.	Unseen	Macro	Micro
This work (original)	✓	✓	63.93	64.76
This work (literalisation)	✓	✓	69.14	72.18
Run 1 (Jimenez et al., 2013)	✗	✓	–	72.2
Run 2 (Jimenez et al., 2013)	✗	✗	–	75.4
Run 1 (Byrne et al., 2013)	✗	✗	–	53.0
Run 2 (Byrne et al., 2013)	✗	✗	–	50.2
Run 3 (Byrne et al., 2013)	✗	✗	–	77.9

SemEval-allwords-test (unknown phrases)

Model	Unsup.	Unseen	Macro	Micro
This work (original)	✓	✓	70.13	66.60
This work (literalisation)	✓	✓	66.35	69.77
Run 1 (Jimenez et al., 2013)	✗	✓	–	66.8
Run 2 (Jimenez et al., 2013)	✗	✗	–	64.5
Run 1 (Siblini and Kosseim, 2013)	✗	✓	–	55.0

Table 6.5: Comparison of scores reported in this and previous work, on various subcorpora. Scores are macro- and micro-averaged (over types) accuracy. 'Unsup.' indicates whether an approach is unsupervised or not, and 'Unseen' indicates whether an approach extends to unseen PIE types or not.

6.6 Conclusion

In this chapter, we have developed and evaluated a re-implemented and optimised version of the lexical cohesion graph classifier for disambiguation of potentially idiomatic expressions. In addition, we developed an alternative to that classifier making use of literalisations of PIE's figurative senses. By evaluating the systems in a comprehensive evaluation setup, we aimed to answer questions about the contribution of literalisations as an information source, and the potential of unsupervised systems to rival supervised systems' performance.

Considering the first research question, we have found that the current approach comparing the connectivity of PIEs and their literalisations by itself is not enough to outperform the original lexical cohesion graph classifier. However, both classifiers do well on different subsets of the data, meaning that literalisations are a useful information source, and there is potential for combining the two types of classification to achieve better performance. Moreover, literalisations are cheap to acquire for many different idiom types. Although we use manually created definitions, they can be acquired and refined automatically, as has been done by Liu and Hwa (2016). Further improvements in this line of research could be made by exploring different similarity measures, such as those tested by Ehren (2017) for German and to use more compositional representations of contexts and literalisations (see also Gharbieh et al. (2016)). This would allow for the information from verbs and modifiers to be used more effectively, as in its current form our method relies on word-to-word comparisons and only nouns contribute to performance.

As for the second research question, we find that unsupervised systems can compete with supervised systems that work for multiple PIE types, but not with supervised systems that work only for a single PIE type. In practice, a system will always have to deal with unseen phrases, limiting the usefulness of per-expression supervised methods. However, the usefulness of unsupervised is similarly limited, given their lower accuracy. As such, we consider semi-supervised or distantly supervised systems to have the greatest potential for combining both reliable performance and applicability to unseen phrases. For example, one could use an unsupervised system to label training data for use in a supervised classifier (Li and Sporleder, 2009; Fazly et al., 2009), or use knowledge bases to automatically collect training data, as has been done for word sense disambiguation (Pasini and Navigli, 2017). Finally, we find that, for evaluation, using both micro- and macro-averaged metrics is an important way of ensuring balanced performance on both infrequent and frequent PIE types and using the harmonic mean is a straightforward way of representing both measures in one metric. Moreover, evaluation can be improved by using a wider range of corpora than has been previously been the norm.

CHAPTER 7

Generalised Disambiguation of Potentially Idiomatic Expressions with Deep Learning

Abstract|The large sense-annotated corpus of PIEs developed in Chapter 5 enables the training of data-hungry PIE disambiguation models, such as those based on deep neural networks. In this chapter we experiment with various LSTM setups, focusing mainly on the representation of the task input. Our goal is to build a highly accurate model which can perform the disambiguation task in a general sense, i.e. extending its performance to unseen types and datasets. We find that a joint setup, combining an LSTM taking sentential context input with an LSTM using canonical form information, yields the best performance, reaching an accuracy score of 90.31% on development data and 87.93% on test data. Furthermore, the model's performance was improved by incorporating an attention mechanism in the network and by flagging the PIE's component words in the input. Comparing the best model's performance to existing approaches, we find that it is competitive, if not better than the previous state-of-the-art, but that the reliability of the comparison is hampered by the limitations of previously used evaluation datasets and metrics.

7.1 Introduction

PIE disambiguation[1] is an established task within NLP with a number of existing approaches. It has proven to be a highly challenging task for which it is difficult to design accurate models. Generally, their performance is inadequate, except when a single classifier for each type has been trained (see Section 3.4.3 for an overview). However, since these classifiers cannot generalise beyond a single type, they do not solve the task. A PIE disambiguation system should work for all types, both seen and unseen in training data, and be highly accurate across those types.

Given that we have created a new, large corpus with a high number of different types, we can improve on previous approaches by using more data-hungry training methods and using large-scale evaluation. The new corpus covers many types, of all kinds of syntactic forms, meaning that it is a better representation of the phenomenon of idiom as a whole. Therefore we attempt to build a model that solves the task properly, i.e. build one that can perform high-quality idiom disambiguation on unseen types across corpora.

Previous work has made use of word embeddings, but work using deep neural networks is rare. One example is the work on multi-task learning by Do Dinh et al. (2018), who combine PIE disambiguation in German with metaphor detection in English, ameliorating the problem of small datasets by using datasets for both tasks simultaneously. They find good results, which is promising: it implies that a general sense of ‘idiomaticity’ or ‘non-literality’ can be learned.

We create a deep neural network model for PIE disambiguation which ideally should be highly accurate, type-independent, and feature-light. Using the corpus described in Chapter 5, we can train such a model and evaluate it on a large number of different types and instances. We experiment with different LSTM setups, focusing on the question of how

[1] Otherwise described as idiom disambiguation, token-level idiom detection, idiom usage disambiguation in previous work (cf. Section 3.2).

to represent this task, i.e. how to approach a sentence labelling task in which certain words have a different function (PIE) than others (context). The model should not rely on expensive pre-processing steps: we limit it to the output of the corpus pre-extraction system.

In this chapter, we will describe the experimental setup, including data selection and splits, the evaluation metric, several existing classifiers, experimental results, and an in-depth analysis of the systems' performance. Ultimately, based on this work, our aim is to answer the following research questions:

1. Does a large annotated corpus enable the training of a type-general high-accuracy PIE disambiguation model?

2. Can such a model capture a general sense of idiomaticity, i.e. extend to unseen types and datasets?

3. Do deep neural networks outperform existing approaches on PIE disambiguation?

7.2 Data

We make use of the corpus described in Chapter 5 with a minimum confidence threshold of 0.75. The confidence value represents annotator quality and inter-annotator agreement, and we choose 0.75 as a threshold which yields as the best trade-off between label quality and corpus size. In addition, we include only instances with 'idiomatic' or 'literal' labels, to the exclusion of false extractions and non-binary labels like 'meta-linguistic'. This restricts the task to distinguishing between idiomatic and literal usages of PIEs, which is the main use case of an idiom disambiguation tool.

We use two different splits of the datasets into training (80%), development (10%), and test (10%) sets. In one, which we call random split (RS), there is overlap between the types in each subset; in the other, which we call type-based split (TBS), there is no overlap at all. This is used to separate performance on seen types from that on unseen types. In total, the corpus contains 48,395 instances across 1,738 different types. 75.07% of the instances are labelled idiomatic, the remainder literal, with an average annotation confidence score of 0.98.

7.3 Baseline Classifiers

Since there are no existing performance scores on this dataset, we run a set of baseline classifiers based on previous work to get a set of baseline results which can put our models' performance into perspective. We use the same combination of evaluation metrics as in Chapter 6, that is, macro-averaged accuracy over types, micro-averaged accuracy, and the harmonic mean of both. Performance should be good on both frequent and infrequent types, so we use the harmonic mean accuracy (HMA) as our main metric since it best reflects the overall quality.

The simplest baseline is a most-frequent sense baseline, which classifies each instance as idiomatic. It scores well, since the corpus is over 75% idiomatic. In addition, we test a most-frequent-sense-per-type baseline, which assigns the most frequent label for a given type in the training set to all instances of that type in the development set, backing off to 'idiomatic' in case of ties or a lack of data. It performs extremely well, reaching almost 92% harmonic mean accuracy. Scores for the various baseline classifiers are reported in Table 7.1.

We also reimplement the unsupervised cohesion-based method of Sporleder and Li (2009), both in its original form, and with optimised parameters (see Section 6.4 for details) The original approach does well on other corpora, but relatively poorly on our data, likely because it tends to generate fairly balanced label distributions. Optimising it helps, especially when a parameter is set to shift the predictions towards more idiomatic. Nevertheless, its performance remains well below the most-frequent sense baseline.

Model	Macro	Micro	HMA
MFS	85.13	75.57	80.07
MFS-per-type	**92.67**	**91.32**	**91.99**
Sporleder (original)	71.72	67.45	69.52
Sporleder (optimised)	74.94	70.59	72.70
Gharbieh	84.54	79.98	82.19
Gharbieh (+dictionary form)	**85.65**	**82.81**	**84.21**

Table 7.1: Baseline scores on the development set of the corpus, using the random split.

As unsupervised systems generally perform worse than supervised systems, a more competitive baseline is a state-of-the-art supervised all-expression classifier like the one developed by Gharbieh et al. (2016). It relies on word embeddings to generate representations of idiom types and contexts, in addition to a canonical form feature, both of which are fed to an SVM classifier. We reimplement their method, re-optimising the context size and embedding size parameters, and using a feature similar to canonical form, called dictionary form. Rather than extracting canonical forms from the corpus, this feature only indicates whether idioms are used in context in the exact same way as they are described in the dictionary.[2] We reimplement this system rather than use the original code for two reasons: the code is not publicly available, and the original implementation only works for verb-noun combinations, whereas our corpus contains many different syntactic types of idioms. The original method represents PIEs by their nouns and verbs. We do the same, but in cases where there are no nouns or verbs in the idiom, adjectives and adverbs are used. Instead of their embeddings, which are not available to us, we use GloVe embeddings (Pennington et al., 2014). The best-

[2] Of course, we do take into account that placeholder words like *someone's* do not need to be matched exactly, but rather match them to any corresponding pronoun or noun phrase.

performing setup has the same parameters as in the original paper: 1 context word on each side of the component word, and 300-dimensional embeddings. It performs well, clearly above the most-frequent sense baseline, and the dictionary-form feature benefits performance.

7.4 Model Architecture

The basis of our PIE disambiguation system is the Long-Short Term Memory (LSTM) network, a type of recurrent neural network (RNN) suited for longer sequences of input than regular RNNs. The input of our task is a sentence containing a PIE, and the desired output is a binary label indicating whether the PIE is literal or idiomatic. Given that these sentences can be lengthy, and relevant context might be found at some distance from the PIE itself, the LSTM architecture is particularly suited for this task.

In the context of our task, the most basic model looks like the one shown in Figure 7.1. The input consists of a snippet of in-sentence context, truncated to be of a fixed maximum length.[3] If the sentence length exceeds the maximum, it is truncated in a symmetric way, i.e. so that there are equal amounts of context words on each side of the PIE. For example, given Example 45 and a maximum length of 7, the input consists of the snippet '`man has made medical history by having`'. The input is represented by word embeddings, which are fed to a uni- or bidirectional LSTM layer. The LSTM layer encodes the input text into a single vector, which is fed to a softmax output layer, which predicts the final label.

(45) A man has **made medical history** by having four organ transplants. (*make history* - BNC - document K28 - sentence 402)

[3] In all reported results, context is limited to the sentence containing the PIE. However, we have experimented with context snippets crossing sentence boundaries, but these yielded consistently worse results in all cases.

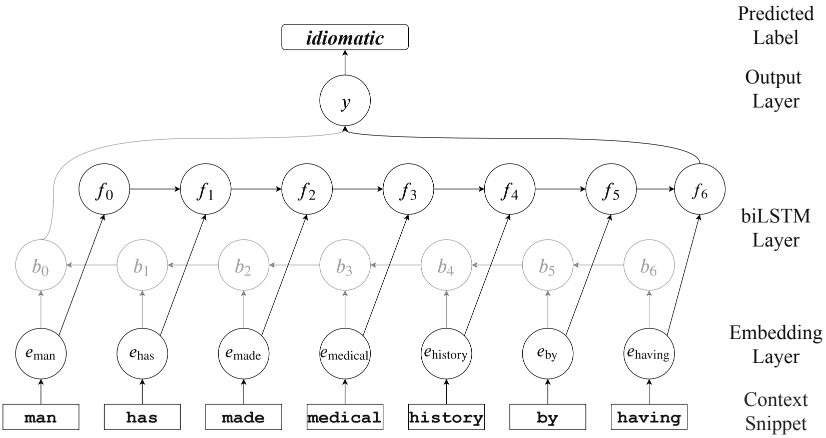


Figure 7.1: Schematic representation of the basic LSTM network, with an example input sentence and output label. Inclusion of the grey nodes and connections turns the uni-directional model into a bi-directional one.

