Automatic Identification and Disambiguation of Verbal Multiword Expressions

A dissertation submitted to the
Faculty of Arts and Humanities at
Heinrich Heine University
for the degree of

Doctor of Philosophy

Author:

Rafael Ehren

Supervisors:

Laura Kallmeyer

Wiebke Petersen

Timm Lichte

Originally submitted June 5th, 2024
Revised version July 18th, 2025

D61

Rafael Ehren
Automatic Identification and Disambiguation of Verbal Multiword Expressions

Acknowledgements

It feels weird to put only one name on this thesis because scientific work seldomly is a purely individual effort and this one was neither. But at least I have this little paragraph to thank some people. First and foremoest, I want to thank my first superviser, Laura Kallmeyer, whose sharp mind is only outmatched by her patience and kindness. My second supervisor, Wiebke Petersen, I want to thank for giving me the tools I needed for this thesis. Without her seminar on machine learning I probably would not have been able to write my own architectures from scratch. Moreover, a big thanks goes to Timm Lichte whose passion for the field was so contagious that it made me follow this path in the first place. And also a huge thanks for being in the only-10-minutes-to-the-deadline trenches with me multiple times. I'm pretty sure I should thank a lot more people right now, but I still have some stuff to do, so I will thank them in person. But I'll finish by thanking my two MVPs (you know who you are) for letting me keep my cool over all this time by reminding me every day that nothing really matters in this world but you.

Düsseldorf, June 2024

Contents

Chapter 1

Introduction

Multiword expressions (MWEs) are sequences of words which exhibit some form of idiosyncratic behavior (Baldwin and Kim, 2010), the most prominent one being non-compositionality. This is why people who *kick the* infamous *bucket* usually have much severer problems than a swollen toe. Instead of adhering to the principle of compositionality, the combination of *kick*, *the* and *bucket* means 'die', a meaning which is not an amalgamation of the meanings of its parts. The same goes for the next two examples (MWEs in bold):

(1) What NBA players **secured the bag** and weren't the same after?[1]

(2) She **gave** him **a call**[2].

Example (1) is not about basketball players protecting their belongings but about them receiving a favorable contract (*bag = bag of money*) and in (2), there is no *call* changing possessions. Because of their idiosyncratic behavior, MWEs have long been one of the more challenging phenomena in natural language processing (NLP) and linguistics in general. Already in 1968, Chafe called idioms an "anomaly in the chomskyan paradigm" and 34 years later Sag et al. (2002) proclaimed MWEs to be a "pain in the neck for NLP". This "pain" stems, among other things, from the fact that a compositional treatment is not possible: For example, a machine translation system cannot translate (1) and (2) word for word if the target language does not have the same types of MWEs.[3] Likewise,

[1] Source: `https://www.reddit.com/r/nba/comments/awpjjc/what_nba_players_secured_the_bag_and_werent_the/` [Accessed: 15.05.2024]

[2] Whenever we do not give a source to an example, we constructed it ourselves.

[3] At the time of this writing, neither DeepL nor Google Translate nor ChatGPT 3.5 give the correct translation into German for Example (1). They all treat *secure the bag* compositionally. But to be fair: DeepL, for example, is capable of generating the correct

during semantic parsing, there is no one-to-one relation between words and concepts for *kick the bucket* as the concept needs to be assigned to the whole expression[4]. Example (2) displays non-compositionality to a lesser degree as *call* carries its regular meaning. The verb *give*, however, does not. It does not contribute much to the meaning of the whole which is indicated by the fact that *give a call* can be paraphrased by a single verb expressing the action denoted by the noun: She *called*. The verb *give* in (2) is only a semantically 'light' version of the full verb which denotes a change of possession, hence the name light-verb construction (LVC). In order to ensure the correct processing of these and other MWE types (cf. Section 2.4 and 2.5), we need to be able to automatically identify MWEs in running text. And for that, we need corpora annotated for MWEs so that systems capable of identifying them can be trained.

Luckily, a lot has happened in recent years in this regard. Not least thanks to PARSEME (PARSing and Multiword Expressions), a scientific network dedicated to MWEs (Savary et al., 2015), MWEs receive much more attention than they used to. One of the main contributions of the PARSEME network is the creation of highly multilingual, MWE-annotated corpora which served as the basis for three different shared tasks on the automatic identification of verbal MWEs (Savary et al., 2017; Ramisch et al., 2018, 2020). However, the results of the most recent shared task suggest that there is still some room for improvement as even the best performing, BERT-based system (Taslimipoor et al., 2020) struggled to generalize well over unseen data (38.53 Unseen MWE-based F1-score). As a matter of fact, compared to other NLP tasks, like POS tagging or dependency parsing, the numbers on the seen data are quite modest, too (70.14 Global MWE-based F1-score). But the fact that we now have these kinds of resources to train supervised systems is already a big step in the right direction.

However, there is one caveat with the PARSEME data sets. This is best illustrated by an example:

(3) The NBA **dropped the ball** not forcing the Pistons to play at least 1 game in front of no fans.[5]

translation for a lot of other idioms. But it seems to have some trouble with the more recently coined ones.

[4]The situation is a bit different for *secure the bag*. Since it is *decomposable*, we actually can assign concepts to all its components, even if those are different from the concepts the words would have if used literally. But we will discuss this in more detail in Section 2.2

[5]`https://www.reddit.com/r/nba/comments/p22cja/after_watching_untold_malice_at_the_palace_it_is/` [Accessed: 16.05.24]

(4) So the offensive player dropped the ball (fumble) and the defensive player knocked it through the back of the end zone, scoring a two point safety for the defensive team.[6]

In (3), we have an instance of the MWE *drop the ball* ('to make an error/miss an opportunity') and in (4), we have the same phrase but used literally. This is an issue, since a classifier not only needs to learn that *drop the ball* is a MWE, it also needs to learn when its not because these instances we need to treat compositionally. Contrast that with *give a call* which has no literal equivalent.[7] That means, for the former, we always need to consider the context to classify it correctly - it would not be enough to store it in a lexicon. We call these kinds of expressions potentially idiomatic expressions (PIEs) and we consider their disambiguation a subtask of MWE identification. Now, what is the issue with the PARSEME corpora? There are two, actually: (1) Literal instances of PIEs are not annotated, so we cannot evaluate how well a classifier handles disambiguation. (2) Even if those literal instances were annotated, there would not nearly be enough data to train a classifier. Savary et al. (2019) manually identified all literal PIE instances in five different data sets (Basque, German, Greek, Polish and Portuguese) of the PARSEME corpus 1.1 and found that the highest literality rate was at 4% (Portuguese). This study suggests that literal instances are – in the words of the authors – "rare birds", but we still think it is warranted to tackle the task of PIE disambiguation, since the resulting errors, as was shown above, can be substantial if we ignore the problem. This is why there is a special need for corpora containing PIE annotation and enough literal instances to successfully train a classifier capable of PIE disambiguation. Furthermore, we of course need architectures that are suitable to perform this task. In this thesis, we tackle both of those issues for German. Furthermore, we also address some aspects of MWE identification as a whole (see below). In doing so, we exclusively focus on verbal MWEs, since they constitute an especially challenging subclass of MWEs with respect to NLP.

[6] `https://www.reddit.com/r/NFLNoobs/comments/1803sbl/whenwhere_is_the_ball_down_exactly/` [Accessed: 16.05.24]

[7] With the exception of constructed examples of course. Also, accidental occurrences are possible like in *She gave him the call sheet.*

Chapter Guide

The primary topics of this thesis are MWE identification and its subtask PIE disambiguation, with the main focus lying on the latter. One of our main contributions is the creation of a German PIE corpus that can be used for the supervised training of classifiers capable of distinguishing between MWE instances and their literal counterparts. This corpus is then used for exactly that purpose in a variety of experiments. In these experiments, we test how well a model performs that is based on contextualizing a PIE's components before classification and whether we find clues that certain types of word embeddings capture morphosyntactic properties which could help during classification. Furthermore, we explore the use of an attention mechanism to uncover which parts of the input the system actually focuses on during classification and whether this corresponds to clues a human annotator would rely on. Finally, we will try to leverage the generating capabilities of a large language model to augment our PIE corpus with more data. Regarding MWE identification, we employ a BiLSTM coupled with a binary labeling scheme and a heuristic that converts them back to PARSEME-style labels. Then, we tackle the issue of overlapping MWE components by training individual classifiers for different MWE types.

Please note that chapters 5 and 6 contain heavily revised versions of previously published work by us. We provide the citation in the introduction of the respective sections but we will not mark passages taken verbatim in the interest of readability.

PART I - Background

Chapter 2 is concerned with defining what we actually mean when we talk about MWEs and their idiosyncratic behavior. Then, we present the different types of verbal MWEs according to the PARSEME typology and discuss how some of their properties might facilitate/exacerbate their automatic identification. In chapter 3, we elaborate on the two tasks that constitute MWE processing: MWE extraction and MWE identification. We provide an overview over existing approaches to both tasks, but with a heavy focus on the latter. Chapter 4 provides an overview over existing corpora that were annotated for verbal MWEs. Here, the main focus lies on corpora similar to the PIE corpus created by us in the course of this work.

PART II - Corpus Construction

In Chapter 5, we present our Corpus of Literal and Figurative Readings of Verbal Idioms (COLF-VID). We discuss how we extracted the VID types and their contexts from an existing corpus as well as the annotation guidelines. Then we will go deeper into the resulting data by investigating how it fits into assumptions made about the flexibility (or their lack thereof) of these expressions. Finally, we present an extension of COLF-VID that was used for the Shared Task on the Disambiguation of German Verbal Idioms we hosted in conjunction with KONVENS 2021.

PART III - Experiments

In Chapter 6, we will present the experiments conducted in the context of this work. It is divided into two sections: Section 6.1 is concerned with the identification of verbal MWEs (VMWEs) and Section 6.2 with PIE disambiguation. For VMWE identification, we employ two different classifiers, one is based on BiLSTMs and the other one is BERT-based. For the latter, we explore the effect of training one classifier per VMWE type instead of one classifier for all types. The goal is to find a way to address the issue of overlapping tokens where one token should receive multiple labels because they belong to different VMWE instances.

Regarding PIE disambiguation, we again employ a BiLSTM-based classifier and explore the use of different types of word embeddings as input. Then we describe the organization and results of the Shared Task on the Disambiguation of German Verbal Idioms we hosted in conjunction with KONVENS 2021. In Section 6.2.3, we extend our BiLSTM architecture with an attention mechanism in order to explore which part of the input the system focuses on most. Finally, we present a pre-study on data augmentation for PIE disambiguation where we try to leverage ChatGPT's capabilities for text generation.

PART IV - Conclusion

In the last chapter, we give an overview of the conclusions drawn from our work and discuss potential future work.

PART I

Background

Chapter 2

Multiword Expressions

The purpose of this chapter is to set the scene for the rest of the thesis by defining some of the crucial terminology. Most importantly, we need to clarify which linguistic entities we refer to when we talk about MWEs. What properties do they have that differentiate them from other sequences of words? What types of MWEs do exist and which of them are of interest to us? What properties do the individual types have that can facilitate/exacerbate their automatic identification? These and other questions will be addressed in the following.

2.1 Idiomaticity

MWEs consist of multiple words. So far, so unsurprising. Obviously, this criterion alone does not suffice for an expression to be classified as a MWE. After all, there must be a reason why we consider the initial verb phrase (VP) in (1) a MWE but not the one in (2).[1]

(1) She **spilled the beans** and immediately contacted her lawyer.

(2) She spilled the water and slipped on it.

Syntactically, both VPs are exactly the same. They both consist of the verb *spill*, followed by the determiner *the* and a direct object (*beans*/*water*). So why is *spill the beans* a MWE but not *spill the water*? And what does the former have in common with expressions like *give a talk* and *traffic light* which we also consider MWEs?

(3) She **gave an** inspiring **talk**!

[1] As we later discuss at length, the VP in (1) could have a literal reading and thus would not constitute a MWE. But this is far less likely in this context.

(4) Please stop at the next **traffic light**.

Besides the crossing of word boundaries, to be classified a MWE an expression needs to exhibit a certain idiosyncratic behavior which we usually term *idiomaticity*. It is a behavior that makes MWEs one of the more inconvenient linguistic phenomena. So inconvenient that Chafe (1968) declared idiomaticity an "anomaly in the chomskyan paradigm", a problem that resisted "the reiterated onslaught of the ablest members of the group within whose competence it f[ell]" (Kuhn (1962, p. 5) as cited in Chafe (1968, p. 109)) and thus required a paradigm shift. So what is idiomaticity exactly and why is it such a big issue that for Chafe it challenged some of the most basic postulates of the discipline at that time? Probably the most structured attempt at a definition and the one we will most heavily rely on from here on was formulated by Baldwin and Kim (2010). They list five different forms of idiomaticity:

- **Lexical**: An expression is lexically idiomatic if one or more of its constituents are not part of the conventional lexicon of a language (e.g. *ad hoc, faux pas*[2]).
- **Syntactic**: An expression is syntactically idiomatic if it violates the syntactic rules of a language (e.g. *kingdom come, to trip the light fantastic*).
- **Semantic**: Semantic idiomaticity denotes the property that the meanings of the components do not combine to form the meaning of the whole, i.e. semantically idiomatic expressions are considered non-compositional (e.g. *kick the bucket, shoot the breeze*).
- **Pragmatic**: An expression exhibits pragmatic idiomaticity if it is associated with a certain context (e.g. *good morning, break a leg*).
- **Statistical**: We speak of statistical idiomaticity when words occur more often together than one would assume when considering the frequency of the individual words and alternate phrasings of the same expression (e.g. *black and white television vs. ?white and black television*). Statistical idiomaticity is closely related to the notion of institutionalization.

Now, going back to Example (1), the reason why we consider *spill the beans* a MWE is that at some point in time this combination of words

[2] *Faux* probably can be considered a part of the English lexicon by now (e.g. *faux fur/remark/...*) but not *pas*.

received the meaning 'reveal a secret' which we cannot derive by virtue of combining the individual literal meanings. As we learned just now, we term this kind of non-compositionality *semantic idiomaticity* and it is the property most closely associated with MWEs. It is also an inevitable result of *lexical* and *syntactic idiomaticity*, since compositionality, as we define it, presupposes the use of conventional vocabulary and the adherence to syntactic rules (Baldwin and Kim, 2010). E.g. *ad hoc* cannot be compositional as neither *ad* nor *hoc* are individual words of the English vocabulary. Analogously for syntax, English does not have syntactic rules that allow for the coordination of a preposition and an adjective to result in an adverb (*by and large*). This also shows that the different types of idiomaticity are not mutually exclusive but can occur at the same time. For example, an expression like *trip the light fantastic* ('dance nimbly') is syntactically, semantically and statistically idiomatic.

Despite a strong association between non-compositionality and MWEs it is not a necessary property for MWEhood[3]. After all, the collocation *traffic light* in Example (4) is perfectly compositional. It is only *statistically idiomatic* as the two words occur together with marked frequency. What makes these kind of expressions idiosyncratic is the existence of anti-collocations like *traffic director* (Sag et al., 2002), i.e. the institutionalization seems to 'block' alternatives which should be acceptable when only considering compositionality. It should be noted, however, that not everyone includes collocations into the class of MWEs. E.g. PARSEME, an interdisciplinary network of scientists concerned with MWEs and Parsing (Savary et al., 2015), does not consider collocations MWEs[4].

We now have seen both ends of the compositionality spectrum for MWEs. Example (3) shows an in between-case: The light verb construction *give a talk* is partly compositional with the noun *talk* contributing its literal sense, while the verb *give* is only a bleached version of the full verb, i.e. there is no *talk* that changes possessions and the verb does not contribute much else to the meaning of the whole expression. What exactly is added by the verb will be discussed in Section 2.5.4.

Naturally, this varying degree of compositionality comes with some considerable challenges for NLP (cf. Sag et al. (2002)). In the following section, we will see that it does not end there as non-compositional MWEs can be divided further into two subclasses.

[3]We use this neologism analogously to the term *verbhood*: It denotes the quality or state of being a MWE.

[4]The PARSEME typology of verbal MWEs as well as their corpora will be discussed at length later on.

2.2 Decomposability

In the preceding section, we discussed the non-compositionality of semantically idiomatic expressions. There are more dimensions to this, however. Nunberg et al. (1994) introduced the distinction between idiomatically combining expressions (ICEs) and idiomatic phrases (IP) that is widely used today when discussing idioms. Both ICEs and IPs are non-compositional in the sense that the literal meanings of their components do not combine to form the meaning of the whole. The difference is that for ICEs we can establish a mapping from their components to their idiomatic meanings which in turn receive a compositional treatment. To illustrate this, consider the following examples:

(5) I wish he would just **spill the beans**.

(6) I did not get any sleep, because he was **sawing wood** all night.

In Example (5), each element of *spill the beans* refers to a part of the idiomatic meaning 'reveal a secret': *spill* corresponds to 'reveal' and *beans* to 'secret'. Once we established this correspondence, we can analyze it compositionally. This is not possible for (6), however, as there is no clear mapping from the components of *saw logs* to parts of its idiomatic meaning ('snoring/sleeping') because we cannot decompose it into multiple constituents.

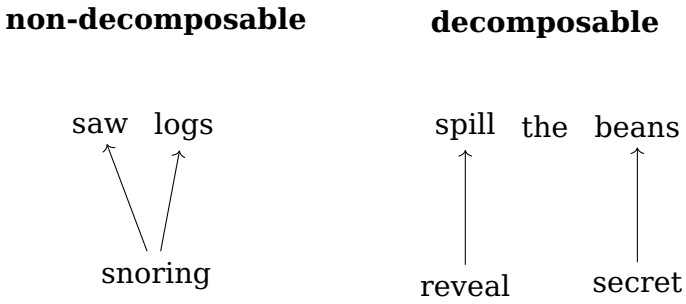


Figure 2.1: Decomposability of MWEs.

By now, the terminology has been reformulated to account for the fact that this kind of analysis starts with the idiomatic meaning and tries to relate it back to the idiom's components (Sag et al., 2002), i.e. we first have to know the meaning of the expression to be able to establish the mapping. The direction of the arrows in Figure 2.1 is supposed to emphasize the direction of the analysis. Accordingly, we usually speak of *decomposable* (ICEs) and *non-decomposable* (IPs) idioms.

Whether an expression is decomposable or non-decomposable is not only a matter of semantics, but the two classes supposedly have diverging properties regarding their syntactic flexibility which we will have a closer look at in the following section.

2.3 Flexibility

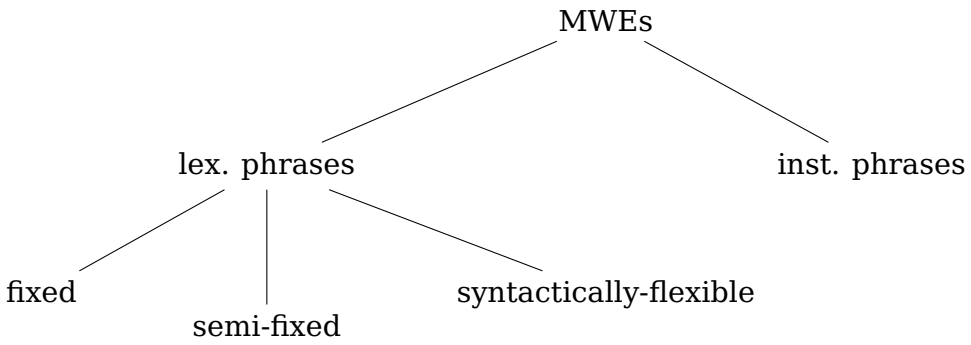


Figure 2.2: Fixedness of MWEs (Sag et al., 2002).

A common MWE classification scheme pertains to the syntactic fixedness (or flexibility, depending on the perspective) of MWEs. Sag et al. (2002) classify MWEs into *lexicalized* and *institutionalized* phrases (cf. Figure 2.2) with the former being in turn subdivided into *fixed*, *semi-fixed* and *syntactically-flexible* expressions. Institutionalized phrases are those that only exhibit statistical idiomaticity as *traffic light* in Example (4). Lexicalized expressions, on the other hand, are semantically idiomatic and their (non-)compositionality is regarded as highly correlated to the degree of syntactic flexibility.

Fixed expressions are those not undergoing any morphosyntactic variation or internal modification at all like *ad hoc* or *kingdom come*. These we can just store in a lexicon without loosing any generalization capabilities.

Semi-fixed expressions at least allow for some degree of variation, for example inflection of the verb:

(7) He **kicked the bucket**.

Other operations like internal modification or passivization are supposedly not allowed:

(8) *He **kicked the** *final* **bucket**.

(9) ***The bucked** was **kicked** by him.

Usually, non-decomposable idioms like *kick the bucket* are classified under semi-fixed expressions with their non-decomposability cited as the reason for their syntactic inflexibility. The reasoning behind this is that the components of the idiom do not have individual referents and thus cannot be passivized or modified individually. This view, however, has been subjected to some scrutiny in recent years with many doubting the perceived fixedness of these kind of expressions. And indeed, it is not difficult to find examples for the modification of idiom-internal NPs without having to rely on metalinguistic markers like *proverbial*, *metaphorical* or similar[5], which, some argue, do not count (cf. Kay et al. (2015)):

(10) With that dumb remark at the party last night, I really **kicked the** *social* **bucket**.

This example by Ernst (1981) shows the phenomenon of *external modification* where the "meaning of the modifier applies to the idiomatic meaning of the idiom as a whole and functions like a domain adverb" (Bargmann et al., 2021, p. 249). So, an alternate phrasing of Example (10) might resemble the following:

(11) Socially, I really **kicked the bucket** with that dumb remark at the party last night.

Hence, the fact that *bucket* has no referent does not prevent it from being syntactically modified. And even passivization is attested for *kick the bucket* (Fellbaum, 2019):

(12) And no one here knows when the bell will toll or when **the bucket** will be **kicked**.

Examples like these contradict the claim that certain syntactic operations are not possible for non-decomposable idioms, at least for English. However, from the perspective of supervised machine learning, it is not so relevant whether this claim is *categorically* true, but whether it is true most of the times. The models we use in the course of this work are probabilistic in nature and should be able to learn from strong tendencies, even if there are a few counter-examples to be found. We will revisit this discussion in Chapter 5 when analyzing the German data we collected and annotated to create a German corpus of verbal idioms and their literal counterparts.

[5] As in: He kicked the *proverbial* bucket.

The class of syntactically flexible expressions encompasses MWEs that show the largest amount of flexibility. Decomposable idioms fall into this category:

(13) In spite of its conservatism, many people were eager to **jump on the** *horse-drawn Reagan* **bandwagon**.

In this example by (Bargmann et al., 2021), the idiomatic sense of bandwagon ('movement') is modified by *Reagan* and *horse-drawn*: It is a movement which is tied to/led by the former US president Ronald Reagan and it is depicted as old-fashioned. In contrast to non-decomposable expressions, this kind of syntactic flexibility is expected because the components actually refer to individual entities and therefore we should be able to also modify them semantically. Accordingly, this is termed *internal modification*.