In its default form, the model uses 300-dimensional GloVe embeddings pre-trained on 6 billion words of Wikipedia and Gigaword text (Pennington et al., 2014). Dropout is applied to the LSTM layer, both to the input and recurrent connections, with a default proportion of 0.2 for both. Other parameters and values are discussed in Section 7.5.

We make two extensions to the basic architecture: PIE encoding and an attention mechanism. PIE encoding is a modification of the input, consisting of adding an indicator feature which marks the individual words or whole span of the PIE. A 1 or 0 is added to the representation of each word (i.e., the embedding), indicating whether it is is part of the PIE. The component words are identified using the offsets generated by the PIE extraction system.

The second extension is an attention layer added after the LSTM layer in the way of Zhou et al. (2016). In this case, the LSTM does not encode the input snippet into a single vector at the end of the sentence, but in-

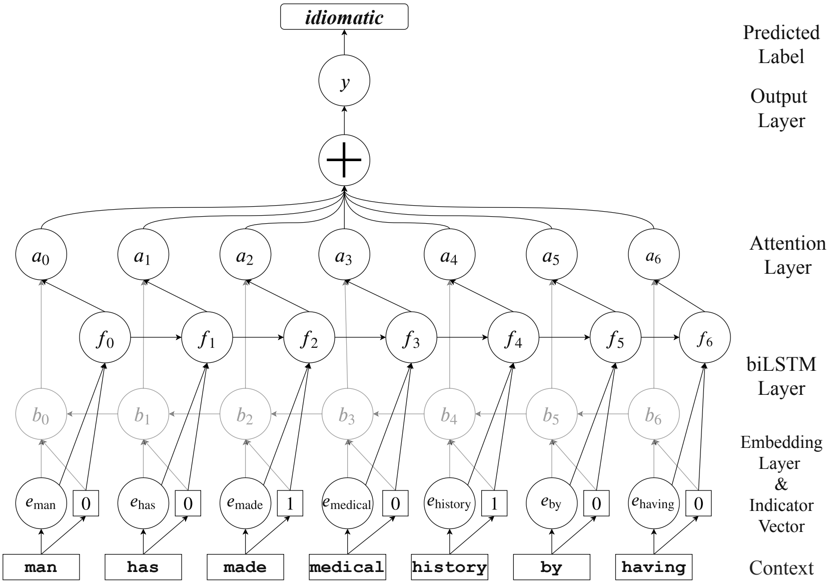

idiomatic
Predicted Label
y
Output Layer
Attention Layer
a_0 a_1 a_2 a_3 a_4 a_5 a_6
f_0 f_1 f_2 f_3 f_4 f_5 f_6
biLSTM Layer
b_0 b_1 b_2 b_3 b_4 b_5 b_6
Embedding Layer & Indicator Vector
e_{man} 0 e_{has} 0 e_{made} 1 e_{medical} 0 e_{history} 1 e_{by} 0 e_{having} 0
man has made medical history by having
Context

Figure 7.2: Schematic representation of the extended LSTM network, with an example input sentence and output label. The network includes an attention mechanism and word-level PIE encoding. Inclusion of the grey nodes and connections turns the uni-directional model into a bi-directional one.

stead provides an output vector at each step of the sentence. The attention mechanism then weighs these outputs, i.e. learning to pay attention only to relevant parts of the input, before they are combined and fed to the output layer. Figure 7.2 shows the network including both a word-level indicator feature and an attention mechanism. Note that the extensions are modular, i.e. the PIE encoding and attention mechanism can be used individually, but also together in one model.

7.5 Experimental Results

The first round of experiments aims to establish roughly optimal values for the most important parameters of the basic LSTM (cf. Figure 7.1). We use the random split of the corpus, which means there is overlap between the types in the training and development set. We examine four factors: the number of units in the hidden layer, uni- vs. bi-directional LSTM, the number of training epochs, and maximum context length.

Results are presented in Table 7.2. The best model uses 128 units, a uni-directional LSTM, a short context length of 16 words, and is trained for 20 epochs. It reaches a harmonic mean accuracy score of 91.88%, clearly better than the Sporleder and Gharbieh baselines, and almost on par with the most-frequent sense per type baseline.

Based on both the average and maximum scores with each parameter value, we can conclude that more units lead to better performance, uni-directional LSTMs outperform their bi-directional counterparts, and a short context of 16 words is better than longer contexts. In most cases, performance increases with the number of epochs, and the performance gain from 5 to 10 epochs is much larger than from 10 to 20 epochs. This indicates that 20 epochs is a suitable number of training iterations, since performance stabilises around that point while still being clearly better than 10 epochs.

For the second round of experiments, we test the same parameter value optimisations, but now extending the model with PIE encoding. There are two variants of this feature: word-level encoding, which indicates just the component words, and span-level encoding, which indicates the whole PIE span, including non-component words. Given the consistent small performance difference between uni- and bi-directional LSTMs, we only evaluate the uni-directional LSTM from here onwards. Similarly, given the consistent trend of 20 training epochs yielding the best performance, we only report scores after 20 training epochs from here onwards. Table 7.3 shows the results for the different models.

Units	Bi	Length	HMA-5	HMA-10	HMA-20
32	No	16	88.55	89.79	90.59
32	No	32	85.96	86.21	86.75
32	No	64	85.08	85.71	87.38
32	Yes	16	89.23	90.29	90.71
32	Yes	32	84.76	86.98	87.70
32	Yes	64	84.46	86.69	87.03
64	No	16	87.47	90.78	91.33
64	No	32	85.37	88.22	88.61
64	No	64	84.94	86.85	87.13
64	Yes	16	89.60	91.36	91.09
64	Yes	32	87.27	87.99	85.56
64	Yes	64	86.85	87.02	86.48
128	No	16	**90.61**	91.59	**91.88**
128	No	32	87.19	88.65	89.10
128	No	64	86.53	88.66	88.64
128	Yes	16	90.58	**91.60**	90.92
128	Yes	32	87.93	88.45	88.84
128	Yes	64	87.11	87.69	87.38

Table 7.2: Model performance when trained with different parameter values. Scores are harmonic mean accuracy on the development set (RS) after different numbers of training epochs. We use the off-the-shelf 300-dimensional glove-6B embeddings. Other, fixed parameter values: batch size 128, dropout 0.2, recurrent dropout 0.2.

Generally, performance is clearly better than without PIE encoding, by about 3 to 4 percentage points. The difference between word- and span-level encoding is small, but span-level encoding is slightly better on average and yields the best-performing model. As for the other parameter values, the patterns are similar to those for the LSTM without PIE encoding. More units and more epochs are better. However, with the encoding, the smallest context does not yield the best results. On the contrary, it seems that more context (i.e. a higher maximum sentence length)

Encoding	Units	Length	HMA@20
Word	32	16	93.31
Word	32	32	93.58
Word	32	64	93.45
Word	64	16	93.98
Word	64	32	94.11
Word	64	64	94.14
Word	128	16	93.95
Word	128	32	94.06
Word	128	64	94.29
Span	32	16	93.02
Span	32	32	93.23
Span	32	64	93.60
Span	64	16	93.98
Span	64	32	94.13
Span	64	64	93.58
Span	128	16	94.15
Span	128	32	94.32
Span	128	64	**94.68**

Table 7.3: Model performance when trained with different parameter values, using either span-level or word-level PIE encoding. Scores are harmonic mean accuracy on the development set (RS) after 20 training epochs. We use the off-the-shelf 300-dimensional glove-6B embeddings. Other, fixed parameter values: batch size 128, dropout 0.2, recurrent dropout 0.2, uni-directional LSTM.

boosts performance. A likely explanation for this is that the explicit encoding of PIE location means that the system's performance is not hurt by the fact that a larger context makes it more difficult to distinguish the target expression (PIE) from its context. Compared to the baseline figures the best model now clearly outperforms the MFS-per-type baseline, reaching close to 95% accuracy.

Even if the location of the PIE is encoded, part of the task is figuring out which parts of the context are relevant. Therefore, we extend the model with the attention mechanism described earlier (see also Figure 7.2). Using the attention layer, we experiment with the same parameter optimisations, in three separate settings: without PIE encoding, with word-level PIE encoding, and with span-level PIE encoding. Table 7.4 contains the results in either setting.

Encoding	Units	Length	HMA@20
None	64	16	91.98
None	64	32	90.07
None	64	64	89.43
None	128	16	92.08
None	128	32	90.03
None	128	64	89.87
Word	64	16	94.01
Word	64	32	94.04
Word	64	64	94.31
Word	128	16	94.65
Word	128	32	94.42
Word	128	64	94.57
Span	64	16	93.81
Span	64	32	93.79
Span	64	64	93.79
Span	128	16	94.44
Span	128	32	**94.81**
Span	128	64	94.54

Table 7.4: Model performance when trained with different parameter values, using either span-level, word-level, or no PIE encoding combined with an attention mechanism. Scores are harmonic mean accuracy on the development set (RS) after different numbers of training epochs. We use the off-the-shelf 300-dimensional glove-6B embeddings. Other, fixed parameter values: batch size 128, dropout 0.2, recurrent dropout 0.2, unidirectional LSTM.

On average, the attention layer slightly improves disambiguation performance. Attention makes the basic model a little ($< 1\%$) better, and similarly the models with PIE encoding show slight improvements. The best model with attention achieves 94.81% harmonic mean accuracy, compared to 94.68% without. The optimal parameter values are the same as earlier, with more units yielding better performance and a short context benefiting the model without PIE encoding. For the combined span encoding and attention model, a context length of 32 words is optimal. As before, span-level encoding yields marginally better results than word-level encoding. As such, we can conclude that attention by itself is not enough to both figure out the relevant parts of the context and the location of the target PIE.

7.5.1 Performance on Unseen Types

The remarkably high accuracy of our models, especially when compared to previous work, does not necessarily mean that the models actually capture PIE disambiguation in a general way. Rather, given the similarly high performance of the MFS-per-type baseline, it is more likely that the well-performing LSTM models mainly learn the idiom distribution per type, and not much more, i.e. they do not learn a general notion of idiomaticity. This is the reason we use two different splits of the corpus: the random split, where types in the development set have been seen during training, and the type-based split, where all types in the development and test sets are not in the training set.

Using the type-based split, we retrain and rerun the baseline classifiers and our best models. Note that the MFS-per-type baseline cannot be applied, since all types in the development set are unseen, so there is no type information to learn from. Scores for both baselines and LSTM models are presented in Table 7.5.

As expected, scores on the development set of the type-based split corpus are substantially lower than on the original random split. For most LSTM models, the drop is around 10 percentage points. For the baselines, the difference is smaller, both because their performance on

Model	Units	Length	HMA (RS)	HMA (TBS)
MFS	–	–	80.07	78.63
MFS-per-type	–	–	91.99	78.63
Sporleder (original)	–	–	69.52	70.40
Sporleder (optimised)	–	–	72.70	72.17
Gharbieh	–	–	82.19	78.44
Gharbieh (+dict. form)	–	–	84.21	80.26
Basic	128	16	91.88	81.64
Word	128	64	94.29	85.42
Span	128	64	94.68	83.84
Attention	128	16	92.08	80.81
Att.+Word	128	16	94.65	**86.41**
Att.+Span	128	32	**94.81**	83.68

Table 7.5: Comparison between scores of baseline classifiers and LSTM models on the random split dataset (RS) and the type-based split dataset (TBS). Scores represent harmonic mean accuracy (HMA). We use the off-the-shelf 300-dimensional glove-6B embeddings. Other, fixed parameter values: batch size 128, dropout 0.2, recurrent dropout 0.2, unidirectional LSTM, 20 epochs.

the random split corpus is lower, and because they are less reliant on information from the training data. However, since the MFS-per-type baseline does not apply here, the best baseline score is 80.26%, compared to 91.99% on the random split corpus. This means that all LSTM models now outperform the best baseline measure. Contrary to before, the models using word-level PIE encoding outperform those using span-level encoding, especially when combined with attention. Since span-level encoding includes determiners and insertions and word-level encoding does not, it is likely that span-level encoding makes the model more sensitive to the form of the PIE. The model using span-level encoding might be more dependent on specific forms of idioms in the training data, which makes it more difficult to generalise to unseen idioms in the development data.

It should be noted, however, that the parameter settings used are not optimised on the type-based split corpus, meaning performance on this set could be higher with different parameter values. Looking at the models' performance in detail, we see that there is substantial overfitting on the training data in the TBS setting; all models achieve >99% accuracy on the training data. As the development set contains only unseen types, the need for the model to generalise is much larger. To improve generalisation capability, we perform additional parameter optimisation. Using a default setup of 300-dimensional GloVe embeddings, 128 hidden units, word-level PIE encoding, and attention, we experiment with dropout levels, batch size, and maximum length, as well as the number of epochs. Results are presented in Table 7.6.