Extraction and passivization are likewise possible:

(14) **The beans**, **the beans**! **Spill them beans** and tell me why I dragged myself down to the hellhole they call Barnacle Bluffs. And don't leave a thing out.[6]

(15) When the **beans were spilled**, producers chalked the pay gap up to Matt Smith having a bigger draw than Foy due to his run on "Doctor Who".[7]

Other flexible MWE types are light-verb constructions (e.g. *take a bath*) and verb-particle constructions (*look up*) which we will discuss in detail in Section 2.5. But first, we will shortly address the types of MWEs which are **not** of interest to us.

2.4 Non-verbal Types

This work is exclusively concerned with verbal MWEs (VMWEs). Nevertheless, we will briefly address other classes of MWEs in order to delineate verbal and non-verbal types.

2.4.1 Nominal MWEs

Nominal MWEs are those MWEs whose head is a noun. A very common nominal MWE type is the noun compound (NC) which consists of two or

[6] Source: Carter (2022)

[7] Source: `https://theplaylist.net/crown-producers-apologize-20180320/`, [Accessed: 10.05.2024]

more nouns:

(16) *car park, bus stop, consumer confidence survey*

In English, nominal MWEs are usually right-headed as in (16), where *park*, *stop* and *survey* function as the respective heads. The modifiers of nominal MWEs are not restricted to nouns, however, as some nouns are coupled with adjectives and verbs:

(17) *black board, dutch uncle*[8]*, législation européenne*

(18) *swimming pool, washing machine, connecting flight*

As the French expression *législation européenne* in (17) shows, nominal MWEs can also be left-headed. Concerning nominal MWEs with verbal modifiers, like in (18), it is important to note that we do not include them when considering verbal MWEs, since we understand VMWEs as MWEs with a verbal head. The same goes for expressions with a deverbal head:

(19) *train spotting, task assignment, crop destruction*

The semantic relations that hold between the components of compounds are intensely studied because detecting and classifying them is far from trivial due to their implicit nature. E.g. it is not in any way externalized why *carrot cake* is a cake made *from* carrots, while a *swimsuit* is a suit made *for* swimming (Baldwin and Kim, 2010). Furthermore, nominal compounds can exhibit syntactic ambiguity if they consist of more then two components like in Example (16). To correctly interpret *consumer confidence survey*, we have to know that *consumer* modifies *confidence* and these two together in turn modify *survey*. The (wrong) alternative would be that *confidence* modifies *survey* and these two are modified by *consumer* (Girju et al., 2005):

(20) (consumer confidence) survey vs. *consumer (confidence survey)

2.4.2 Prepositional MWEs

Prepositional MWEs are another common type which can be subdivided into determinerless prepositional phrases (PP-Ds) and complex prepositions.

[8] "Dutch uncle is an informal term for a person who issues frank, harsh or severe comments and criticism to educate, encourage or admonish someone." Source: `https://en.wikipedia.org/wiki/Dutch_uncle` [Accessed: 20.01.2023]

PP-Ds are defined as combinations of a preposition and a singular noun without a determiner:

(21) *in school, in isolation, in winter, by bus, on film, at hand, on ice*[9]

The degree of the resulting syntactic markedness depends on the nouns behaviour outside the PP: If it has a tendency to appear without an article, the PP itself is less marked. That means PPs with uncountable nouns (e.g. *in isolation*) are less marked than PPs with countable nouns (e.g. *by bus*). In many cases, PP-Ds also exhibit semantic markedness when the semantics of the noun is different from the simplex noun (Baldwin et al., 2006). As with syntactic markedness, this is a matter of degree. The examples *in school* and *on ice* illustrate both ends of the spectrum.

Complex prepositions are expressions of the form *in spite of* or *in addition to*. They are either completely fixed or they allow for internal modification or determiner insertion (Baldwin and Kim, 2010).

2.5 Verbal Types

Above we mainly discussed general properties of MWEs: namely that they cross word boundaries and exhibit some (or multiple) forms of idiomaticity. Now, we simply define verbal MWEs (VMWEs) as those MWEs whose syntactic head is a verb. As we will see, it is probably the class most challenging for NLP. In this section, we will heavily rely on the PARSEME typology[10], which encompasses six different classes of VMWEs. We will explore how these types are discussed in the literature and how the PARSEME definitions differ in some regards. Furthermore, we will discuss how their respective properties could potentially facilitate or exacerbate their identification in an automated setting.

Please note that we do not cover one of the categories in PARSEME, the inherently adpositional verbs (IAVs), as they only have the status of a "special optional and experimental category" (Khelil et al., 2022)[11]. Thus, they may not remain part of the typology in future releases of the PARSEME corpus. We will only shortly discuss them in Section 4.1 when we present the PARSEME corpora.

[9] As in *put something on ice*.

[10] `https://parsemefr.lis-lab.fr/parseme-st-guidelines/1.2/`, [Accessed: 22.03.2024]

[11] We do not provide page numbers with verbatim quotes from the PARSEME annotation guidelines because they are only available online and accordingly there are no page numbers.

2.5.1 Verb-particle Constructions

Verb-Particle Constructions (VPCs) (also called phrasal verbs) consist of a verb and a particle "which is typically homonymous with an adverb or a preposition" (Thim, 2012, p. 10):

(22) We should **take** the food **in**.

(23) You should **eat up** your lunch, otherwise it will rain tomorrow.

(24) They gonna **blow up** the old shopping mall in the afternoon.

To distinguish particles from non-particles we can leverage their respective language-specific syntactic properties. To illustrate this, consider the VPC *look up* which is homonymous to a prepositional verb:

(25) a. I **looked up** the word.
b. I **looked** the word **up**.
c. *It was **up** the word that I **looked**.

(26) a. I **looked up** the road.
b. *I **looked** the road **up**.
c. It was **up** the road that I **looked**.

As (25-a) and (25-b) show, respectively, the object of the VPC *look up* may occur either after or before the particle. The object of prepositional *up*, however, cannot precede it (cf. (26-b)). Another difference is that the VPC's object and its particle cannot be fronted together (cf. (25-c)), while the preposition and its object can (cf. (26-c)) (Dehé, 2015). This allows us to conclude that the two instances of *look up* in (25) and (26) are involved in two different constructions.

As for every verbal construction type that follows, we have to answer the question why (or rather, under what circumstances) we consider VPCs a VMWE type. In this case, the answer lies in the varying degree of compositionality between the verb and the particle. Examples (22) to (24) illustrate this. (22) is completely compositional, since both *take* and *in* contribute their literal meaning. It is an example of a well-known and very productive class of VPCs where a verb combines with a directional particle. This directional particle can usually be replaced by a PP (Thim, 2012):

(27) We should **take** the food **in[to the house]**.

Example (24), on the other hand, is completely idiomatic[12] (i.e. non-

[12]When we talk about idiomaticity without a specifier, then we mean semantic id-

compositional) as the meaning change of *blow* cannot be attributed to the meaning of *up* alone. Consequently, this class is much less productive albeit the number of idiomatic VPCs being quite large. Example (23) could be seen as partly compositional: *eat* has its literal meaning, but *up* does not. It does not indicate a direction but an aspectual meaning (completion like in *drink up*, *dry up*, *clean up*, etc.). These types of VPCs again form a very productive class of VPCs. Figure 2.3 shows an overview of this common classification scheme:

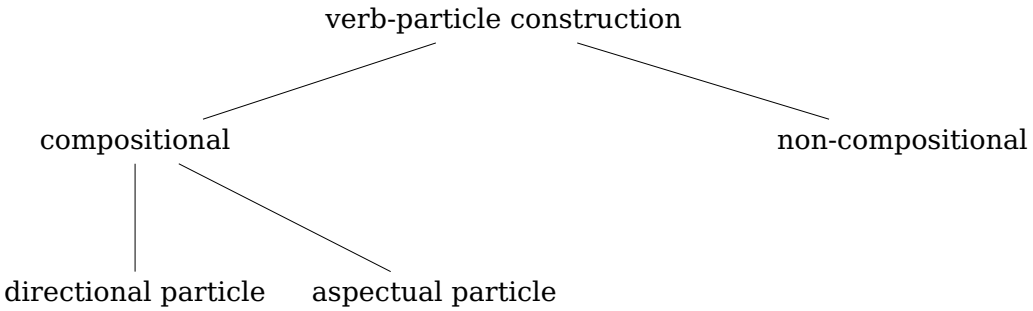


Figure 2.3: Semantic classification of VPCs acc. to Thim (2012).

Now, which of these VPC types do we classify as VMWEs? Non-compositional VPCs have the answer in their name, so that is the easy case. The same goes for directional VPCs which are classified under *compositional* and according to our understanding should not count as VMWEs. And indeed, their productivity, illustrated by the following example by Thim (2012), suggests this as well:

(28) George { tossed / took / put / carried / threw } the food { up. / in. / away. / back. / out. }

Thim (2012, pp. 14-15) states:

> In such syntagms, the paradigmatic insertion of any verb and any particle seems possible, as long as the combination of verb and particle allows an interpretation of motion through space, with the particle expressing the direction and the verb expressing the kind of the verbal action.

iomaticity. This is in line with much of the literature where the term is often used in this way.

Hence, this kind of regularity hardly seems to justify a treatment as VMWEs.

That leaves aspectual VPCs, which are a trickier case. The attentive reader will not have failed to notice that aspectual particles are subsumed under *compositional* in Figure 2.3 while above we claim they do not contribute their literal meaning. It turns out, this is a matter of debate: "[...] this aspectual *up* should be listed as an independent lexical item, free to combine with verbs that meet its selectional restrictions." (Jackendoff, 2002, p. 76). Given the high degree of productivity, this viewpoint does not seem unjustified as an example by Jackendoff (2002) illustrates:

(29) drink/eat/finish/pack/close/clean/gobble/... up

Nevertheless, PARSEME does not consider aspectual VPCs as compositional in their annotation guidelines, but as semi-non-compositional and thus treats them as VMWEs. We do not aim to settle the debate at this point, but it is worth to bear in mind that the question for MWEhood is not so easy to answer for some cases.

VPCs are quite challenging when it comes to the identification task. As we have seen, particles have to be distinguished from prepositions or adverbs by virtue of their different syntactic properties. Another challenge is orthography because in some languages, such as German, verb and particle can be written as one word:

(30) Er **fuhr** den Baum **um**.
He drove the tree Part
'He drove against the tree and it toppled.'

(31) Er hat den Baum **umgefahren**.
He has the tree Part+driven.
'He drove against the tree and it toppled.'

On top of that, a lot of VPCs are subject to ambiguity, i.e. they have literal counterparts:

(32) Could you **put up** a friend of mine when he's in town?

(33) Could you put up a note so they know we're not coming?

Example (32) contains an instance of non-compositional *put up* which means 'offer accommodation', while Example (33) contains an instance of its literal counterpart.

Last but not least, the verb and the particle can be discontiguous (cf. (22),(25-b) and (30)) and even the order of the particle and verb can vary

(cf. (31)).

But there is also at least one property that potentially can help during identification: While the set of verbs a VPC can draw from is unrestricted, the set of particles is rather small. Hence, in a supervised context, a classifier should be able to pick up a large ratio of eligible particles during training.

2.5.2 Inherently Reflexive Verbs

Following Geniusiené (2011), we understand reflexive verbs as verbs with a reflexive pronoun. If the pronoun expresses semantic reflexivity, which does not necessarily have to be the case, as we will later see, the agent and patient of a sentence are the same:

(34) John hit himself.

Example (34) shows an instance of a verb where we usually would expect the agent and patient refer to different entities, i.e. people tend to hit other people and not themselves. We follow Schäfer (2012) in calling these *naturally disjoint verbs* (NDVs). Contrary to this, *naturally reflexive verbs* (NRVs) carry "inherent in their meaning [...] the lack of expectation that the two semantic roles they make reference to will refer to distinct entities" (Kemmer (1993, p. 58), as cited in Schäfer (2012)). This is typical for so-called *grooming* verbs that denote actions of bodily care one usually performs unto oneself:

(35) John shaved himself.

But despite their preference for coreferentiality, NRVs can be used non-reflexively:

(36) The barber shaved John.

Contrast that with the verb in (37):

(37) The criminal **perjured himself**/*John.

These types of verbs we call *inherently reflexive verbs* (IRVs), since the reflexive pronoun is mandatory and cannot be replaced by a non-reflexive NP. In addition, in some languages, such as Dutch, IRVs only allow for the light reflexive pronoun, while for NDVs the heavy[13] version is strongly preferred (Alexiadou and Schäfer, 2014):

[13]The distinction of light vs. heavy reflexive pronouns is based on phonological weight.

(38) Jan **schaamt zich**/*zichzelf/*Marie
John shames REFL/REFL.SELF/Mary
'John is ashamed'

Figure 2.4 gives an overview over the three different kinds of reflexive verbs. Of these three, we consider IRVs VMWEs because – as the name already suggests – the verbs never occurs without their reflexive clitic. And if they do, the verbs have very different meanings from the IRV version.

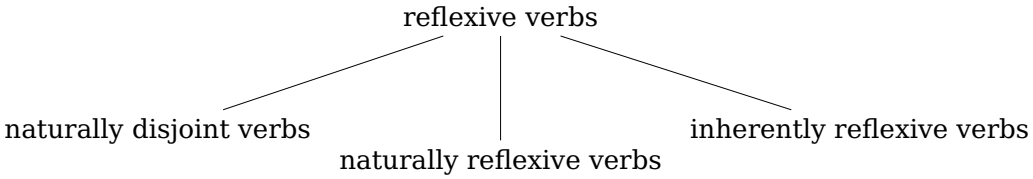


Figure 2.4: Types of reflexive verbs.

Regarding challenges during identification we have already seen how we have to distinguish between IRVs on one side and NRVs and NDVs on the other:

(39) John shaves/hits/**perjures himself**.

In (39) *perjure himself* should be tagged as a VMWE, while the NRV *shave* and the NDV *hit* should not.

But the challenges do not end there. Reflexive pronouns "can express an extremely broad range of meanings, including semantic reflexivity (coreference of Agent and Patient), reciprocity, possessivity, anticausativity, modality, etc. Moreover, in many languages the [reflexive pronoun] is used as a passive and impersonal marker." (Geniusiené, 2011, p. 1). That means, we do not only have to distinguish IRVs from NRVs and NDVs, but also from non-reflexive usages. Consider for example reciprocity:

(40) They hugged each other.

While English has a reciprocal anaphor (*each other*) that is different from the reflexive pronoun (*themselves*), German uses *sich* to express both reflexivity (cf. (41)) as well as reciprocity (cf. (42)), which makes it harder to discriminate both phenomena:

(41) Sie **schämten sich**.
They shamed REFL.
'They were ashamed'.

(42) Sie umarmten sich.
They hugged each other.

Another challenge, which occurs in some languages, is the concatenation of the verb and the pronoun, resulting in a single token[14] (much like VPCs, cf. Example (31)):

(43) Debería **avergonzarse**, Primer Ministro, por haber hecho eso.
Be ashamed+RCLI Prime Minister, for have done this.
'Be ashamed, Prime Minister, for having done this.'

With regard to morphosyntactic features a classifier could potentially leverage, we have seen that for some languages at least the pronoun (heavy vs. light) could serve as clue whether a certain candidate is inherently reflexive or not (cf. Example (38)).

2.5.3 Multi-verb Constructions

In using the term multi-verb construction (MVC), we follow the PARSEME terminology. It has to be noted, however, that the literature on the topic seems to lack a consistent definition of the term. In some cases, it seems to be used synonymously to serial verb construction (SVC) – which in itself is a hotly debated term (cf. Haspelmath (2016)) – and sometimes it seems to denote a hypernym of SVCs. To exacerbate the confusion even further, in the PARSEME annotation guidelines, as far as we understand it, it denotes a subclass of SVCs. We as well consider MVCs to be a type of SVC and thus will go into more detail what constitutes a SVC and which SVC types can be considered VMWEs, i.e. MVCs.[15] Then we will see how it aligns with the PARSEME annotation guidelines.

SVCs are combinations of two or more independent verbs in the same clause without any element linking them for coordination or subordination and without them being in a predicate-argument relationship (Lovestrand, 2021; Aikhenvald, 2018; Haspelmath, 2016) as shown in the exam-

[14] Source for Example (43): `https://www.europarl.europa.eu/doceo/document/CRE-6-2009-03-24_ES.html?redirect` [Accessed: 23.05.2024]

[15] Haspelmath (2016, p. 297) does not consider MVCs to be a type of SVC, but only closely related, because they are not "pure instances of a regular schematic construction", meaning idiomatic expressions cannot be of the same type as the regular pattern, even if they are syntactically equivalent. In the same vein, he describes *kick the bucket* as only closely related to the transitive construction. But consequently this would also mean that *look up* in *She looked up the tower* would be a phrasal verb, but not in *She looked looked up the name* (since it is an idiomatic VPC). At the end, it is a question whether we define these phenomena syntactically or semantically. We lean towards the former.

ple from Domine (2019):

(44) joshi-ha gohan-o **tabe-nokoshi-ta.**
girl-TOP rice-ACC eat-leave-PST.
'The girl left rice after eating some.'

The two verbs in (44) (*eat* and *leave*) appear in the same clause and there is no linking marker as in the converbal *-te* construction, which is another type of complex predicate in Japanese. Furthermore, there is no predicate-argument relationship between the two verbs: Both share the same agent and patient and are independent from one another.

There have been further defining criteria proposed – e.g. SVCs should behave as a single predicate or they should depict a single event – but these are not undisputed. For example, Haspelmath (2016) rightfully points out that whether something is regarded as one or multiple events is highly subjective and therefore not suitable to be part of a definition. As a matter of fact, even the criteria mentioned before are heavily scrutinized: It is not always clear what constitutes a verb or a clause and it has been proposed to weaken the no-marker constraint (Lovestrand, 2021). But it would be beyond the scope of this work to go deeper into this discussion and since these criteria are the ones most commonly cited, we will also rely on them.

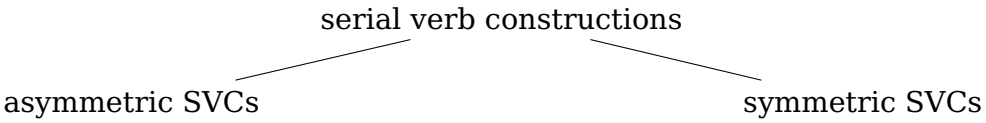


Figure 2.5: Types of SVCs.

In the SVC literature, there is made a distinction between *asymmetric* and *symmetric* SVCs (cf. Figure 2.5). Asymmetric SVCs consist of a verb from a relatively unrestricted class, the *major verb*, and a verb from a "semantically or grammatically restricted (or closed) class" (Aikhenvald, 2018, p. 56), the *minor verb*. The major verb can be seen as the semantic head of the construction as its meaning constitutes the lion's share of the semantics of the whole construction (hence the name). Accordingly, the transitivity value of the whole also depends on the major verb. The minor verb specifies the major verb by contributing something like direction or motion and often undergoes grammaticalization. E.g. Hopper and Traugott (2003) describe a couple of asymmetric SVCs, where the verb *take* evolved into a case marker for direct objects in the respective languages:

(45) Zuì **bǎ** zhū-gēn-zǐ xì **kàn**.
drunk *ba* dogwood-tree careful look.

In Example (45), *ba* can have the meaning *take*, resulting in the reading *While drunk, I took the dogwood tree and carefully looked at it*, or it could function as an accusative case marker, resulting in *I carefully looked at the dogwood-tree*.[16]

One of the most common types of asymmetric SVCs are motion verbs where the verb from the open class expresses a manner of motion and the verb from the closed class a direction.

Symmetric SVCs on the other hand draw their constituents from only unrestricted classes, i.e. all verbs have the same status and there is not one which serves as head of the whole construction. The order of these verbs tends to reflect the temporal order of the actions they describe. According to Aikhenvald (2018, p. 56), symmetric SVCs "often get lexicalized, and then become idiom-like and non-compositional in their meanings":

(46) du-wheta du-matsiketa
3sgf-stay+CAUS 3sgf-be.bad+CAUS
she makes (it) stay she makes it bad
'She prepares fermented manioc beer'

Naturally, these kind of expressions are of interest to us as they pass the requirement for MWEhood. The definition of MVCs in the PARSEME annotation guidelines, however, only seems to describe a certain kind of asymmetric SVCs. According to these guidelines, MVCs "are VMWEs composed by a sequence of two adjacent verbs" that "are unaccompanied by any explicit coordination, subordination, or dependency marker" and consist of a "V-gov (vector) verb [which] is semantically delexicalized and [a] V-dep (polar) verb [that] contains the core meaning of the whole"[17] (Khelil et al., 2022). This pretty clearly fits the description of asymmetric SVCs, although not explicitly stated.[18] SVCs are only mentioned in some of language-specific tests to demarcate them from MVCs.

This is not to say the PARSEME definition of MVCs does not make sense. On the contrary: The non-compositionality of asymmetric SVCs due to the minor verb's tendency to be 'bleached' to the point of gram-

[16] Examples like this, however, beg the question, whether they can still be considered SVCs at all (Enfield, 2009).

[17] In addition, the definition contains a few of the more disputed criteria like the same-event criterion discussed above.

[18] The only difference is that the delexicalized verb (V-gov) which is equivalent to the minor verb is seen as the governing one.

maticalization (Aikhenvald, 2018) is obviously a good reason to classify them as VMWEs. All the more so when we consider the strong resemblance to light-verb constructions (cf. Section 2.5.4). But it is a bit surprising that non-compositional symmetric SVCs as in Example (46) are not included in the definition. A possible explanation is that they are included in the class of verbal idioms instead. As we will see in Section 2.5.5, it is quite challenging to distinguish them from other types, so the guidelines contain a section about possible dependents of the VMWE head and the resulting annotation choices. They say the following about verbal dependents: "Verb with no lexicalized dependent: fine-grained tests need to be applied in order to discriminate between a MVC and a VID" (Khelil et al., 2022). The problem is that it is unclear what would be the head and what the dependent in a symmetric SVC. Another issue is the insistence that components of MVCs need to be "adjacent" which clearly is not the case for asymmetric SVCs as many examples from the literature attest.

All things considered, the PARSEME definition of MVCs could benefit from some terminological clarification, as it clearly draws on the definition of SVCs, but fails to address all the relevant phenomena and how MVCs are supposed to relate to SVCs overall.

2.5.4 Light-verb Constructions

Light-verb constructions (LVCs) are combinations of a verb and a complement in which the verb is semantically *bleached*, i.e. the semantic content of the verb is significantly reduced compared to its usage as a full verb (Wiese, 2006). This leaves the lion's share of semantic contribution to the complement which is illustrated by the fact that very often a LVC can be paraphrased by a simple verb denoting the action implied by the complement:

(47) She **takes a plunge**.

(48) She plunges.