Based on the average and maximum scores for each parameter value, we make several observations. The effect of batch size is negligible, with a batch size of 128 perhaps providing a slight benefit. For epochs, the pattern is similar to before: the increase from 10 to 20 epochs is small but consistent, using more than 20 epochs yields no benefits. Given the need for increased generalisation capability, it is not surprising that higher dropout proportions than the 0.2 used before yield better performance, with 0.4 being the sweet spot between generalisation and modelling capacity. As for context length, a higher maximum is better, but only if there are more epochs to train on this larger context, perhaps to optimise the attention mechanism.

The best setup uses exactly this setup: 20 epochs, 0.4 dropout, 64 word context, 128 instance batch size, yielding a harmonic mean accuracy of 88.88%. Accuracy scores on the training set range from 92.43% to 99.81%. The highest-scoring model on the development set achieves 97.57% accuracy on the training set. As such, it seems that this model achieves a compromise between generalisation on the one hand, i.e. not approaching 100% accuracy on training set too closely and thoroughly learning from this dataset on the other hand. Compared to the previous best model, which scored 86.41% on the same dataset, it is clear that

Dropout	Length	Batch Size	HMA-5	HMA-10	HMA-20
0.2	16	128	**87.08**	86.41	84.51
0.2	16	256	85.93	87.50	85.73
0.2	16	512	84.07	87.04	86.71
0.2	32	128	86.61	**88.17**	87.09
0.2	32	256	86.15	88.07	86.96
0.2	32	512	84.14	86.93	87.12
0.2	64	128	86.38	87.89	86.52
0.2	64	256	85.73	87.83	87.26
0.2	64	512	84.03	87.00	88.27
0.4	16	128	85.71	86.05	87.09
0.4	16	256	84.56	86.75	87.49
0.4	16	512	83.67	85.38	86.30
0.4	32	128	85.90	86.41	88.38
0.4	32	256	84.92	87.79	88.16
0.4	32	512	83.83	85.33	87.05
0.4	64	128	85.64	87.38	**88.88**
0.4	64	256	85.54	87.41	87.46
0.4	64	512	83.93	85.07	87.00
0.6	16	128	83.87	85.04	86.81
0.6	16	256	83.28	84.37	86.61
0.6	16	512	83.10	84.49	86.32
0.6	32	128	84.16	85.82	86.53
0.6	32	256	83.85	85.15	86.69
0.6	32	512	82.59	84.18	85.00
0.6	64	128	84.23	86.17	85.77
0.6	64	256	83.63	84.70	86.44
0.6	64	512	82.01	84.02	85.29

Table 7.6: Model performance when trained with different parameter values, using word-level PIE encoding combined with an attention mechanism. Scores are harmonic mean accuracy on the development set (TBS) after different numbers of training epochs. Dropout value is both regular and recurrent dropout. We use the off-the-shelf 300-dimensional glove-6B embeddings. Other, fixed parameter values: 128 hidden units, uni-directional LSTM.

there is a lot to gain by optimising parameter values, and that the optimal values differ significantly between the random split and type-based split corpus.

7.5.2 Integrating Dictionary Form

The models developed until now have only used the sentence containing the PIE as input, represented by word embeddings with or without a binary vector encoding the position of the PIE. We do not want to use any information that is not part of the output of the pre-extraction system, but we can make use of dictionary form information, which is present in the output, especially given that existing research has shown that this information is helpful for PIE disambiguation (King and Cook, 2018, for example). The question remains of how to integrate such a feature into the existing LSTM setup. On the one hand, it is a sentence-level feature, since it is not tied to any particular word. On the other hand, it is a sequence in itself, composed of words in a meaningful order.

These two characteristics are brought together by feeding the dictionary form input to a separate (bi)LSTM and combining it with the sentence-LSTM. This way, the sequential nature of the dictionary form input is utilised, without being part of the sentence, but as an extra-sentential information source. The input to the dictionary-form-LSTM (df-LSTM) consists of a concatenation of the PIE form (i.e. component words) and the dictionary form, e.g. `on his broken nose on the nose`. These are represented by the same embeddings as in the other LSTM, which are fed to a (bi)LSTM layer. The (bi)LSTM output is then concatenated with the other LSTM's output, and this is passed to a softmax output layer to produce the final label, as illustrated in Figure 7.3.

We expect very different parameter values to be optimal for the df-LSTM than the sent-LSTM, so we optimise it in a separate setup where it solves the PIE disambiguation task using only the dictionary form and PIE form information. We then use the best parameter settings as fixed

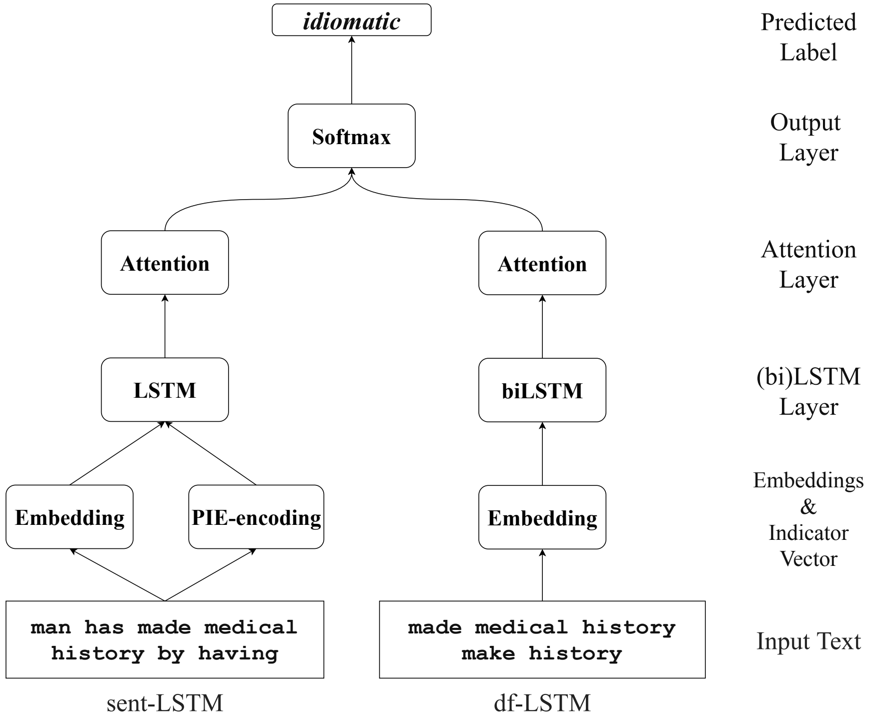


Figure 7.3: High-level schematic representation of the joint LSTM network in its final configuration, with example input snippets and output label.

parameters for the df-LSTM, and optimise the sent-LSTM again, combined with the fixed-parameter df-LSTM. The performance of the df-LSTM by itself is unexpectedly good – after optimising for batch size, number of hidden units, dropout proportion, and the use of attention, the best performing model achieves 88.41% accuracy. This means it is competitive with the sent-LSTM, which reaches 88.88%. The best parameter values for the df-LSTM are 8 hidden units, a batch size of 256, dropout proportion of 0.6, and using an attention mechanism.

Finally, we optimise the parameters of the sent-LSTM in the com-

Dropout	Length	Batch Size	HMA-5	HMA-10	HMA-20
0.2	32	128	88.53	88.95	88.03
0.2	32	256	88.28	89.58	88.63
0.2	64	128	**88.71**	**90.31**	**89.03**
0.2	64	256	88.25	88.54	86.59
0.4	32	128	87.40	89.10	88.19
0.4	32	256	87.42	87.96	88.13
0.4	64	128	87.31	88.62	87.66
0.4	64	256	86.92	88.20	88.47
0.6	32	128	87.55	88.41	87.61
0.6	32	256	87.03	87.53	87.79
0.6	64	128	87.25	88.98	87.75
0.6	64	256	86.78	88.19	88.43

Table 7.7: Model performance when trained with different parameter values, using word-level PIE encoding combined with an attention mechanism and combined sentence-level and dictionary form LSTMs. Scores are harmonic mean accuracy on the development set (TBS) after different numbers of training epochs. Dropout value is both regular and recurrent dropout. We use off-the-shelf 300-dimensional glove-6B embeddings. Other, fixed parameter values: 128 hidden units, uni-directional LSTM.

bined setup, called joint-LSTM. Resulting performance scores are displayed in Table 7.7. Although both models separately reach nearly 89% accuracy, the combination of the models in this setup does not yield much benefit, with the highest score being 89.03% at 20 epochs, and 90.31% at 10 epochs. The best parameter settings are similar to those of the sent-LSTM, with the only change being a lower dropout value.

To put this joint performance in perspective, we also evaluate a simple ensemble method. In this case, the models are trained separately, and their predictions combined by summing them together. This yields a small improvement over the separate models, to 89.54% accuracy. A possible explanation for this is that the two models get the same predictions correctly. However, they are both correct in only 79.85% of cases, e.g. on Example 46, and in just 4.28% of instances both models are wrong, e.g. on Example 47. As such, there is potential for improvement by combining the two models' judgements in the 15.87% of cases where they disagree. For example, only the sent-LSTM predicts the label for Example 48 correctly, whereas the opposite is true for Example 49. However, this potential is not exploited currently, neither in the joint-LSTM or the ensemble setup.

(46) Through the window beside him Owen could see a large rat sunning itself **on a mooring rope**. (*on the ropes* - BNC - document J10 - sentence 817)

(47) Why grow things you can't eat? Why water luxuriant canna lilies [..] trained **on ropes** to make living swags around the croquet lawn? (*on the ropes* - BNC - document FAJ - sentence 336)

(48) The Prime Minister is clearly now **on the ropes** and doesn't seem able to fight back. (*on the ropes* - BNC - document K3T - sentence 132)

(49) Jill, a two year old infant, discovers the joy of swinging **on a rope**. (*on the ropes* - BNC - document B29 - sentence 1168)

7.5.3 Held-out Test Set Performance

Assuming the highest scoring model on the development set is also the best model, we evaluate whether it generalises well within the corpus. That is, the model parameter values have been optimised for performance on the development set, and we test on the held-out test to assess whether these parameter settings make for a good, general model. On the test set, the model achieves a macro-accuracy of 89.47%, a micro-accuracy of 86.45%, making for a harmonic mean of 87.93%. This is in line with performance on the development set, where it achieves a macro-accuracy of 89.47%, a micro-accuracy of 91.30%, and a harmonic mean of 90.31%. On this basis we conclude that there are likely other, optimal parameter values for test set performance, but that it generalises well, at least within the corpus.

7.5.4 Performance on Other Corpora

Using the best model, which scores 90.31% on the development set, we evaluate on other corpora than our own, in order to facilitate comparison to previous work and assess generalisation capability (see Section 3.3 for an overview of these corpora). Note that we include only classifiers which can classify unseen expressions, to make for a meaningful comparison. Table 7.8 contains our models' performance and that of the best models in existing research.

Corpus	Model	Macro	Micro	HMA
VNC-dev	MFS	62.09	60.88	61.48
VNC-dev	Gharbieh-SVM	68.9	–	–
VNC-dev	Gharbieh-K	71.8	–	–
VNC-dev	Our Model	**73.89**	**72.85**	**73.37**
VNC-dev	Balanced	48.38	44.35	46.28
VNC-test	MFS	61.83	63.18	62.49
VNC-test	Gharbieh-SVM	69.4	–	–
VNC-test	Fazly	72.4	–	–
VNC-test	Gharbieh-K	**76.5**	–	–
VNC-test	Our Model	72.13	**73.16**	**72.64**
VNC-test	Balanced	53.63	53.03	53.33
SemEval-unseen-test	UNAL-1	–	55.0	–
SemEval-unseen-test	MFS	66.21	61.75	63.90
SemEval-unseen-test	CLaC-1	–	66.8	–
SemEval-unseen-test	Our Model	**78.42**	**75.92**	**77.15**
SemEval-unseen-test	Balanced	42.68	36.50	39.35

Table 7.8: Accuracy scores on three different corpora, comparing the best LSTM model developed in this work to the state-of-the-art on each corpus.

First, these results show that performance on other datasets is remarkably worse than on the development set of our own corpus, by almost 20 percentage points. This is the case despite our model being optimised on unseen expressions in the development set, meaning that it should be robust to the unseen types encountered in the VNC and SemEval data. As such, there must be another reason for the large drop in performance. Upon closer inspection, it becomes clear that our model is trained to generate very skewed distributions for almost all idiom types, because that is the situation in the corpus it is trained on. The VNC and SemEval corpora, however, were designed to contain mostly balanced types, to the exclusion of imbalanced types, which make up the majority of idioms. As a result, the model performs significantly worse on the VNC and SemEval data.

Second, when compared to previous work, our model performs very well. On the VNC-dev and SemEval data it outperforms the previous best model by some margin. On the VNC-test data, however, it lacks behind the Gharbieh-K model. It is unclear what this difference stems from, and it should be noted that these comparisons are based only on a single metric, which prevents us from getting more insight into the nature of performance differences.