In example (47), there is no *plunge* that changes possession as would be the case if *take* were used in one of its 'heavy' forms, but someone performs the action of *plunging* (somewhere). Consequently, little is lost when we compare (47) to (48). So the question is: What is it exactly that light verbs contribute to the predication?

In the past, some approaches treated light verbs merely as verbal licensers for nouns that do not provide any semantic content, but only functional properties like tense, aspect or agreement (Butt and Geuder,

2001). And indeed, as Butt (2010) notes, light verbs are a very productive device for incorporating loan words into a language by acting as verbalizers. For example, Japanese uses the *suru*-construction ('to do') to this effect:

(49) John-ha **arubaito**-o **shita**.
John-Top work-Acc do-Past.
'John worked part-time.'

Example (49) contains the loan word *arubaito* which means 'part-time job'[19] and the combination with *suru* allows for it to act as a verb.

Still, *light* does not mean the verb has to be completely void of semantic content as an example from (Fleischhauer and Gamerschlag, 2019) illustrates:

(50) Der Verbrecher **steht unter Beobachtung** (durch die Polizei).
'The criminal is under surveillance (by the police).'

(51) Die Polizei **stellt** den Verbrecher **unter Beobachtung**.
'The police places the criminal under surveillance.'

If the respective light verbs were completely empty with regard to semantics, both LVCs would be the same since the only difference between them is the light verb. But the two examples differ insofar as 'stellen' contributes a causative meaning to (50) with an additional argument for the causer ('*Die Polizei*').

Butt (2010, pp. 15-16) goes even further:

> [...] [T]he semantic contribution goes beyond that of the purely functional tense/aspect kind. While light verbs generally do signal some kind of boundedness or telicity or causation (crosslinguistically), they also go beyond that and signal volitionality, benefaction, forcefulness, surprise, etc. The degree to which they signal this differs from language to language, but this component always seems to be present [...].

Furthermore, diachronic evidence for a contribution beyond grammatical functionality is given by Butt and Lahiri (2013) who argue against an analysis of light verbs analogous to auxiliaries. According to their diachronic data (unlike auxiliaries) light verbs are always form identical to their full verb counterpart which in turn suggests that light and full usage are drawn from the same lexical entry "whose lexical information

[19] The loan word *arubaito* is used for part-time (especially student) jobs which is curious given the fact that the German word *Arbeit* does not entail this meaning at all.

plays out in different ways depending on its syntactic environment" (Butt, 2010, p. 5).

To discuss the exact extent of semantic contribution of light verbs is beyond the scope of this work, but it should be clear that they are not just functional elements.

Till now, we only saw examples of light verbs in combination with NPs and PPs, but some authors also include adjectival and verbal complements. But if we include the latter, we have to deal with the aforementioned resemblance of light verbs and minor verbs in asymmetric SVCs. E.g. when (Aikhenvald, 2018, p. 56) writes "a grammaticalized 'minor' verb will still retain full lexical status in the language outside the constructions", it sounds an awful lot like a passage from above. This is a known problem discussed in Seiss (2009).

The PARSEME annotation guidelines limit LVCs to eventive nouns as complements except for Hindi for which they also allow adjectives that are morphologically identical to an eventive noun. Two types of light verbs are allowed: 1. The verb contributes only "by bearing morphological features: person, number, tense, mood, as well as morphological aspect." (Khelil et al., 2022) (annotated as LVC.full) 2. The verb is causative (annotated as LVC.cause). As we have seen, this is in line with certain views about light verbs. But it is unclear whether it covers all relevant types. E.g. 1. does not seem to cover directionality.

LVCs are the mirror image of VPCs in that the latter "involve a wide range of verbs in combination with a small number of particles, [while the former] involve a small number of verbs in combination with a wide range of co-verbal elements" (Stevenson et al., 2004)[20]. Analogously to VPCs, the fact that one of the classes, the LVC draws from, is restricted, might be advantageous if enough eligible light verbs were seen during training.

2.5.5 Verbal Idioms

The most prominent and most discussed VMWE type (and the one which we will put our main focus on, too) is the verbal idiom (VID). The reason for its prominence might be that it is most closely associated with semantic idiomaticity. As we have seen, there are other types which share this property, but not to the same extent, since VIDs are typically completely non-compositional with no component providing its literal meaning. Beyond that, VIDs are very salient and 'stick out' in a text because they

[20]Some papers do not have page numbering, so in these cases we provide no page numbers with verbatim quotes.

often involve some kind of figuration like metaphors, hyperboles, etc. (Nunberg et al., 1994):

(52) John **kicked the bucket** after a long illness.
'John died after a long illness.'

(53) i need to **touch some grass** bruh.[21]
'I need to get in touch with reality/go outside more.'

(54) **Auf fremdem Arsch ist gut durch's Feuer reiten**.
On someone else's ass is convenient through fire ride.
It is convenient to ride through the fire on another (person's) ass.
'To get an advantage to the detriment of another person.'

Undoubtedly, the most famous example in the MWE literature is *kick the bucket* (cf. Example (52)), which very likely has a past as a metaphor, even if the underlying motivation is lost to present-day speakers. There exist various theories on the origin of the expression, but none of them seem verifiable.[22]

On the other hand, the motivation for the rather recently coined expression *touch (some) grass* is more clear once you get to know the meaning. It is often used in a context where people express the wish to get away from the virtual and back into the 'real' world (i.e. outside; the word *grass* elicits associations with nature). For example, the author of the tweet in Example (53), a gaming-influencer, reacts to his false assumption that a person was talking about a video game instead of real life.

Finally, the motivation for the proverb in Example (54) is quite easy to deduce and probably does not require an extensive elaboration. Hence, there are shades when it comes to the transparency of the motivation for idioms. It ranges from completely opaque (cf. Example (52)) over transparent, after one gets to know to the meaning (cf. Example (53)), to completely transparent (cf. Example (54)).

The VID subclass *kick the bucket* belongs to – verb noun idiomatic combinations (VNICs) (Baldwin and Kim, 2010) – is often the focus of studies regarding idioms. VNICs consist of a verb and a noun in object position and are quite frequent in English. When we talk and think about VIDs, it is often VNICs that come to mind, but it is important to note that

[21] Source: https://twitter.com/h7une/status/1552332420606615553, [Accessed: 02.02.2023]

[22] See https://en.wikipedia.org/wiki/Kick_the_bucket for a variety of different theories. [Accessed: 23.05.2024]

VIDs can take many forms and they are not as easy to distinguish from other VMWE categories as one might expect. It is actually more convenient to define them in distinction to other categories than to list all the possible patterns VIDs appear in. According to the PARSEME annotation guidelines a candidate expression is classified as a VID if its dependent is neither a reflexive pronoun nor a particle and if fine-grained tests exclude the categories MVC and LVC. Hence, in cases in which the dependent is an adjectival phrase, a relative clause, a non-reflexive pronoun, etc. the candidate is usually classified as a VID. According to PARSEME, even proverbs, i.e. whole clauses (cf. Example (54)) are subsumed under VIDs, despite having no open slots. But while it is relatively straightforward to distinguish VIDs from IRVs and VPCs, it is sometimes more difficult to draw the line between VIDs and LVCs, because they can occur in the same pattern. In some cases, they can even be identical:

(55) I told him to **take a hike**. It is beautiful around here in the spring!

(56) He was annoying me. I told him to **take a hike**.

Depending on the context, *take a hike* is either an instance of a LVC (cf. (55)) or a VID (56) and can have the meanings 'hike' or 'go away'.

Table 2.1 summarizes the possible patterns for the respective VMWE types according to PARSEME.

VMWE Type	**Pattern**
VPC	V + Particle
IRV	V + Refl. Clitic
LVC	V + N (or ADJ in Hindi)
MVC	V + V (+ V)*
VID	V + N, V + V (+ V)*, etc.

Table 2.1: Patterns of VMWE types (PARSEME).

It shows that all VMWE types, except VIDs, are quite restricted with respect to their syntactic patterns. In fact, the syntactic patterns are part of their definitions. VIDs are syntactically constrained in that they cannot have the same pattern as VPCs or IRVs, but otherwise their definition is purely based on semantic properties. This is explicitly stated in Ramisch et al. (2018, p. 224) who describe VIDs as "VMWEs not belonging to other categories, and most often having a relatively high degree of semantic non-compositionality". This is potentially challenging for classifiers, as VIDs have the largest degree of freedom regarding which components the classifier has to tag as VMWEs.

Another issue, which is not limited to, but mostly associated with VIDs, is the fact that there exist literal counterparts for some VID types:

(57) Full of anger, he kicked the bucket against the wall.

Here, a classifier needs to recognize that contrary to Example (52) there is some actual bucket kicking going on and the instance of *kick the bucket* should not be labeled a VMWE. The disambiguation of VIDs and their literal counterparts will be the main focus of this thesis, so we will go into much more detail about it in the following chapters.

Conclusion

In this chapter, we explored the different properties of MWEs, more precisely the different forms of idiomaticity. Furthermore, we discussed (non)-decomposability and its purported influence on the flexibility of certain MWEs. Finally, we presented the different types of VMWEs according to the PARSEME typology and tried to relate the PARSEME definitions to the literature about them. We also addressed some of their properties which might facilitate/exacerbate their automatic identification. The main focus of this chapter was put on linguistic aspects, however. In the following chapter, we will go much deeper into MWE processing in the context of NLP.

Chapter 3

MWE Processing

In the previous chapter, we looked at VMWEs mainly as a linguistic phenomenon but said little about how they are handled in NLP and what challenges arise during their processing. MWE processing encompasses two main tasks: MWE extraction and MWE identification. In the following, we explore how these two tasks are defined according to Constant et al. (2017) and how they interconnect. Then, we give an overview over common approaches to extraction and identification, respectively. Since our focus is on the latter, we cover it much more extensively. The section on extraction mainly serves as means to demarcate the two tasks. Finally, we cover a subtask of MWE identification: the disambiguation of MWEs and their literal counterparts.

3.1 MWE Extraction

MWE extraction describes the process of automatically identifying MWE **types** in a given text, that is, during extraction, we are not interested in the individual occurrences of a MWE (its tokens), but its underlying canonical form. Consider the following example that contains instances of multiple MWE types:

(1) Yes, I will **spill the beans** to the FBI, but you **spilled the beans** first! It seems, **the beans** will be **spilled** by both of us. So I need you to **get all the way off my back** about trustworthiness!

We would expect an extraction system to recognize that (1) contains the MWE types *all the way*, *get off sb.'s back* and *spill the beans*. And ideally it would not return all three variants of *spill the beans*, but only its canonical form. Also, we only want to extract the MWE types and not their contexts. Thus, the input to an extraction system is running text

and the output is a list of MWE types extracted from said text (cf. Figure 3.1). These lists can subsequently be reviewed by human experts in order to filter out false positives. The motivation for this is to create and update MWE lexicons which in turn can be used to identify MWE tokens. This is a crucial step for MWE processing, since some classes of MWEs are very productive and new types are created on a regular basis, not least driven by internet culture (e.g. expressions like *touch some grass*, *rage farming*[1], etc.).

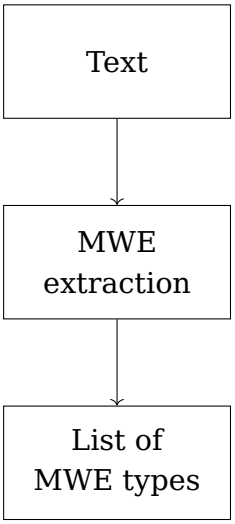


Figure 3.1: MWE extraction acc. to Constant et al. (2017).

One note on terminology: What we call MWE extraction is also known by many other terms in the literature, such as *discovery*, *acquisition*, *dictionary induction*, *learning* and – quite confusingly – *identification* (Constant et al., 2017). We opted for *extraction* because it is used quite commonly in the literature and does a good job emphasizing how the task differs from identification, since something is actually *extracted* (a list of MWE types) from a text as opposed to the identification task where we receive the same text as output but with annotations. Admittedly, we could also make the argument for *discovery*, as it emphasizes the detection of new MWEs but it seems to be less commonly used.

One of the most employed approaches to MWE extraction is the use of association measures, like pointwise mutual information (PMI), which measure how more often two words co-occur together than we would expect them to co-occur by chance (e.g. Evert (2005)). Hence, this approach is aimed at statistically idiomatic expressions and therefore

[1] `https://en.wikipedia.org/wiki/Rage-baiting` [Accessed: 04.06.2024]

includes pure collocations, that is, expressions that do not exhibit any other form of idiomaticity (like *traffic light*).

A more restrictive approach to MWE extraction is based on semantic similarity as it tries to leverage the semantic idiomaticity of MWEs and accordingly excludes collocations from its scope. These kind of approaches usually make use of word embeddings to represent the semantics of MWEs as well as its component words. The basic idea is then to compare the MWE embeddings with the component embeddings by measuring their similarity (often with the cosine measure). The problem with these kind of approaches is that they are typically demonstrated only on small sets of hand-crafted data, for example a list of noun compounds or verb-particle pairs coupled with judgements on their compositionality (Pickard, 2020). Please note how these works deviate from the definition of extraction above, since they do not use raw text as input but lists of pre-chosen expression types which may or may not be MWEs and are not presented in context. In terms of Figure 3.1 *Text* should be replaced by something like *Expression Types*, i.e. input and output are both on the type-level. Because of this somehow artificial setup, it is unclear whether these approaches are suitable for large-scale applications. In Section 3.3, we will see very similar approaches to MWE disambiguation, so it is important to emphasize how they differ.

The evaluation of MWE extraction is not trivial, especially if we want to evaluate the discovery of new MWEs, since we would not find them in conventional dictionaries. Thus, if we compared an MWE list generated by an extraction system to the entries of a lexicon, all types on our list not occurring in the lexicon would have to be regarded as false positives. Given the speed with which new MWEs are coined, this seems somewhat less than perfect. Another evaluation method is based on the judgement of human experts who rate a part of the extracted types. Due to the subjectivity of the method, this is also not without caveat. Last but not least, as for many other NLP tasks, extrinsic evaluation is possible by examining if the performance of a downstream task can be enhanced with the results of the extraction system (Constant et al., 2017).

3.2 MWE Identification

MWE Identification describes the process of automatically identifying **tokens** of MWE types in running text. The input to an identification system is running text and the output is said text with annotations marking the MWE instances (Constant et al., 2017). Hence, the difference between MWE extraction and identification lies in the output. While an extraction

system would take example 3.1 as input and produce a list of MWE types as output, an identification tool would return the same text, but with annotations marking all the instances of *spill the beans*, *all the way* and *get of sb.'s back*. Extraction is a potential earlier step for identification, since a list of MWE types could be used to match instances of said types (cf. Figure 3.2) in running text. But as we will see in the following, this is not without challenges.

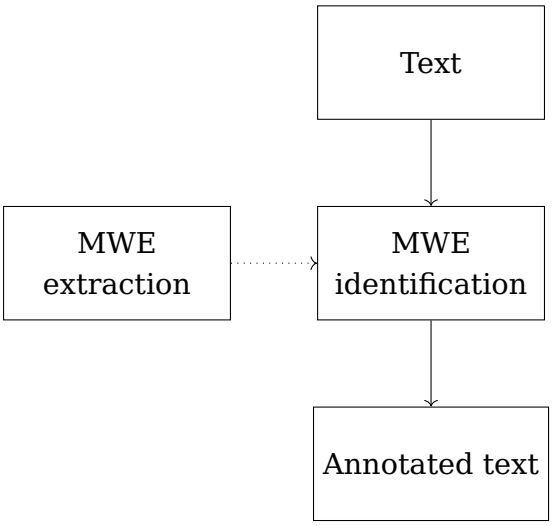


Figure 3.2: MWE identification acc. to Constant et al. (2017).

As mentioned above, MWE extraction is sometimes also called identification and it is understandable if for some the separation of the two tasks seems artificial. After all, the type/token-distinction disappears as soon as a system is supposed to identify previously unseen MWEs without a MWE lexicon because it then has to implicitly perform the extraction task by identifying new types[2]. But it nevertheless makes sense to separate the two tasks, since in general, their outputs are quite distinct.

Challenges to Identification

The challenges we face during MWE identification are manifold, especially for VMWEs. Example (2) from Constant et al. (2017) shows a sentence containing three MWEs: *now that*, *look up* and *dirty word*:

(2) **Now that** I **looked** the **dirty word up** , I understand .
B I O B o b i I O O O O

[2] *New* meaning in this context: not seen during training.

The instance of *look up* illustrates why VMWEs are especially challenging when it comes to identification. It would be acceptable to treat an expression like *now that* as a word-with-spaces because it is fixed and accordingly is not subject to morphosyntactic variation, but we cannot treat *look up* accordingly, since it is much more flexible. For one, *look up* receives the full range of verbal inflection. Furthermore, its components can be discontiguous, i.e. there can be intervening words between *look* and *up*. In (2), three words intervene (*the dirty word*) and this is a relatively mild case of discontinuity when compared to what German has to offer:

(3) Zusätzlich zum an sich teuren Atomprogramm **steht**
Additionally to the in itself expensive nuclear program stands
der Iran also einerseits mit weiteren, extrem hohen
the Iran therefore on the one hand with further, extreme high
Nebenkosten **da**.
additional costs there.
Additionally to the in itself very expensive nuclear program, Iran therefore on the one hand is burdened with further, extremly high additional costs. PARSEME shared task 1.1, DE data set (dev)

In Example (3), there is a gap of nine words between the two components of the VPC *dastehen* ('stand there'⇒'be burdened with') and this is not just some singular case as German is notorious for long distance dependencies of this nature. Of all the data sets in the PARSEME 1.0 corpus, German has the highest number of words between VMWE components (Savary et al., 2017).

Another challenge that arises from non-adjacency is that MWEs can overlap. In (2), the MWE *dirty word* appears between the two components of *look up*. This is problematic when it comes to labeling these kind of examples. One possible labeling scheme which accounts for embedded expressions is the IOB (Inside–outside–beginning) format (also referred to as BIO format) after (Schneider et al., 2014a) which is a variant of the original IOB scheme (Ramshaw and Marcus, 1995), a labeling format popular for tasks such as named entity recognition (NER). In both variants, the modified one and the original, the first constituents of MWEs are tagged with a B, while all subsequent parts belonging to the same expression are labeled with an I. Tokens not belonging to any MWE instance are labeled with O. Where the two annotation schemes differ is with regard to embedded expressions. If a MWE is embedded within another, lowercase letters are used (cf. Example (2)) . However, even this

labeling scheme is not fit to handle overlaps where MWEs share tokens:

(4) We should **turn up the heat** until they get nervous and make a
* * 1;2 1;2 1 1 * * * * * * *
mistake.
*

In Example (4), the sentence contains the VPC *turn up* ('increase') and the VID *turn up the heat* ('pressure sb.') with the VPC being part of the VID. The IOB labeling scheme would not be able to take this into account, since in cases like these we need to be able to assign multiple labels to a token. This is shown in the annotation layer below the sentence. The VPC/VID components *turn* and *up* are assigned two identifiers, separated by a semicolon, with every identifier representing a VMWE instance: *1* stands for the VID *turn up the heat* and 2 for the VPC *turn up*. This is a (simplified) example of the labeling scheme employed in the PARSEME data sets (Savary et al., 2017; Ramisch et al., 2018, 2020; Savary et al., 2023) which we will discuss in more detail later on.

This kind of overlap is no trivial issue for a system that is supposed to perform MWE identification because it has to assign multiple labels to a token. A possible workaround would be to always choose the longest sequence:

(5) We should **turn up the heat** until they get nervous and make a
* * B I I I * * * * * * *
mistake.
*

The problem is that we loose information about the MWE that is a factor of the other (*turn up* in this case) and it will not work "if MWEs share several tokens, but one MWE is not a factor of the other one despite sharing some elements, as in coordinated structures" (Constant et al., 2017, p. 856):

(6) He **took a walk** and **a shower**.

Due to the coordination, the two LVCs in Example (4), *take a walk* and *take a shower*, share the same verb and tagging the longest sequence would not yield the desired result.

An issue that arises on the token level is the ambiguity of some expressions which have a literal in addition to an idiomatic reading:

(7) If we do not want to miss the boat, we should be at the pier half an hour earlier.

This is especially problematic, if we use an MWE lexicon and just try to match its entries against usages in a text, since we have to consider the context to infer the correct reading. For example, if the lexicon contained *miss a boat* ('miss an opportunity') as an entry and we tried to perform string matching in example (7), we would receive a false positive, since the sentence actually contains an instance of its literal counterpart. We will spend much time exploring ambiguity and this is why we will gloss over it for the moment and come back to it in the next section.

Another non-trivial challenge is to decide which elements actually belong to a MWE and which do not. Regarding this matter, the PARSEME annotation guidelines[3] distinguish between lexicalized components and open slots. The former are those arguments of a MWE type that are always realized by the same lexemes, while the latter are arguments that can be realized by different ones:

(8) She **took** *him* **by surprise**.

Example (8) shows an instance *take someone by surprise* which is a VMWE where the head verb has two obligatory arguments: a direct object and a prepositional phrase. But only the prepositional phrase is lexicalized, i.e. it always has to be realized by *by surprise* in order to count as a valid usage example. The direct object on the other hand can be realized by a variety of lexemes. While this example represents a relatively easy case, the experimental category of inherently adpositional verbs (IAVs)[4] shows that it can be much more difficult to determine which elements are mandatory. According to the guidelines, IAVs consist of a verb or VMWE and a preposition which is always required, but the fact that it was an experimental category indicates a disagreement on the status of such expressions as VMWEs, i.e. a disagreement regarding the question whether the preposition is part of the expression or not.

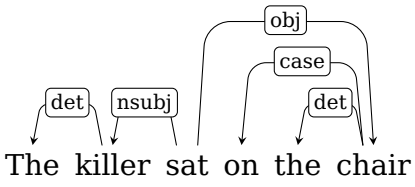
Motivation

One important question we have not addressed yet is how we can benefit from MWE identification. Why should we go through the trouble and perform this – as we have seen – very challenging task? Constant et al. (2017) cite improved parsing performance as one of the reasons. If identification is performed ahead of parsing, the search space could poten-

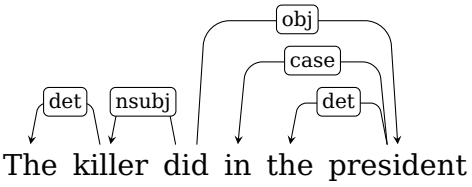
[3] `https://parsemefr.lis-lab.fr/parseme-st-guidelines/1.1/?page=lexicalized#lexicalized` [Accessed: 04.06.2024]

[4] "Experimental" in the context of the PARSEME data sets means that the annotation of IAVs was optional.