The effect of training data distribution on our model begs the question of whether performance on the more balanced VNC and SemEval corpora can be improved by training on a more balanced subset of our own training data. To examine this possibility, we take the subset of our corpus containing types with at least 10% of each label. The resulting subset of data totals 14,009 instances, split across 11,503 instances and 356 types in the training set and 1,066 instances and 23 types in the development set. After optimising parameter values for the joint-LSTM on this data, it achieves a score of 73.24% harmonic mean accuracy on the development set of the 'balanced' corpus. However, this performance does not translate to the VNC and SemEval data, where it performs abominably poorly. Clearly, the loss of training data quantity in this case hampers the model's learning and generalisation capability.

7.6 Analysis

The best joint-LSTM model scores 90.31% harmonic mean accuracy on the development set. If this model has indeed learned to disambiguate PIEs in a general way, its performance should be robust to variations in the dataset, i.e. it should achieve similar accuracy on different subsets of the data. Here, we will look at various factors, including annotation confidence, genre, label distribution, and form variation.

Considering annotation confidence, one might expect instances on which annotators disagree to be similarly difficult for the LSTM model to disambiguate. However, almost all instances in the data have 100% annotation agreement, leaving too few instances with lower agreement levels to reliably discern any relation between annotation confidence and disambiguation accuracy.

With genre, on the contrary, there are clear differences. Average micro-accuracy on the spoken part of the BNC corpus is 86.08%, on the written part it is 89.61%, and on the PMB part it is 91.63%. It is likely that the transcribed spoken-language data is more difficult to disambiguate, given its noisy nature and shorter sentences. Looking at the different genres within the BNC written data, differences are relatively small, ranging from 87.55% on fiction to 93.07% on academic writing. Although the model is robust, there is a trend of higher performance on more formal, factual writing and lower performance on informal, creative writing.

Given the large drop in performance when evaluated on the VNC data as compared to the development data for all models, including ours, it is likely that there is a specific trait of these corpora causing the performance difference. There are two major differences: the VNC contains only verb-noun combinations, whereas our data contains idiom types of all kinds of syntactic patterns, and the VNC development and test sets contain only relatively balanced idiom types, i.e. types which are not exclusively literal or idiomatic.

To examine the first, we compare performance of the best model on VNC-type idioms in our data to that of non-VNCs. The 137 idiom types in the development data were manually labelled as VNC or non-VNC. Micro-averaged accuracy on the VNC idiom types is 89.64%, while the micro-averaged accuracy on the non-VNC types is 89.10%. As such, it is unlikely that the syntactic form of the idioms in the VNC data is the cause of the performance drop.

% Idiomatic	Accuracy (%)	# Instances
$0 - 10$	82.07	820
$10 - 30$	72.77	382
$30 - 70$	82.74	336
$70 - 90$	82.18	348
$90 - 100$	94.96	2,954

Table 7.9: Micro-averaged accuracy for instances of types with the given label distribution in the development set, with the number of instances in each distribution band. Note that the two middle frequency bands have been joined, to make up for the low number of instances in that band.

As for the second difference, in label distribution, we examine performance in more detail. By micro-averaging accuracy for instances of PIE types with a certain label distribution, we get an overview of how label distribution affects performance (Table 7.9). Here, a stronger effect appears. Performance is highest, by far, on the highly idiomatic types (90–100%), and clearly lower on all others. The most difficult idiom types are those with a minority of idiomatic instances (10–30%), whereas the other distribution groups, including the highly literal group, are somewhere in between. The VNC development and test corpora contain 60.94% and 63.30% idiomatic instances respectively, making it likely that the label distribution of the idiom types in the VNC corpora is the main cause of the lower performance of our model on that corpus.

The form of the PIE in its context, and the difference from its dictionary form, is a useful feature for disambiguation. Generally, the more similar the PIE to its dictionary form, the more likely it is to be used idiomatically. As such, we expect there might be a relation between disambiguation performance and form variation. Table 7.10 displays the average accuracy score for different categories of form variation. Two clear trends emerge: case variation and idioms written as one word with dashes yield higher performance than average, whereas larger deviations from dictionary form, such as determiner variation, inflection, and combined variation, yield below-average performance. There might be a direct effect of form variation on model performance, but it is more likely that larger variation indicates a higher proportion of mixed and literal labels, which we know causes our model's performance to drop.

% Variation	Accuracy (%)	# Instances
Identical	90.76	2284
Case	95.65	138
Dashes	94.48	453
Placeholder	90.21	143
Determiner	86.54	207
Inflection	84.66	678
Combined	87.05	757

Table 7.10: Micro-averaged accuracy for instances with the given category of form variation. Categories with fewer than 100 instances have been excluded.

Finally, we evaluate the generalisation capability of the model by looking at word-level overlap between types in the training and development data. For example, the model might be able to exploit partial overlap between idiom types by learning that *play ball* in a context containing the word ‘ball’ multiple times is likely literal, and extend this knowledge to instances of the unseen type *have a ball* in the development set. In order to quantify this, we assign a word overlap percentage to each idiom type in the development data. This percentage is the percentage of component words which are also found in idiom types in the training data. For example, *drop like flies* has an overlap percentage of 66.67%, since 'drop' and 'like' occur in training set idiom types, whereas 'flies' does not. The micro-averaged accuracy on the 3,198 instances of types with $\geq$75% overlap is 93.47%, whereas the score on the 1,642 instances with $<$75% overlap is clearly lower, at 88.15%. Although 75% is an arbitrary cutoff value, the pattern at different cutoff values is similar, indicating that the model does benefit from word-level overlap between idioms in training and development sets.

7.7 Conclusions

In this chapter, we have trained various LSTM models on a sense-annotated corpus of PIEs. These models were evaluated both on the corpus from Chapter 5, and on other, pre-existing corpora. We find that a joint setup of two LSTMs, one taking sentential context input, and another taking dictionary form input, combined with word-level PIE encoding and an attention mechanism, performs the best. Based on the results from our experiments and analysis, we can answer the three main research questions from Section 7.1.

First, we consider whether this large corpus enables training of a type-general, high-accuracy model. Given the high accuracy of over 90% on the development set, we conclude this is the case. Training on a smaller subset of the corpus yield drastically lower performance, indicating the size of the data is crucial in improving performance. Moreover, other models trained on the same data do not yield high accuracy scores either.

As for the second question, which concerns generalisation to unseen types and datasets, the answer is mixed. On the one hand, the fact that the development set contains only idiom types not present in the training data means that the model manages to generalise to unseen types of all kinds. On the other hand, we do see a clear effect of label distribution: although the model does work for more ambiguous idioms, performance clearly drops, which is corroborated by the results on the VNC and SemEval datasets.

Finally, we consider the performance of our best LSTM model as compared to existing PIE disambiguation approaches. The answer to this question is somewhat obscured by different works using different evaluation measures, none of which are the harmonic mean we use. Nevertheless, we can compare to some extent, and see that our model outperforms previous work on two of the three datasets. We have also reimplemented two existing approaches and evaluated them on our corpus, where the LSTM model outperforms the other model by a large margin. Clearly, a more consistent evaluation setup is needed to improve the quality of comparisons.

PART IV

Conclusions

work out (somehow) to result in a good conclusion; to finish positively. ● *Don't worry. I am sure that everything will work out somehow.* ● *Things always work out in the end.*

CHAPTER 8

All Things Considered

Idiomatic expressions are a fascinating, idiosyncratic, and colourful part of language. They are an integral part of language, making them a challenging research topic we cannot afford to ignore. As such, in this thesis, we have approached idioms from a broad perspective, with the general goal of benefiting further research on idioms in NLP. More concretely, we have built a large dataset of annotated idioms and develop more advanced idiom processing models. Our goals are summarised by the research questions posed in the introduction to this thesis, and we answer them here.

RQ 1: What constitutes a potential idiom extraction system, and how can it be evaluated?

Chapter 4 deals with both parts of this question, describing a PIE extraction system and the PIE corpus which enables its evaluation. More specifically, we have defined the task of dictionary-based PIE extraction, which involves extracting all potential instances of idioms, taken from one or more dictionaries, from a text. To this end, we combined several idiom dictionaries and quantified their mutual overlap. They showed only little overlap, ranging from 20% to 55%, leading to the clear conclusion that high-coverage extraction requires a set of multiple idiom dictionaries.

As for the extraction systems themselves, we have experimented with several types of systems, of increasing complexity. This includes straightforward string matching, string matching generalising over inflection, and several forms of parse-tree based extraction. Overall, parse-tree based extraction is clearly superior in terms of performance. However, while complex systems have higher recall, they are less precise, and vice versa. Therefore, we find that the best extraction system takes the best of both worlds, combining the high precision of inflectional string matching with the high recall of parse-tree based extraction. Evaluating these systems on the PIE corpus showed that the best combined system achieves an F1-score of 92.01%. The PIE corpus is a publicly available dataset containing 2,239 instances, approximately half of which are PIEs. The main contribution of this corpus is that it exhaustively annotates PIEs (for a given set of idioms) in a collection of texts, so that we can evaluate recall in addition to precision for the task of PIE extraction.

RQ 2: To what extent do automatic pre-extraction and crowdsourced annotation facilitate the construction of a large-scale idiom corpus?

Based on the corpus-building process described in Chapter 5, the answer to this question is that they facilitate large-scale corpus construction to a great extent, but the devil is in the detail. The pre-extraction system greatly reduces the amount of manual extraction and filtering work necessary to extract the PIEs initially, especially when compared to previous work, but it is not perfect; it still extracts a small but non-negligible number of non-PIEs. These non-PIEs have to be filtered out, either by manual effort by researchers, or by including it in the crowdsourcing task.

Similarly, crowdsourcing is what makes construction of a large idiom corpus (56,622 instances, in our case) possible in the first place, both time-wise and budget-wise. However, the task is complex, especially for non-expert annotators, and the presence of non-PIEs in the pre-extraction output further complicates the task. Nevertheless, we find that crowdsourcing can be done efficiently and successfully by carefully crafting instructions, training and selecting a fixed pool of crowdworkers, and developing a step-by-step annotation interface.

RQ 3: Can unsupervised idiom disambiguation methods, enriched with additional information, rival supervised methods' performance?

Chapter 6 discusses various extensions to an existing unsupervised disambiguation approach, and based on their performance, the answer to this question is generally negative. More precisely though, we find that unsupervised systems can compete with supervised systems that work for multiple PIE types, but not with supervised systems that work only for a single PIE type. Adding additional information does benefit system performance on certain instances, but it also introduces additional errors, compared to the system without additional information. This means there is potential for performance gains by effectively combining the two, and this should be explored in future work.

RQ 4: Do deep neural network methods provide the same performance improvements for idiom processing as for the processing of non-idiomatic text?

Our final question concerns the application of deep neural network approaches to the task of PIE disambiguation. We experiment with these approaches in Chapter 7. Using the corpus from Chapter 5, we were able to train a network combining two LSTMs, exploiting both contextual and idiom form information. Similarly, we could not have evaluated our model as extensively without making use of this corpus. Ultimately, our model achieved a harmonic mean accuracy score of over 90%. Given that the previous state-of-the-art model, when trained and evaluated on the same data, achieves 80.26%, the answer to **RQ 4** is a clear *yes*.

Outlook

In conclusion, this work shows the feasibility of building a large corpus of sense-annotated potentially idiomatic expressions, and the benefits such a corpus provides for further research. It provides the possibility for quick testing of hypotheses about the distribution and usage of idioms, it enables the training of data-hungry machine learning methods for PIE disambiguation systems, and it permits fine-grained, reliable evaluation of such systems. As such, we hope that this resource will be widely used, and that similar resources will be created for other phenomena, such as other types of multiword expressions, and for other languages.

As for future work, we see as the main avenue for progress a change of focus from idiom processing in isolation to idiom processing as part of broader NLP approaches. That is, evaluating idiom detection and disambiguation not by intrinsic accuracy, but by their effect on downstream tasks, such as sentiment analysis or machine translation. Similarly, idiom processing should be integrated on the modelling side as well. That is, idiom processing should be improved by building models which handle idiomatic expressions as well as non-idiomatic language, either natively or by having a dedicated idiom-processing subsystem.

Appendices

bring up the rear to walk behind everyone else; to be at the end of the line. (Originally referred to marching soldiers.) ● *Here comes John, bringing up the rear.*

APPENDIX A

Crowdsourcing Instructions

Overview

In this job, you will be presented with sentences on various topics, all containing an instance of a given expression. We would like to know the meaning of these instances in the given context, so that we can better understand them in the future.

Steps

1. Read the sentence and pay special attention to the phrase marked in red.

2. Read the question.

3. If you are not familiar with the idiom in question, mouseover the link at the bottom to get its meaning.

4. Determine whether the marked phrase in the sentence is used idiomatically (as an idiomatic expression) or literally (as a regular phrase).

5. If it's neither idiomatic nor literal, specify what it is instead.

Rules and Tips

There are **3** main answer options: **idiomatic**, **literal**, and **other**.

An idiomatic expression is a phrase which can have a meaning that is different from the meaning of its words. For example, spill the beans can mean "to reveal secrets", which has nothing to do with either spilling or beans. Almost all expressions you are presented with in this task are such idiomatic expressions. Your task is to determine whether the expression in a given sentence has its idiomatic meaning or its literal meaning.

Idiomatic

If the expression is clearly used in the same meaning as its definition describes, please mark it as **idiomatic**.

Literal

If the words of the expression have their regular meaning in the given sentence, please mark it as **literal**. For example, *spill the beans* in “I spilled the beans while I was opening the can.”

Other

If the expression is not used literally but it is not idiomatic either, select the **other**-option. This category is divided into three different types: **non-instance**, **non-standard usage**, or **unclear**.

Sometimes, the sentence does not contain the expression we are looking for. For example, *spill the beans* does not actually occur in the sentence “I spilled sauce on the beans.” Please mark these sentences as **non-instance**.

Context is needed to actually know what the expression means. If there is not enough context, like in short sentences, it can be unclear which meaning the expression is actually used in. In those cases, select the option **unclear**.

Quite rarely, for example in word play or other creative writing, the idiomatic expression is used in an unorthodox way. For example, *rule the roost* is used as a linguistic example in “This expression survives today as ’rule the roost’.” Another example is when the idiom in question is actually a different idiom, as *see the light* is in “I saw the light at the end of the tunnel”. Please mark these cases as **non-standard usage** and shortly describe their usage.

Examples

Idiomatic Example

Over the top in the sentence “I was pleased that he was more diplomatic afterwards instead of being as over the top as ever.” has the meaning **idiomatic**, because it has the meaning of “bold; beyond reasonable limits; outrageous”.

Literal Example

Over the top in the sentence “Arrange fruit and nuts over the top then , using a pastry brush , glaze carefully with apricot glaze” has the meaning **literal**, because there is an actual top over which things are arranged.

Non-instance Example

The sentence “The biggest are more than 1000 gross tonnes and can carry more than 1200 tonnes of tuna .” does not actually contain the idiom *carry the can*, so it is a **non-instance**. However, we do count modified expressions as instances. For example, *carry the can* in “He carried the over-filled petrol can back to the car.” should be marked as **literal**, even though it is very different from carry the can.

Unclear Example

Join the club in the sentence “Join the club!” does not have enough context to determine the meaning. It might as well be an expression of sympathy as an actual invitation to join a club, so it is **unclear**. Please, use this option sparingly.

Challenging Cases

Modification

Distinguishing between non-instances and instances is not always easy. In general, if the essence of the expression is preserved, it is counted as an instance. For example, *on the cards* in “He had debts on multiple expired credit cards”. Only when the whole idiom is lost, mark it as a **non-instance**, for example *on the cards* in “He relied on his daughter having the right cards. “.

Names

In cases where the idiom is partially or completely inside a name, it should be marked as a **non-instance**. Examples of this are *coals to Newcastle* in “The author’s new book was called ’Bringing Coals to Newcastle’. “ and *out of the blue* in “We came out of the National Blues Museum to catch some fresh air.”

Idiom-within-idiom

Sometimes the wrong idiom is identified in the context. This happens with very similar idioms, such as *on the ground* (meaning in the field) and *thin on the ground* (meaning hard to find). In those cases, please select **other** and **non-standard usage**.

No Literal Meaning

It can be tricky to distinguish **idiomatic** from **literal** if the sentence contains an idiom that does not have a realistic literal meaning. For example, there is no way *think the world (of someone)* could be interpreted literally. In these cases, please select **idiomatic**. If you’re unsure about this, please check the provided definition.

Bibliography

Abzianidze, L., Bjerva, J., Evang, K., Haagsma, H., van Noord, R., Ludmann, P., Nguyen, D.-D., and Bos, J. (2017). The Parallel Meaning Bank: Towards a multilingual corpus of translations annotated with compositional meaning representations. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 242–247, Valencia, Spain. Association for Computational Linguistics.

Ayto, J., editor (2009). *From the horse's mouth: Oxford dictionary of English Idioms*. Oxford University Press, Oxford; New York, 3rd edition.

Baldwin, T. (2005). The deep lexical acquisition of English verb-particle constructions. *Computer Speech and Language*, 19(4):398–414.

BNC Consortium (2007). The British National Corpus, version 3 (BNC XML Edition). Distributed by Bodleian Libraries, University of Oxford.

Bott, S. and Schulte im Walde, S. (2017). Factoring ambiguity out of the prediction of compositionality for German multi-word expressions. In *Proceedings of the 13th Workshop on Multiword Expressions (MWE 2017)*, pages 66–72, Valencia, Spain. Association for Computational Linguistics.

Boukobza, R. and Rappoport, A. (2009). Multi-word expression identification using sentence surface features. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 468–477, Singapore. Association for Computational Linguistics.

Burnard, L. (2007). Reference guide for the British National Corpus (XML edition).

Byrne, L., Fenlon, C., and Dunnion, J. (2013). IIRG: A naive approach to evaluating phrasal semantics. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 103–107, Atlanta, Georgia, USA. Association for Computational Linguistics.

Cap, F. (2017). Show me your variance and I tell you who you are - deriving compound compositionality from word alignments. In *Proceedings of the 13th Workshop on Multiword Expressions (MWE 2017)*, pages 102–107, Valencia, Spain. Association for Computational Linguistics.

Chen, D. and Manning, C. (2014). A fast and accurate dependency parser using neural networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 740–750, Doha, Qatar. Association for Computational Linguistics.

Constant, M., Eryiğit, G., Monti, J., Plas, L. v. d., Ramisch, C., Rosner, M., and Todirascu, A. (2017). Survey: Multiword expression processing: A survey. *Computational Linguistics*, 43(4):837–892.

Cook, P., Fazly, A., and Stevenson, S. (2008). The VNC-Tokens dataset. In *Proceedings of the LREC Workshop: Towards a shared task for Multiword Expressions*, pages 19–22.

Do Dinh, E.-L., Eger, S., and Gurevych, I. (2018). Killing four birds with two stones: Multi-task learning for non-literal language detection. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1558–1569. Association for Computational Linguistics.

Ehren, R. (2017). Literal or idiomatic? identifying the reading of single occurrences of German multiword expressions using word embeddings. In *Proceedings of the Student Research Workshop at the 15th Conference of the European Chapter of the Association for Computational Linguistics*, pages 103–112, Valencia, Spain. Association for Computational Linguistics.

Fadaee, M., Bisazza, A., and Monz, C. (2018). Examining the tip of the iceberg: A data set for idiom translation. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Fazly, A., Cook, P., and Stevenson, S. (2009). Unsupervised type and token identification of idiomatic expressions. *Computational Linguistics*, 35(1):61–103.

Fazly, A. and Stevenson, S. (2006). Automatically constructing a lexicon of verb phrase idiomatic combinations. In *11th Conference of the European Chapter of the Association for Computational Linguistics*, Trento, Italy. Association for Computational Linguistics.

Fernando, C. (1996). *Idioms and Idiomaticity*. Oxford University Press, Oxford.

Ferraresi, A., Zanchetta, E., Baroni, M., and Bernardini, S. (2008). Introducing and evaluating ukWaC, a very large web-derived corpus of English. In *Proceedings of the WAC4 Workshop at LREC 2008*, pages 47–54. European Languages Resources Association (ELRA).

Fischer, I. and Keil, M. (1996). Parsing decomposable idioms. In *COLING 1996 Volume 1: The 16th International Conference on Computational Linguistics*.

Flor, M. and Beigman Klebanov, B. (2018). Catching idiomatic expressions in EFL essays. In *Proceedings of the Workshop on Figurative Language Processing*, pages 34–44, New Orleans, Louisiana. Association for Computational Linguistics.

Gharbieh, W., Bhavsar, V., and Cook, P. (2016). A word embedding approach to identifying verb-noun idiomatic combinations. In *Proceedings of the 12th Workshop on Multiword Expressions*, pages 112–118, Berlin, Germany. Association for Computational Linguistics.

Glucksberg, S. (2001). *Understanding Figurative Language*. Oxford University Press, Oxford.

Gong, H., Bhat, S., and Viswanath, P. (2016). Geometry of compositionality. *CoRR*, abs/1611.09799.

Graff, D. and Cieri, C. (2003). English Gigaword. Web Download.

Grégoire, N. (2009). *Untangling Multiword Expressions: A study on the representation and variation of Dutch multiword expressions*. PhD thesis, Universiteit Utrecht.

Gulland, D. M. and Hinds-Howell, D., editors (2002). *The Penguin dictionary of English Idioms*. Penguin Books, London, 2nd edition.

Gustawsson, E. (2006). *Idioms Unlimited: A study of non-canonical forms of English verbal idioms in the British National Corpus.* PhD thesis, Göteborg University.

Haagsma, H., Nissim, M., and Bos, J. (2018). The Other Side of the Coin: Unsupervised Disambiguation of Potentially Idiomatic Expressions by Contrasting Senses. In *Proceedings of the Joint Workshop on Linguistic Annotation, Multiword Expressions and Constructions (LAW-MWE-CxG-2018)*, pages 178–184, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Haagsma, H., Nissim, M., and Bos, J. (2019). Casting a Wide Net: Robust Extraction of Potentially Idiomatic Expressions. arXiv:1911.08829.

Haagsma, H., Nissim, M., and Bos, J. (2020). MAGPIE: A Large Corpus of Potentially Idiomatic Expressions. In *Proceedings of LREC 2020.* (to appear).

Hovy, E., Marcus, M., Palmer, M., Ramshaw, L., and Weischedel, R. (2006). OntoNotes: The 90% solution. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, pages 57–60, New York City, USA. Association for Computational Linguistics.

Hwang, A. and Hidey, C. (2019). Confirming the non-compositionality of idioms for sentiment analysis. In *Proceedings of the Joint Workshop on Multiword Expressions and WordNet (MWE-WN 2019)*, pages 125–129, Florence, Italy. Association for Computational Linguistics.

Ide, N. and Véronis, J. (1994). Machine readable dictionaries: What have we learned, where do we go? In *Proceedings of the International Workshop on the Future of Lexical Research*, pages 137–146, Beijing, China.

Iñurrieta, U., Díaz de Ilarraza, A., Labaka, G., Sarasola, K., Aduriz, I., and Carroll, J. (2016). Using linguistic data for English and Spanish verb-noun combination identification. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 857–867, Osaka, Japan. The COLING 2016 Organizing Committee.

Isabelle, P., Cherry, C., and Foster, G. (2017). A challenge set approach to evaluating machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2476–2486. Association for Computational Linguistics.

Jimenez, S., Becerra, C., and Gelbukh, A. (2013). UNAL: Discriminating between literal and figurative phrasal usage using distributional statistics and POS tags. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 114–117, Atlanta, Georgia, USA. Association for Computational Linguistics.

Kato, A., Shindo, H., and Matsumoto, Y. (2018). Construction of large-scale English verbal multiword expression annotated corpus. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

King, M. and Cook, P. (2018). Leveraging distributed representations and lexico-syntactic fixedness for token-level prediction of the idiomaticity of English verb-noun combinations. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 345–350, Melbourne, Australia. Association for Computational Linguistics.

Köper, M. and Schulte im Walde, S. (2017). Applying multi-sense embeddings for German verbs to determine semantic relatedness and to detect non-literal language. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 535–542, Valencia, Spain. Association for Computational Linguistics.

Korkontzelos, I., Zesch, T., Zanzotto, F. M., and Biemann, C. (2013). SemEval-2013 task 5: Evaluating phrasal semantics. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 39–47, Atlanta, Georgia, USA. Association for Computational Linguistics.

Lee, D. (2000). Genres, registers, text types, domains and styles: Clarifying the concepts and navigating a path through the BNC jungle. In *Proceedings of the Fourth International Conference on Teaching and Language Corpora*, Language and Computers, pages 245–292. Brill | Rodopi.

Li, L. and Sporleder, C. (2009). Classifier combination for contextual idiom detection without labelled data. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 315–323, Singapore. Association for Computational Linguistics.

Liebeskind, C. and HaCohen-Kerner, Y. (2016). Semantically motivated Hebrew verb-noun multi-word expressions identification. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1242–1253, Osaka, Japan. The COLING 2016 Organizing Committee.

Liu, C. and Hwa, R. (2016). Phrasal substitution of idiomatic expressions. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 363–373, San Diego, California. Association for Computational Linguistics.

Liu, C. and Hwa, R. (2018). Heuristically informed unsupervised idiom usage recognition. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1723–1731. Association for Computational Linguistics.

Liu, D. (2003). The most frequently used spoken American English idioms: A corpus analysis and its implications. *TESOL Quarterly*, 37(4):671–700.