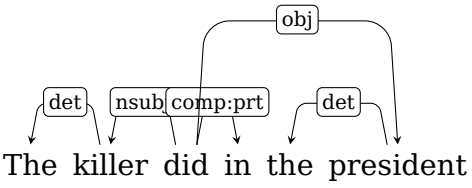
tially be narrowed down by excluding wrong parses involving MWE tokens. If, for example, we would correctly identify VPCs in a sentence, we could resolve some ambiguity with respect to the particles which often are homonymous with prepositions. Because of this homonymity, there is some ambiguity regarding the possible heads of the particle. Figure 3.3 shows different dependency parses for the sentence *The killer did in the president* ('The killer killed the president') (3.3 (b) and (c)).



(a) Correct parse of preposition.



(b) Wrong parse of VPC.



(c) Correct parse of VPC.

Figure 3.3: MWE identification and parsing.

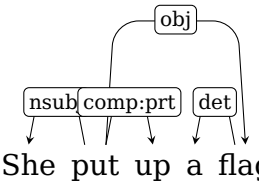
3.3 (b) shows an erroneous parse[5] where *in* – just like the preposition *on* in 3.3 (a) – is treated as a preposition by the parser and accordingly receives the wrong head (*president*) and the wrong dependency relation (*case*[6]). By contrast, in 3.3 (c), what would have been the correct parse, *in* is headed by *did* with the correct dependency relation *comp:prt*. Hence, if *did in* were to be correctly identified as a VPC prior to parsing, one wrong parsing step would already have been excluded.

Conversely, parsing can itself provide clues for identification, so the relationship between the two tasks is bidirectional. E.g. Figure 3.4 shows

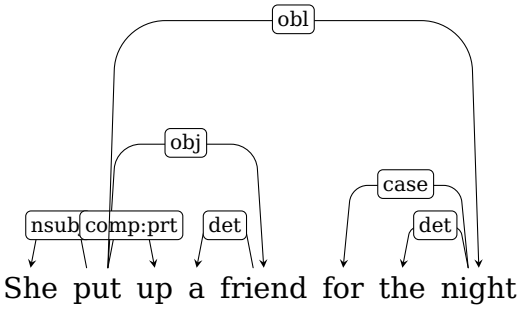
[5]Parsed with UDPipe 2.10 (Straka, 2018) and model english-ewt-ud-2.10-220711.

[6]In UD relations hold primarily between content words. This is why the noun is the head of the preposition and not the other way round as it would be the case in more traditional approaches.

two sentences containing the verb *put up*, one with the compositional reading ('erect/install') and one with the idiomatic reading ('accommodate').



(a) Compositional reading of put up.



(b) Idiomatic reading of put up.

Figure 3.4: MWE identification and parsing II.

Both sentences are correctly parsed in that *put* is the head of *up* and that it is the relation *comp:prt* holding between them, i.e. both are actually phrasal verbs. But according to our (and PARSEME's) definition of MWEs, we only consider *put up* in 3.4 (b) to be a VPC because *put up* in 3.4 (a) is compositional. Thus, even a correctly established *comp:prt*-relationship does not necessarily correspond to a correctly identified VPC. What it does, however, is that it helps to identify possible candidates for VPCs, so it could benefit the identification process as shown by Nagy T and Vincze (2014). Therefore, it can make sense perform the identification step after parsing. The obvious question is then how to orchestrate the tasks, i.e. whether to perform identification before, during or after parsing. Exploring this question is beyond the scope of this work, however.

Another task that can benefit from MWE identification is Machine Translation (MT), since the non-compositionality of semantically idiomatic expressions often prevents a word-for-word translation as in example (9)[7]:

[7]Source: `https://www.deepl.com/de/translator` [Accessed: 13.04.2023]

(9) You really need to leave the virtual world and **touch**
Du wirklich musst verlassen die virtuelle Welt und anfassen
some grass.
etwas Gras.
Translation: Sie müssen wirklich die virtuelle Welt verlassen und etwas Gras anfassen.

Here, the sentence contains the expression *touch some grass* ('go outside'/'get in touch with reality') which, at the time of this writing, is still translated literally into German by the online translation tool DeepL. It should be noted, however, that with the emergence of deep learning and huge parallel corpora, this problem has been greatly alleviated. Many of the former 'problem children' like *kick the bucket* and *spill the beans* are now correctly translated by DeepL and similar services. The fact that this remains an issue highlights the speed with which new MWEs are coined and the need for constant updates to parallel corpora or whatever MWE resources are used in order to keep track of them.

3.2.1 Approaches to MWE Identification

Not least because of the three editions of the PARSEME shared task (Savary et al., 2017; Ramisch et al., 2018, 2020), the amount of work on MWE identification increased dramatically in recent years. We will roughly divide those works into parsing-based and other approaches. When we talk about parsing-based identification methods, we mean joint methods (simultaneous parsing and MWE identification) and approaches that use tree structures gained by parsing prior to identification. We do not cover parsing with prior MWE identification because our focus lies on how parsing can benefit the identification task and not the other way round. The effect of MWE identification on parsing is of course also a very interesting topic, but not in the scope of this thesis.

Since many of the works discussed in this section were created in the context of the PARSEME shared task, we will briefly sketch it. As mentioned earlier, the PARSEME corpora (which we will present in more detail in Section 4.1) are highly multilingual, so the competing systems were evaluated on a high number of languages with the average score as the basis for the ranking. That means, if a team only submitted results for a few languages, it usually ranked lower compared to teams which included all (or almost all) languages. Participants of the shared task were able to submit results to two tracks: a closed and an open one. Systems competing in the closed track only were allowed to use resources pro-

vided by the organizers, while systems in the open track were free to use resources beyond that, e.g. lexicons, other corpora, word embeddings, etc. In the first two editions of the shared task (1.0 and 1.1), evaluation was based on precision, recall and F1-measure based on complete MWEs and MWE tokens. The former only included MWEs that were identified as a whole, while the latter also took partial hits into account. For the following example this would mean that, if a system only tagged *spill* and *beans* but not *the*, it would increase the true positive count for the token-based, but not for the MWE-based evaluation:

(10) He finally **spilled the beans**.
* * 1:VID * 1.

And it definitely makes sense to also consider partial matches in the evaluation. We presuppose in Example (10) that the determiner is a lexicalized part of the VMWE, but the question, which part of a MWE is lexicalized and which is not, is not so easy to answer. Consider the following example:

(11) Is anyone else feeling bummed that Marlene **spilled** too many **beans** in her interview after 7x19?[8]

This illustrates pretty clearly how the canonical form (*spill the beans*) is not always set in stone. One could even argue in light of this evidence that *the* should not be annotated at all as being part of this VMWE type, but it is likely that many annotators will annotate the canonical form of an expression when the encounter it. Hence, it makes a lot of sense to account for such uncertainties. We will return to this discussion in a different context in Section 5.1.1.

For the first edition, systems were ranked according to the per-token score and in subsequent editions according to the per-MWE score. Edition 1.2 based the per-MWE ranking on the identification of MWEs unseen during training, since a high amount of VMWE types seen during training were available in the test sets. Other, "inofficial" metrics were *Discontinuous vs Continuous VMWEs*, *Unseen-in-traindev vs Seen-in-traindev VMWEs*, *Variant-of-traindev vs Identical-to-traindev VMWEs* and *Single-token vs Multi-token VMWEs*.

If we do not state otherwise, we report the numbers for the evaluation metric that was decisive for the ranking in the respective shared task version, that is Token-based F1-score for 1.0, MWE-based for 1.1 and Unseen MWE-based for 1.2. Furthermore, contrary to subsequent

[8]Source: `https://prettylittleliars.fandom.com/f/p/3036680525185413979`, [Accessed: 28.03.2024]

editions, version 1.0 did not have an overall ranking (an average score over all languages), but only reported numbers for individual languages. Hence, we will content ourselves with discussing only the rankings for systems competing in version 1.0 as it seems a bit excessive to report numbers for 18 languages.

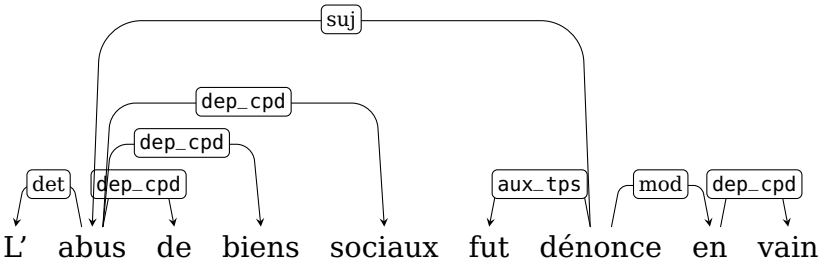
Please note that we can compare results of different PARSEME shared tasks only under reservation because the PARSEME corpora versions differ with respect to the languages they cover, e.g. English is part of 1.1, but not 1.0 and 1.2. In addition, even the language-specific data sets can vary as some were revised between corpora versions.

3.2.2 Parsing-based Approaches

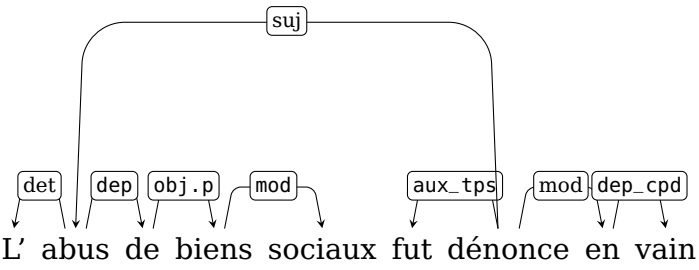
As mentioned in Section 3.2, the common expectation is that parsing benefits from MWE identification and vice versa, so it makes sense to perform both tasks jointly. Candito and Constant (2014) explore different strategies for joint dependency parsing and contiguous MWE identification. Among other things, they experiment with different representations for regular and irregular MWEs to study their effect on the performance of an off-the-shelf graph-based parser. To this end, they modify the annotation of regular MWEs in the data set for the SPMRL 2013 shared task (Seddah et al., 2013). Originally, the data set represents all types of MWEs as flat trees, where the leftmost MWE component is the head and all other MWE parts are its dependents (cf. Figure 3.5 (a)). While this makes sense for MWEs with irregular syntax, some valuable information could get lost if the structure of regular MWEs is not preserved. This is why the authors try to recover their regular syntax and represent them accordingly. Figure 3.5 (b) shows how the tree of the MWE *abus de bien sociaux* ('misuse of corporate assets') is not flat anymore (i.e. all dependents have the same head), but structured. For example, the fact that *sociaux* modifies *biens* is represented.

Furthermore, the authors experiment with a labeled representation that also incorporates the POS of the MWE and the information whether it is regular or not. Vincze et al. (2013) use a similar representation for LVCs which they employ to identify them during dependency parsing (likewise using a graph-based parser).

While the work above is an example for the modification of the MWE representation in a treebank to be later used in an off-the-shelf parser, Constant and Nivre (2016) not only propose a novel representation for regular and irregular MWEs, but also a new parsing architecture that jointly performs syntactic analysis and MWE identification. Their system



(a) Flat representation



(b) Structured representation

Figure 3.5: Different tree representations of MWEs.

is an extension of the arc-standard transition system and, in an effort to better accommodate for MWEs, jointly conducts syntactic and lexical analysis, i.e. it predicts dependency trees and lexical units of a sentence at the same time. This is achieved by adding three new transitions to the arc-standard system: $merge_F$, $merge_N$ and *complete*. Furthermore, it operates with two stacks: one for syntactic and one for lexical analysis. Figure 3.6 illustrates the representation of regular and irregular (i.e. fixed) MWEs. Again, the syntactic structure of regular MWEs – the nominal compound *prime minister* and the LVC *make decision* in the example – is preserved, but the representation of fixed ones is an alternative to the flat tree approach (cf. Figure 3.5, the MWEs connected by the *dep_cpd* labeled arcs), as they are represented by atomic nodes which are part of the syntactic structure. This is done by by the $merge_F$ transition which operates on both the syntactic and lexical stack at the same time. This way, not only a lexical node is created, but also a syntactic one. The $merge_N$ transition by contrast only creates a lexical node, that is, it merges the components of a regular MWE, but does not generate a special node in the syntactic tree, since its syntax is not out of the ordinary. In the example, it is the fixed expression *a few* whose tokens are grouped together and share the same incoming arc. As a consequence, the authors argue, the parser is not confused by flat subtrees which, despite their special label, might appear as ordinary dependency struc-

tures. In contrast to Candito and Constant (2014) this work is not limited to contiguous MWEs, but also accommodates non-contiguous ones.