Liu, P., Qian, K., Qiu, X., and Huang, X. (2017). Idiom-aware compositional distributed semantics. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1204–1213. Association for Computational Linguistics.

Liu, Y., Pang, B., and Liu, B. (2019). Neural-based Chinese idiom recommendation for enhancing elegance in essay writing. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5522–5526, Florence, Italy. Association for Computational Linguistics.

Manning, C. D. (2015). Last words: Computational linguistics and deep learning. *Computational Linguistics*, 41(4):701–707.

McCarthy, M. (1998). *Spoken language and applied linguistics.* Cambridge University Press, Cambridge.

Minnen, G., Carroll, J., and Pearce, D. (2001). Applied morphological processing of English. *Natural Language Engineering*, 7(3):207–223.

Minugh, D. (1999). The frequency of idioms in newspaper CDs as corpora. In Kirk, J. M., editor, *Corpora Galore: Analyses and Techniques in Describing English*, pages 57–72. Rodopi, Amsterdam.

Minugh, D. (2002). The COLL corpus: Towards a corpus of web-based college student newspapers. In Peters, P., Collins, P., and Smith, A., editors, *New frontiers of corpus research*, pages 71–90. Rodopi, Amsterdam.

Minugh, D. (2007). The filling in the sandwich: internal modification of idioms. In Facchinetti, R., editor, *Corpus Linguistics 25 Years on*, pages 205–224. Rodopi, Amsterdam.

Minugh, D. (2008). The college idiom: Idioms in the COLL corpus. *ICAME Journal: Computers in English Linguistics*, 32:115–138.

Moon, R. (1998). *Fixed Expressions and Idioms in English: A Corpus-Based Approach.* Oxford University Press, Oxford.

Muzny, G. and Zettlemoyer, L. (2013). Automatic idiom identification in Wiktionary. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1417–1421, Seattle, Washington, USA. Association for Computational Linguistics.

Nissim, M. and Zaninello, A. (2013). Modeling the internal variability of multiword expressions through a pattern-based method. *ACM Transactions on Speech and Language Processing (TSLP)*, 10(2):7.

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

Pasini, T. and Navigli, R. (2017). Train-o-Matic: Large-scale supervised word sense disambiguation in multiple languages without manual training data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 78–88, Copenhagen, Denmark. Association for Computational Linguistics.

Pasquer, C., Savary, A., Ramisch, C., and Antoine, J.-Y. (2018). If you've seen some, you've seen them all: Identifying variants of multiword expressions. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2582–2594, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Paul, D. B. and Baker, J. M. (1992). The design for the Wall Street Journal-based CSR corpus. In *Speech and Natural Language: Proceedings of a Workshop Held at Harriman, New York, February 23-26, 1992.*

Peng, J., Feldman, A., and Jazmati, H. (2015). Classifying idiomatic and literal expressions using vector space representations. In *Proceedings of the International Conference Recent Advances in Natural Language Processing*, pages 507–511, Hissar, Bulgaria. INCOMA Ltd. Shoumen, Bulgaria.

Pennington, J., Socher, R., and Manning, C. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Pershina, M., He, Y., and Grishman, R. (2015). Idiom paraphrases: Seventh heaven vs cloud nine. In *Proceedings of the First Workshop on Linking Computational Models of Lexical, Sentential and Discourse-level Semantics*, pages 76–82, Lisbon, Portugal. Association for Computational Linguistics.

Ramisch, C., Cordeiro, S. R., Savary, A., Vincze, V., Barbu Mititelu, V., Bhatia, A., Buljan, M., Candito, M., Gantar, P., Giouli, V., Güngör, T., Hawwari, A., Iñurrieta, U., Kovalevskaitė, J., Krek, S., Lichte, T., Liebeskind, C., Monti, J., Parra Escartín, C., QasemiZadeh, B., Ramisch, R., Schneider, N., Stoyanova, I., Vaidya, A., and Walsh, A. (2018). Edition 1.1 of the PARSEME shared task on automatic identification of verbal multiword expressions. In *Proceedings of the Joint Workshop on Linguistic Annotation, Multiword Expressions and Constructions (LAW-MWE-CxG-2018)*, pages 222–240, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Reppen, R., Ide, N., and Suderman, K. (2005). American National Corpus (ANC) second release.

Riehemann, S. Z. (2001). *A Constructional Approach to Idioms and Word Formation.* PhD thesis, Stanford University.

Sag, I. A., Baldwin, T., Bond, F., Copestake, A., and Flickinger, D. (2002). Multiword expressions: A pain in the neck for NLP. In *Proceedings of CICLING*, pages 1–15.

Salehi, B., Cook, P., and Baldwin, T. (2014a). Detecting non-compositional MWE components using Wiktionary. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1792–1797, Doha, Qatar. Association for Computational Linguistics.

Salehi, B., Cook, P., and Baldwin, T. (2014b). Using distributional similarity of multiway translations to predict multiword expression compositionality. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 472–481, Gothenburg, Sweden. Association for Computational Linguistics.

Salehi, B., Cook, P., and Baldwin, T. (2015). A word embedding approach to predicting the compositionality of multiword expressions. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 977–983, Denver, Colorado. Association for Computational Linguistics.

Salton, G., Ross, R., and Kelleher, J. (2014a). An empirical study of the impact of idioms on phrase based statistical machine translation of English to Brazilian-Portuguese. In *Proceedings of the 3rd Workshop on Hybrid Approaches to Machine Translation (HyTra)*, pages 36–41, Gothenburg, Sweden. Association for Computational Linguistics.

Salton, G., Ross, R., and Kelleher, J. (2014b). Evaluation of a substitution method for idiom transformation in statistical machine translation. In *Proceedings of the 10th Workshop on Multiword Expressions (MWE)*, pages 38–42, Gothenburg, Sweden. Association for Computational Linguistics.

Salton, G., Ross, R., and Kelleher, J. (2016). Idiom token classification using sentential distributed semantics. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 194–204, Berlin, Germany. Association for Computational Linguistics.

Salton, G., Ross, R., and Kelleher, J. (2017). Idiom type identification with smoothed lexical features and a maximum margin classifier. In *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*, pages 642–651, Varna, Bulgaria. INCOMA Ltd.

Savary, A., Candito, M., Mititelu, V. B., Bejček, E., Cap, F., Čéplö, S., Cordeiro, S. R., Eryiğit, G., Giouli, V., van Gompel, M., HaCohen-Kerner, Y., Kovalevskaitė, J., Krek, S., Liebeskind, C., Monti, J., Escartín, C. P., van der Plas, L., QasemiZadeh, B., Ramisch, C., Sangati, F., Stoyanova, I., and Vincze, V. (2018). *PARSEME multilingual corpus of verbal multiword expressions*, chapter 4, pages 87–147. Language Science Press., Berlin.

Savary, A. and Cordeiro, S. R. (2017). Literal readings of multiword expressions: as scarce as hen's teeth. In *Proceedings of the 16th International Workshop on Treebanks and Linguistic Theories*, pages 64–72, Prague, Czech Republic.

Schneider, N., Hovy, D., Johannsen, A., and Carpuat, M. (2016). SemEval-2016 task 10: Detecting minimal semantic units and their meanings (DiMSUM). In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 546–559, San Diego, California. Association for Computational Linguistics.

Senaldi, M. S. G., Lebani, G. E., and Lenci, A. (2016). Lexical variability and compositionality: Investigating idiomaticity with distributional semantic models. In *Proceedings of the 12th Workshop on Multiword Expressions*, pages 21–31, Berlin, Germany. Association for Computational Linguistics.

Seretan, V. (2008). *Collocation Extraction Based on Syntactic Parsing*. PhD thesis, University of Geneva.

Siblini, R. and Kosseim, L. (2013). ClaC: Semantic relatedness of words and phrases. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 108–113, Atlanta, Georgia, USA. Association for Computational Linguistics.

Simpson, R. C., Briggs, S. L., Ovens, J., and Swales, J. M. (1999). The Michigan Corpus of Academic Spoken English.

Simpson, R. C. and Mendis, D. (2003). A corpus-based study of idioms in academic speech. *TESOL Quarterly*, 37(3):419–441.

Spasić, I., Williams, L., and Buerki, A. (2017). Idiom-based features in sentiment analysis: Cutting the Gordian knot. *IEEE Transactions on Affective Computing*.

Sporleder, C. and Li, L. (2009). Unsupervised recognition of literal and non-literal use of idiomatic expressions. In *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*, pages 754–762, Athens, Greece. Association for Computational Linguistics.

Sporleder, C., Li, L., Gorinski, P., and Koch, X. (2010). Idioms in context: The IDIX corpus. In *Proceedings of the Seventh International Conference on Language Re-*

sources and Evaluation (LREC'10), Valletta, Malta. European Language Resources Association (ELRA).

Street, L., Michalov, N., Silverstein, R., Reynolds, M., Ruela, L., Flowers, F., Talucci, A., Pereira, P., Morgon, G., Siegel, S., Barousse, M., Anderson, A., Carroll, T., and Feldman, A. (2010). Like finding a needle in a haystack: Annotating the American National Corpus for idiomatic expressions. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta. European Language Resources Association (ELRA).

Tsvetkov, Y. and Wintner, S. (2010). Extraction of multi-word expressions from small parallel corpora. In *Coling 2010: Posters*, pages 1256–1264, Beijing, China. Coling 2010 Organizing Committee.

Verma, R. and Vuppuluri, V. (2015). A new approach for idiom identification using meanings and the web. In *Proceedings of the International Conference Recent Advances in Natural Language Processing*, pages 681–687, Hissar, Bulgaria. INCOMA Ltd. Shoumen, BULGARIA.

Villada Moirón, B. (2005). *Data-driven Identification of Fixed Expressions and their Modifiability*. PhD thesis, University of Groningen.

Villada Moirón, B. and Tiedemann, J. (2006). Identifying idiomatic expressions using automatic word-alignment. In *Proceedings of the Workshop on Multi-word-expressions in a multilingual context*.

Villavicencio, A. (2005). The availability of verb–particle constructions in lexical resources: How much is enough? *Computer Speech and Language*, 19(4):415–432.

Waszczuk, J., Ehren, R., Stodden, R., and Kallmeyer, L. (2019). A neural graph-based approach to verbal MWE identification. In *Proceedings of the Joint Workshop on Multiword Expressions and WordNet (MWE-WN 2019)*, pages 114–124, Florence, Italy. Association for Computational Linguistics.

Weeds, J., Kober, T., Reffin, J., and Weir, D. (2017). When a red herring in not a red herring: Using compositional methods to detect non-compositional phrases. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 529–534, Valencia, Spain. Association for Computational Linguistics.

Williams, J. (2017). Boundary-based MWE segmentation with text partitioning. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, pages 1–10, Copenhagen, Denmark. Association for Computational Linguistics.

Williams, L., Bannister, C., Arribas-Ayllon, M., Preece, A., and Spasić, I. (2015). The role of idioms in sentiment analysis. *Expert Systems with Applications*, 42(21):7375–7385.

Zhou, P., Shi, W., Tian, J., Qi, Z., Li, B., Hao, H., and Xu, B. (2016). Attention-based bidirectional long short-term memory networks for relation classification. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 207–212, Berlin, Germany. Association for Computational Linguistics.

Summary

In recent years, great progress has been made in the quality of natural language processing (NLP) systems, both in accuracy and practical applicability, mainly due to the surge of deep neural network methods. Generally, mainstream text in major languages can now be processed reliably, meaning that it is time for research to move on to more challenging topics. This includes non-canonical domains, such as social media text, under-resourced and minority languages, and challenging language phenomena like sarcasm, metaphor and idiom.

In this thesis, we are concerned with the last topic, namely idiomatic expressions and how to handle them within NLP. Idiomatic expressions are a type of multiword phrase with specific characteristics, such as not allowing for much lexical and syntactic variation, and having a meaning that is not a direct combination of the meaning of its parts. Examples of such expressions are *make hay while the sun shines*, *at a crossroads*, and *move the goalposts*. Due to their relative scarcity, idioms might seem a marginal area for research, but they do in fact pose a significant problem for a wide range of applications in natural language processing, including machine translation, semantic parsing, and sentiment analysis.

We aim to improve the automatic processing of idioms in two main ways. First, we collect a large number of idiom instances to get a more representative picture, which in turn can inform additional idiom processing models. Second, we come up with models which can detect the meaning of idiom instances in text in a general way — dealing well with both unseen and seen expressions.

In Part I, we provide a general introduction to idiomatic expressions and an overview of observations regarding idioms based on corpus data (Chapter 2). In addition, we discuss existing research on idioms from an NLP perspective (Chapter 3), providing an overview of existing tasks, approaches, and datasets. This informs us what the state of the art in automatic idiom processing is and where the most promising avenues for research lie.

In Part II of this thesis, we focus on the building of a large idiom corpus, consisting of two stages. In Chapter 4, we develop a system for the automatic extraction of potentially idiom expressions and annotate a small corpus to evaluate such a system. Ultimately, this system serves as a pre-processing step in the creation of a large idiom corpus. We build such a corpus using a crowdsourced annotation setup in Chapter 5.

Finally, in Part III, we consider both unsupervised and supervised methods for the disambiguation of potentially idiomatic expressions. In Chapter 6, we improve an extend an existing unsupervised classifier and compare it to other, existing classifiers, including supervised ones. Given the relatively poor performance of this unsupervised classifier, we develop a supervised deep neural network-based system in Chapter 7. We find that a model involving two separate modules looking at different information sources yield the best performance, surpassing previous state-of-the-art approaches.

In conclusion, this work shows the feasibility of building a large corpus of sense-annotated potentially idiomatic expressions, and the benefits such a corpus provides for further research. It provides the possibility for quick testing of hypotheses about the distribution and usage of idioms, it enables the training of data-hungry machine learning methods for PIE disambiguation systems, and it permits fine-grained, reliable evaluation of such systems. As such, we hope that this resource will be widely used, and that similar resources will be created for other phenomena, such as other types of multiword expressions, and for other languages.

Samenvatting

De laatste jaren is er grote vooruitgang geboekt in de kwaliteit van natuurlijke taalverwerkingssystemen, zowel qua nauwkeurigheid en kwaliteit als praktische toepasbaarheid. Dit is hoofdzakelijk te danken aan de opkomst van zogeheten diepe neurale netwerken. In het algemeen kan tekst in standaardtaal tegenwoordig goed worden verwerkt door deze netwerken, waardoor de tijd rijp is voor onderzoek op meer uitdagende vlakken, zoals de taal van sociale media, minderheidstalen en taalverschijnselen zoals sarcasme, metafoor en idioom.

In dit proefschrift worden idiomatische uitdrukkingen in het Engels onderzocht en hoe deze binnen de natuurlijke taalverwerking moeten worden behandeld. Idiomatische uitdrukkingen zijn woordgroepen met specifieke kenmerken: zo staan ze weinig variatie toe qua woorden en syntaxis en hebben ze een betekenis die niet direct gebaseerd is op de de betekenis van de losse woorden. Voorbeelden van zulke uitdrukkingen zijn het Nederlandse *ongelikte beer* en *aan het kortste eind trekken*, en het Engelse *make hay while the sun shines*, *at a crossroads*, en *move the goalposts*. Idiomen zijn problematisch voor een breed scala aan toepassingen binnen de natuurlijke taalverwerking, zoals automatisch vertalen, het begrijpen van teksten en het analyseren van de emotionele lading van een tekst.

Het doel van dit onderzoek is om de automatische verwerking van idiomen op twee manieren te verbeteren. Eerst verzamelen we een groot aantal voorbeelden van idiomen uit teksten, met context, om een representatief beeld te krijgen van het fenomeen. Deze dataset maakt het mogelijk om nieuwe modellen voor idioomverwerking te trainen, bijvoorbeeld modellen die de betekenis van idiomen in context kunnen herleiden. Het doel is om dit op zo'n manier te doen dat zowel uitdrukkingen die niet in de dataset voorkomen als uitdrukkingen die wel in de dataset voorkomen goed kunnen worden geïnterpreteerd.

Deel I bestaat uit een algemene inleiding over idiomatische uitdrukkingen en geeft een overzicht van bestaand onderzoek naar idiomen dat gedaan is op basis van grote tekstverzamelingen (Hoofdstuk 2). Daarna wordt bestaand onderzoek naar de automatische verwerking van idiomen besproken in Hoofdstuk 3. Dit hoofdstuk geeft een overzicht van bestaande taken, systemen en datasets op dit gebied. Samengenomen biedt dit een overzicht van wat de stand van zaken is in het vakgebied en waar de meeste ruimte voor verder onderzoek ligt.

In Deel II richten we ons op het maken van een grote dataset van idiomen uit tekst, met context. In Hoofdstuk 4 ontwikkelen we een systeem om automatisch mogelijke idiomen uit tekst te halen en annoteren we handmatig een kleine dataset om dit systeem te kunnen evalueren. Later fungeert dit systeem als voorbereidingsstap bij het bouwen van een grote idiomendataset. Deze dataset wordt geannoteerd met behulp van crowdsourcing, d.w.z. door grote aantallen leken. De opzet van de annotatie en een analyse van de dataset worden beschreven in Hoofdstuk 5.

Tot slot kijken we in Deel III naar zowel 'unsupervised' als 'supervised' methodes voor het interpreteren van idiomatische uitdrukkingen. Supervised methodes leren op basis van data met labels (in dit geval, de betekenis van een idioom), terwijl unsupervised methodes leren op basis van data zonder zulke labels. In Hoofdstuk 6 verbeteren we een bestaand unsupervised systeem en vergelijken we deze met andere bestaande systemen. Gezien de relatief slechte prestaties van het unsupervised systeem ontwikkelen we in Hoofdstuk 7 een supervised model gebaseerd op diepe neurale netwerken. Hieruit blijkt dat een model met twee afzonderlijke modules, die elk naar verschillende informatiebronnen kijken, het best presteert, beter dan het beste bestaande systeem.

Al met al laat dit werk de haalbaarheid van het creëren van een grote dataset met potentieel idiomatische uitdrukkingen zien en de voordelen die zo’n dataset biedt voor verder onderzoek. Het biedt de mogelijkheid om snel theorieën over idiomen te testen, het maakt het mogelijk om data-verslindende methoden zoals diepe neurale netwerken te gebruiken en het zorgt ervoor dat systemen voor het interpreteren van idiomen uitgebreider en nauwkeurig geëvalueerd kunnen worden. Daarom hopen we dat deze dataset veel zal worden gebruikt om systemen beter te kunnen evalueren en met elkaar te kunnen vergelijken, om zo het onderzoek naar het automatisch verwerken van idiomen vooruit te helpen.

Groningen dissertations in linguistics (GRODIL)

1. Henriëtte de Swart (1991). *Adverbs of Quantification: A Generalized Quantifier Approach*.
2. Eric Hoekstra (1991). *Licensing Conditions on Phrase Structure*.
3. Dicky Gilbers (1992). *Phonological Networks. A Theory of Segment Representation*.
4. Helen de Hoop (1992). *Case Configuration and Noun Phrase Interpretation*.
5. Gosse Bouma (1993). *Nonmonotonicity and Categorial Unification Grammar*.
6. Peter I. Blok (1993). *The Interpretation of Focus*.
7. Roelien Bastiaanse (1993). *Studies in Aphasia*.
8. Bert Bos (1993). *Rapid User Interface Development with the Script Language Gist*.
9. Wim Kosmeijer (1993). *Barriers and Licensing*.
10. Jan-Wouter Zwart (1993). *Dutch Syntax: A Minimalist Approach*.
11. Mark Kas (1993). *Essays on Boolean Functions and Negative Polarity*.
12. Ton van der Wouden (1994). *Negative Contexts*.
13. Joop Houtman (1994). *Coordination and Constituency: A Study in Categorial Grammar*.
14. Petra Hendriks (1995). *Comparatives and Categorial Grammar*.
15. Maarten de Wind (1995). *Inversion in French*.
16. Jelly Julia de Jong (1996). *The Case of Bound Pronouns in Peripheral Romance*.
17. Sjoukje van der Wal (1996). *Negative Polarity Items and Negation: Tandem Acquisition*.
18. Anastasia Giannakidou (1997). *The Landscape of Polarity Items*.
19. Karen Lattewitz (1997). *Adjacency in Dutch and German*.
20. Edith Kaan (1997). *Processing Subject-Object Ambiguities in Dutch*.
21. Henny Klein (1997). *Adverbs of Degree in Dutch*.
22. Leonie Bosveld-de Smet (1998). *On Mass and Plural Quantification: The case of French ‘des’/‘du’-NPs*.
23. Rita Landeweerd (1998). *Discourse semantics of perspective and temporal structure*.
24. Mettina Veenstra (1998). *Formalizing the Minimalist Program*.
25. Roel Jonkers (1998). *Comprehension and Production of Verbs in aphasic Speakers*.
26. Erik F. Tjong Kim Sang (1998). *Machine Learning of Phonotactics*.
27. Paulien Rijkhoek (1998). *On Degree Phrases and Result Clauses*.
28. Jan de Jong (1999). *Specific Language Impairment in Dutch: Inflectional Morphology and Argument Structure.*
29. H. Wee (1999). *Definite Focus.*
30. Eun-Hee Lee (2000). *Dynamic and Stative Information in Temporal Reasoning: Korean tense and aspect in discourse.*

31. Ivilin P. Stoianov (2001). *Connectionist Lexical Processing.*
32. Klarien van der Linde (2001). *Sonority substitutions.*
33. Monique Lamers (2001). *Sentence processing: using syntactic, semantic, and thematic information*.
34. Shalom Zuckerman (2001). *The Acquisition of "Optional" Movement*.
35. Rob Koeling (2001). *Dialogue-Based Disambiguation: Using Dialogue Status to Improve Speech Understanding*.
36. Esther Ruigendijk (2002). *Case assignment in Agrammatism: a cross-linguistic study.*
37. Tony Mullen (2002). *An Investigation into Compositional Features and Feature Merging for Maximum Entropy-Based Parse Selection.*
38. Nanette Bienfait (2002). *Grammatica-onderwijs aan allochtone jongeren.*
39. Dirk-Bart den Ouden (2002). *Phonology in Aphasia: Syllables and segments in level-specific deficits.*
40. Rienk Withaar (2002). *The Role of the Phonological Loop in Sentence Comprehension.*
41. Kim Sauter (2002). *Transfer and Access to Universal Grammar in Adult Second Language Acquisition.*
42. Laura Sabourin (2003). *Grammatical Gender and Second Language Processing: An ERP Study.*
43. Hein van Schie (2003). *Visual Semantics.*
44. Lilia Schürcks-Grozeva (2003). *Binding and Bulgarian.*
45. Stasinos Konstantopoulos (2003). *Using ILP to Learn Local Linguistic* Structures.
46. Wilbert Heeringa (2004). *Measuring Dialect Pronunciation Differences using Levenshtein Distance*.
47. Wouter Jansen (2004). *Laryngeal Contrast and Phonetic Voicing: A Laboratory Phonology*.
48. Judith Rispens (2004). *Syntactic and phonological processing in developmental dyslexia*.
49. Danielle Bougaïré (2004). *L'approche communicative des campagnes de sensibilisation en santé publique au Burkina Faso: Les cas de la planification familiale, du sida et de l'excision*.
50. Tanja Gaustad (2004). *Linguistic Knowledge and Word Sense Disambiguation.*
51. Susanne Schoof (2004). *An HPSG Account of Nonfinite Verbal Complements in Latin*.
52. M. Begoña Villada Moirón (2005). *Data-driven identification of fixed expressions and their modifiability*.
53. Robbert Prins (2005). *Finite-State Pre-Processing for Natural Language Analysis.*
54. Leonoor van der Beek (2005) *Topics in Corpus-Based Dutch Syntax.*
55. Keiko Yoshioka (2005). *Linguistic and gestural introduction and tracking of referents in L1 and L2 discourse.*

56. Sible Andringa (2005). *Form-focused instruction and the development of second language proficiency*.

57. Joanneke Prenger (2005). *Taal telt! Een onderzoek naar de rol van taalvaardigheid en tekstbegrip in het realistisch wiskundeonderwijs.*

58. Neslihan Kansu-Yetkiner (2006). *Blood, Shame and Fear: Self-Presentation Strategies of Turkish Women's Talk about their Health and Sexuality*.

59. Mónika Z. Zempléni (2006). *Functional imaging of the hemispheric contribution to language processing.*

60. Maartje Schreuder (2006). *Prosodic Processes in Language and Music.*

61. Hidetoshi Shiraishi (2006). *Topics in Nivkh Phonology.*

62. Tamás Biró (2006). *Finding the Right Words: Implementing Optimality Theory with Simulated Annealing.*

63. Dieuwke de Goede (2006). *Verbs in Spoken Sentence Processing: Unraveling the Activation Pattern of the Matrix Verb.*

64. Eleonora Rossi (2007). *Clitic production in Italian agrammatism.*

65. Holger Hopp (2007). *Ultimate Attainment at the Interfaces in Second Language Acquisition: Grammar and Processing.*

66. Gerlof Bouma (2008). *Starting a Sentence in Dutch: A corpus study of subject- and object-fronting.*

67. Julia Klitsch (2008). *Open your eyes and listen carefully. Auditory and audiovisual speech perception and the McGurk effect in Dutch speakers with and without aphasia.*

68. Janneke ter Beek (2008). *Restructuring and Infinitival Complements in Dutch.*

69. Jori Mur (2008). *Off-line Answer Extraction for Question Answering.*

70. Lonneke van der Plas (2008). *Automatic Lexico-Semantic Acquisition for Question Answering.*

71. Arjen Versloot (2008). *Mechanisms of Language Change: Vowel reduction in 15th century West Frisian.*

72. Ismail Fahmi (2009). *Automatic term and Relation Extraction for Medical Question Answering System.*

73. Tuba Yarbay Duman (2009). *Turkish Agrammatic Aphasia: Word Order, Time Reference and Case.*

74. Maria Trofimova (2009). *Case Assignment by Prepositions in Russian Aphasia.*

75. Rasmus Steinkrauss (2009). *Frequency and Function in WH Question Acquisition. A Usage-Based Case Study of German L1 Acquisition.*

76. Marjolein Deunk (2009). *Discourse Practices in Preschool. Young Children's Participation in Everyday Classroom Activities.*

77. Sake Jager (2009). *Towards ICT-Integrated Language Learning: Developing an Implementation Framework in terms of Pedagogy, Technology and Environment.*

78. Francisco Dellatorre Borges (2010). *Parse Selection with Support Vector Machines.*

79. Geoffrey Andogah (2010). *Geographically Constrained Information Retrieval.*

80. Jacqueline van Kruiningen (2010). *Onderwijsontwerp als conversatie. Probleemoplossing in interprofessioneel overleg.*

81. Robert G. Shackleton (2010). *Quantitative Assessment of English-American Speech Relationships*.

82. Tim Van de Cruys (2010). *Mining for Meaning: The Extraction of Lexico-semantic Knowledge from Text.*

83. Therese Leinonen (2010). *An Acoustic Analysis of Vowel Pronunciation in Swedish Dialects.*

84. Erik-Jan Smits (2010). *Acquiring Quantification. How Children Use Semantics and Pragmatics to Constrain Meaning*.

85. Tal Caspi (2010). *A Dynamic Perspective on Second Language Development.*

86. Teodora Mehotcheva (2010). *After the fiesta is over. Foreign language attrition of Spanish in Dutch and German Erasmus Student.*

87. Xiaoyan Xu (2010). *English language attrition and retention in Chinese and Dutch university students.*

88. Jelena Prokić (2010). *Families and Resemblances*.

89. Radek Šimík (2011). *Modal existential wh-constructions*.

90. Katrien Colman (2011). *Behavioral and neuroimaging studies on language processing in Dutch speakers with Parkinson's disease.*

91. Siti Mina Tamah (2011). *A Study on Student Interaction in the Implementation of the Jigsaw Technique in Language Teaching.*

92. Aletta Kwant (2011).*Geraakt door prentenboeken*. *Effecten van het gebruik van prentenboeken op de sociaal-emotionele ontwikkeling van kleuters.*

93. Marlies Kluck (2011). *Sentence amalgamation.*

94. Anja Schüppert (2011). *Origin of asymmetry: Mutual intelligibility of spoken Danish and Swedish.*

95. Peter Nabende (2011). *Applying Dynamic Bayesian Networks in Transliteration Detection and Generation.*

96. Barbara Plank (2011). *Domain Adaptation for Parsing.*

97. Cagri Coltekin (2011).*Catching Words in a Stream of Speech: Computational simulations of segmenting transcribed child-directed speech.*

98. Dörte Hessler (2011). *Audiovisual Processing in Aphasic and Non-Brain-Damaged Listeners: The Whole is More than the Sum of its Parts.*
99. Herman Heringa (2012). *Appositional constructions.*
100. Diana Dimitrova (2012). *Neural Correlates of Prosody and Information Structure.*
101. Harwintha Anjarningsih (2012). *Time Reference in Standard Indonesian Agrammatic Aphasia*.
102. Myrte Gosen (2012). *Tracing learning in interaction. An analysis of shared reading of picture books at kindergarten.*
103. Martijn Wieling (2012). *A Quantitative Approach to Social and Geographical Dialect Variation*.
104. Gisi Cannizzaro (2012). *Early word order and animacy*.
105. Kostadin Cholakov (2012). *Lexical Acquisition for Computational Grammars. A Unified Model.*
106. Karin Beijering (2012). *Expressions of epistemic modality in Mainland Scandinavian. A study into the lexicalization-grammaticalization-pragmaticalization interface.*
107. Veerle Baaijen (2012). *The development of understanding through writing.*
108. Jacolien van Rij (2012). *Pronoun processing: Computational, behavioral, and psychophysiological studies in children and adults.*
109. Ankelien Schippers (2012). *Variation and change in Germanic long-distance dependencies.*
110. Hanneke Loerts (2012).*Uncommon gender: Eyes and brains, native and second language learners, & grammatical gender*.
111. Marjoleine Sloos (2013). *Frequency and phonological grammar: An integrated approach. Evidence from German, Indonesian, and Japanese*.
112. Aysa Arylova. (2013) *Possession in the Russian clause. Towards dynamicity in syntax.*
113. Daniël de Kok (2013). *Reversible Stochastic Attribute-Value Grammars.*
114. Gideon Kotzé (2013). *Complementary approaches to tree alignment: Combining statistical and rule-based methods*.
115. Fridah Katushemererwe (2013). *Computational Morphology and Bantu Language Learning: an Implementation for Runyakitara*.
116. Ryan C. Taylor (2013). *Tracking Referents: Markedness, World Knowledge and Pronoun Resolution.*
117. Hana Smiskova-Gustafsson (2013). *Chunks in L2 Development: A Usage-based Perspective.*
118. Milada Walková (2013). *The aspectual function of particles in phrasal verbs.*
119. Tom O. Abuom (2013). *Verb and Word Order Deficits in Swahili-English bilingual agrammatic speakers.*

120. Gülsen Yılmaz (2013). *Bilingual Language Development among the First Generation Turkish Immigrants in the Netherlands.*

121. Trevor Benjamin (2013). *Signaling Trouble: On the linguistic design of other-initiation of repair in English conversation.*

122. Nguyen Hong Thi Phuong (2013). *A Dynamic Usage-based Approach to Second Language Teaching.*

123. Harm Brouwer (2014). *The Electrophysiology of Language Comprehension: A Neurocomputational Model.*

124. Kendall Decker (2014). *Orthography Development for Creole Languages.*

125. Laura S. Bos (2015). *The Brain, Verbs, and the Past: Neurolinguistic Studies on Time Reference.*

126. Rimke Groenewold (2015). *Direct and indirect speech in aphasia: Studies of spoken discourse production and comprehension.*

127. Huiping Chan (2015). *A Dynamic Approach to the Development of Lexicon and Syntax in a Second Language.*

128. James Griffiths (2015). *On appositives*.

129. Pavel Rudnev (2015). *Dependency and discourse-configurationality: A study of Avar.*

130. Kirsten Kolstrup (2015). *Opportunities to speak. A qualitative study of a second language in use.*

131. Güliz Güneş (2015). *Deriving Prosodic structures.*

132. Cornelia Lahmann (2015). *Beyond barriers. Complexity, accuracy, and fluency in long-term L2 speakers' speech.*

133. Sri Wachyunni (2015). *Scaffolding and Cooperative Learning: Effects on Reading Comprehension and Vocabulary Knowledge in English as a Foreign Language.*

134. Albert Walsweer (2015). *Ruimte voor leren. Een etnogafisch onderzoek naar het verloop van een interventie gericht op versterking van het taalgebruik in een knowledge building environment op kleine Friese basisscholen.*

135. Aleyda Lizeth Linares Calix (2015). *Raising Metacognitive Genre Awareness in L2 Academic Readers and Writers.*

136. Fathima Mufeeda Irshad (2015). *Second Language Development through the Lens of a Dynamic Usage-Based Approach.*

137. Oscar Strik (2015). *Modelling analogical change. A history of Swedish and Frisian verb inflection.*

138. He Sun (2015). *Predictors and stages of very young child EFL learners' English development in China.*

139 Marieke Haan (2015). *Mode Matters. Effects of survey modes on participation and answering behavior*.

140. Nienke Houtzager (2015). *Bilingual advantages in middle-aged and elderly populations.*

141. Noortje Joost Venhuizen (2015). *Projection in Discourse: A data-driven formal semantic analysis.*

142. Valerio Basile (2015). *From Logic to Language: Natural Language Generation from Logical Forms*.

143. Jinxing Yue (2016). *Tone-word Recognition in Mandarin Chinese: Influences of lexical-level representations.*

144. Seçkin Arslan (2016). *Neurolinguistic and Psycholinguistic Investigations on Evidentiality in Turkish.*

145. Rui Qin (2016). *Neurophysiological Studies of Reading Fluency. Towards Visual and Auditory Markers of Developmental Dyslexia.*

146. Kashmiri Stec (2016). *Visible Quotation: The Multimodal Expression of Viewpoint*.

147. Yinxing Jin (2016). *Foreign language classroom anxiety: A study of Chinese university students of Japanese and English over time.*

148. Joost Hurkmans (2016). *The Treatment of Apraxia of Speech. Speech and Music Therapy, an Innovative Joint Effort.*

149. Franziska Köder (2016). *Between direct and indirect speech: The acquisition of pronouns in reported speech.*

150. Femke Swarte (2016). *Predicting the mutual intelligibility of Germanic languages from linguistic and extra-linguistic factors.*

151. Sanne Kuijper (2016). *Communication abilities of children with ASD and ADHD. Production, comprehension, and cognitive mechanisms.*

152. Jelena Golubović (2016). *Mutual intelligibility in the Slavic language area*.

153. Nynke van der Schaaf (2016). *"Kijk eens wat ik kan!" Sociale praktijken in de interactie tussen kinderen van 4 tot 8 jaar in de buitenschoolse opvang.*

154. Simon Šuster (2016). *Empirical studies on word representations.*

155. Kilian Evang (2016). *Cross-lingual Semantic Parsing with Categorial Grammars.*

156. Miren Arantzeta Pérez (2017). *Sentence comprehension in monolingual and bilingual aphasia: Evidence from behavioral and eye-tracking methods*.

157. Sana-e-Zehra Haidry (2017). *Assessment of Dyslexia in the Urdu Language.*

158. Srđan Popov (2017). *Auditory and Visual ERP Correlates of Gender Agreement Processing in Dutch and Italian.*

159. Molood Sadat Safavi (2017). *The Competition of Memory and Expectation in Resolving Long-Distance Dependencies: Psycholinguistic Evidence from Persian Complex Predicates.*
160. Christopher Bergmann (2017). *Facets of native-likeness: First-language attrition among German emigrants to Anglophone North America.*
161. Stefanie Keulen (2017). *Foreign Accent Syndrome: A Neurolinguistic Analysis.*
162. Franz Manni (2017). *Linguistic Probes into Human History.*
163. Margreet Vogelzang (2017). *Reference and cognition: Experimental and computational cognitive modeling studies on reference processing in Dutch and Italian.*
164. Johannes Bjerva (2017). *One Model to Rule them all. Multitask and Multilingual Modelling for Lexical Analysis: Multitask and Multilingual Modelling for Lexical Analysis.*
165. Dieke Oele (2018). *Automated translation with interlingual word representations*.
166. Lucas Seuren (2018). *The interactional accomplishment of action.*
167. Elisabeth Borleffs (2018). *Cracking the code - Towards understanding, diagnosing and remediating dyslexia in Standard Indonesian.*
168. Mirjam Günther-van der Meij (2018). *The impact of degree of bilingualism on L3 development English language development in early and later bilinguals in the Frisian context*.
169. Ruth Koops van 't Jagt (2018). *Show, don't just tell: Photo stories to support people with limited health literacy*.
170. Bernat Bardagil-Mas (2018).*Case and agreement in Panará*.
171. Jessica Overweg (2018). *Taking an alternative perspective on language in autism*.
172. Lennie Donné (2018). *Convincing through conversation*: *Unraveling the role of interpersonal health communication in health campaign effectiveness.*
173. Toivo Glatz (2018). *Serious games as a level playing field for early literacy: A behavioural and neurophysiological evaluation.*
174. Ellie van Setten (2019). *Neurolinguistic Profiles of Advanced Readers with Developmental Dyslexia*.
175. Anna Pot (2019). *Aging in multilingual Netherlands: Effects on cognition, wellbeing and health*.
176. Audrey Rousse-Malpat (2019). *Effectiveness of explicit vs. implicit L2 instruction: a longitudinal classroom study on oral and written skills*.

177. Rob van der Goot (2019). *Normalization and Parsing Algorithms for Uncertain Input*.

178. Azadeh Elmianvari (2019). *Multilingualism, Facebook and the Iranian diaspora*.

179. Joëlle Ooms (2019). *"Don't make my mistake": Narrative fear appeals in health communication.*

180. Annerose Willemsen (2019). *The floor is yours: A conversation analytic study of teachers' conduct facilitating whole-class discussions around texts*.

181. Frans Hiddink (2019). *Early childhood problem-solving interaction: Young children's discourse during small-group work in primary school*.

182. Hessel Haagsma (2020). *A Bigger Fish to Fry: Scaling up the Automatic Understanding of Idiomatic Expressions*.

GRODIL
Center for Language and Cognition Groningen (CLCG)
P.O. Box 716
9700 AS Groningen
The Netherlands