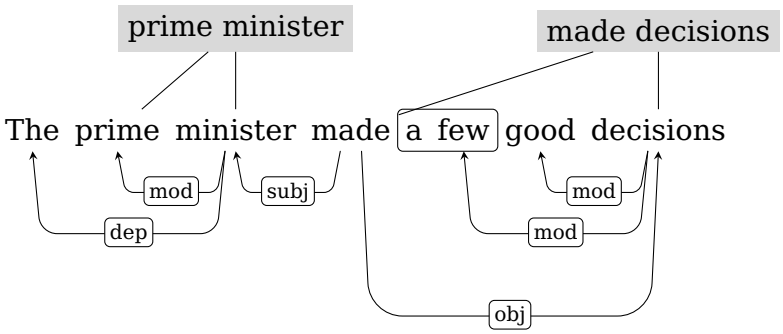


Figure 3.6: Representation of lexical and syntactic structure.

Not really parsing-*based* but parsing-*inspired* is the ATILF-LLF system by Al Saied et al. (2017). We still include it in this section because it is strongly related to the work of Constant and Nivre (2016). The ATILF-LLF system is basically a simplified version of the former, as it only includes the aforementioned transitions *merge*$_F$, *merge*$_N$ and *complete*, but none of the transitions that would create a syntactic tree. In other words: it only performs the lexical segmentation task and with it MWE identification. The reason for this is that the system was a participant for the PARSEME shared task 1.0 which did not include syntactic information for all of the 18 participating languages, so the authors decided to strip the syntactic parsing functionality. Nevertheless, syntactic information was included in the feature template for the MWE tagger whenever available. The system ranked first for all but two of the 18 languages in the PARSEME shared task 1.0.

TraPacc (Stodden et al., 2018) is another modified version of the system by Constant and Nivre (2016), i.e. an arc-standard transition-based parser without the operations relevant for dependency parsing. But in contrast to the system of Al Saied et al. (2017), the classifier used for learning and labelling transitions is a convolutional neural net (CNN). On top of that, the authors present TraPacc$_s$, a modified version of TraPacc whose softmax layer is replaced by a SVM. TraPacc$_s$ and TraPacc ranked second (49.74) and third (49.57) in the PARSEME shared task 1.1, respectively.

A parsing-based entry for the first PARSEME shared task was the FIPS system by Foufi et al. (2017). FIPS is a multilingual constituency parser that comes with a lexicon in which MWEs are already represented. The MWE identification is performed during parsing and basically consists of matching candidates against the database. The parser does not seem to

have been trained, but it was only used to generate output for the test set which was subsequently tagged according to the PARSEME annotation scheme using heuristics. FIPS was the only system submitted to the open track during the PARSEME shared task 1.0, so there are no other systems to compare its performance with.

Simkó et al. (2017) approached the shared task in a similar fashion as Vincze et al. (2013) the identification of LVCs: The dependency labels of the subtrees which represented multi-token MWEs were replaced by the type-specific MWE label. E.g. the relation *obj* was replaced by *LVC*. Subsequently, the Bohnet parser (Bohnet, 2010) was trained using this representation. Since it is not possible to represent single-token MWEs this way (such as German VPCs), the authors changed their POS-tags instead of their dependency labels and employed the POS tagging module of the Bohnet parser for identification. The system ranked first for two of the nine languages it competed in, but for the rest it only displayed a moderate to low performance.

TRAVERSAL, a system submitted to edition 1.1 of the PARSEME shared task (Waszczuk, 2018), is an example for a parsing-based approach that performs the identification after and not during parsing, as the system relies on dependency trees being created beforehand. The assumption being that MWEs form connected syntactic components, the possible labelings of these trees are then encoded as tree traversals, where each traversal corresponds to a different labeling. The optimal global labeling of a tree is determined with multiclass logistic regression. TRAVERSAL ranked first in the official rankings of the PARSEME shared task 1.1 with an MWE-based F1-score of 54. Extending on this work, Waszczuk et al. (2019) tackled the task with a neural graph parsing-based approach which employed a multilayer perceptron (MLP) to model the probabilities of different labelings. This method represented an improvement over TRAVERSAL and showed similar performance to the state of the art system at that time (Saied et al., 2018).

For edition 1.2 of the PARSEME shared task, Taslimipoor et al. (2020) employed a system that jointly performed MWE identification and dependency parsing with the latter serving as an auxiliary task, i.e. the parsing performance was only secondary and consequently not evaluated. The motivation behind this approach is the assumed benefit of syntactic parsing on MWE identification. The model, MTLB-STRUCT consists of BERT with two classifiers on top: One linear classifier for MWE tagging and a dependency parser consisting of a linear and a bilinear layer, followed by a tree CRF. For both classifiers, the BERT weights are shared and fine-tuned during training. The objective function consists of adding the

loss for MWE tagging, $Loss_{mwe}$, and the loss of the predicted dependency parsed tree, $Loss_{dep}$. However, the impact of $Loss_{dep}$ is diminished by multiplying it with a constant α which is larger than zero:

$$Loss = Loss_{mwe} + \alpha * Loss_{dep} \tag{3.1}$$

Depending on the performance, the value of α was empirically set to $\frac{1}{300}$ for some languages and to $\frac{1}{700}$ for others. These low values beg the question of how much influence the simultaneous parsing really has on the performance of the MWE tagger at the end. The same goes for an examination of the results. For both, the global MWE-based and the unseen MWE-based scores, the performance was better without the dependency parsing as auxiliary task (i.e. for systems only tagging MWEs) for six of 14 languages. And for some languages, the performance gain of the multi-task setting was only marginal, e.g. 66.22 vs. 66.01 for IT or 76.42 vs. 76.36 for PT (global MWE-based F1-score). Hence, a more thorough investigation of the effect of parsing on MWE identification needs to be conducted. Be that as it may, MTLB-STRUCT was the best performing system at the PARSEME shared task 1.2 with an Unseen MWE-based F1-score of 38.53 (Global MWE-based: 70.14, Global Token-based: 74.14).

3.2.3 Other Approaches

While the first PARSEME shared task was dominated by parsing-related architectures, the subsequent editions saw a surge in sequence labeling approaches, in particular neural ones. But also other non-parsing-based graph-based approaches were employed.

Maldonado et al. (2017) were one of two participating teams employing sequence labeling with CRFs during the PARSEME shared task 1.0. The morphosyntactic features provided by the organizers were used to form one feature template per language family. To boost their scores in an unofficial experiment (i.e. it was not submitted to the shared task), they re-ranked the 10 most likely predictions made by the CRF using distributional vectors. Their system ranked second for most languages in the official evaluation. Furthermore, they make the interesting point that a simple lookup baseline system could have beaten all systems in the competition, since the test set contains a large amount of VMWE types seen during training.

Although the rise of neural architectures was already well underway in 2017, during the first edition, the only system relying on neural nets was submitted by Klyueva et al. (2017). MUMULS, as it was named, is a bidirectional recurrent neural net with gated recurrent units (GRUs).

The intuition behind employing a recurrent neural net is to leverage their capabilities of capturing long-distance dependencies which is potentially useful given the discontiguity of some VMWEs. The input to the neural architecture were randomly initialized embeddings for tokens, lemmas and POS tags with 100 dimensions each and the optimization was conducted with ADAM. Despite relying on a neural model, the performance of MUMULS was relatively modest for most languages.

Another sequence labeling approach based on neural nets stems from (Zampieri et al., 2018). They also use a BiLSTM with randomly initialized embeddings, but in contrast to Klyueva et al. (2017) the embeddings are pre-trained on the PARSEME corpora and then fine-tuned during training. Another difference is that the authors convert the PARSEME annotations into the BIO labeling format introduced in section 3.2. More precisely, they experimented with different BIO variants and determined which one to use based on the dev set performance. They settled for a BIO scheme enriched with VMWE category labels, shown in example (12):

(12) La musique n’ adoucit pas toujours les mœurs .
BVID IVID g IVID g g IVID IVID O

In this annotation format, the beginning of a VMWE is tagged with an uppercase *B*, while the following VMWE constituents are labeled with an uppercase *I*. Interfering tokens not belonging to the VMWE are labeled with lowercase *g* (for *gap*). The system ranked 9th overall (of 13 entries) in the PARSEME shared task 1.1 with a score of 36.94.

Taslimipoor and Rohanian (2018) employ a neural architecture by the name of SHOMA that incorporates two convolutional layers coupled with a BiLSTM. The role of the convolutional layers is to function as N-gram detectors and their respective outputs are concatenated to be fed into the BiLSTM. Furthermore, the authors experiment with a CRF layer to capture potential interdependencies between labels. However, the system performed significantly worse with the CRF than without. Since SHOMA used pre-trained word embeddings, it competed in the open track of the PARSEME shared task 1.1. Among the four entries for the open track, it ranked first and achieved the best performance of all systems in the competition with a score of 58.09 (if we include closed track-systems).

Boroş and Burtica (2018) employ a BiLSTM (GDB-NER) together with graph-based encoding/decoding. But unlike Waszczuk (2018), their approach does not rely on syntactic dependency graphs but on fully connected subgraphs. In this architecture, the input is fed into the BiLSTM which outputs contextualized representations which are then split into

three projection embeddings using tanh layers. The first and second of these projection embeddings are subsequently used to build an adjacency matrix which contains the probabilities whether a pair of words is part of the same high-level expression (i.e. a MWE). Next, subgraphs (subgraphs = MWEs) are extracted from this matrix via backtracking. Finally, the third projection embeddings are fed into a an LSTM in many-to-one fashion to determine the label of an expression. The authors submitted two versions of their system to the PARSEME shared task 1.1, GBD-NER-standard and GBD-NER-resplit, which ranked 7th and 8th, respectively. However, they claim that the performance of their system was influenced by a bug during the competition and in their system description paper they report results that would have ranked them first in the competition with an F1-score of 56.65.

For the PARSEME shared task 1.2, Kurfali (2020) tackled the task as a sequence labeling problem using BERT. Figure 3.7[9] illustrates how token classification with BERT generally works. First, the input tokens are fed into the BERT encoder which outputs contextualized representations. These representations are then used as input to a linear layer which conducts the classification by assigning a label to every one of them. To apply this architecture to the PARSEME data, the annotations are converted to the IOB labeling scheme in similar fashion to Zampieri et al. (2018). An issue that arises when using BERT is that tokens can be split into multiple sub-tokens, so one can end up with more BERT representations than input tokens. In these cases, only the first sub-token was fed into the linear classifier and thus the basis for assigning the label. Furthermore, they compared the performance of mono- vs multi-lingual BERT and found that the former offers much greater generalization capabilities. The two systems, TRAVIS-multi and TRAVIS-mono, ranked second (30.48) and third (26.04), respectively. However, TRAVIS-multi only ranked higher because for TRAVIS-mono less results were reported which decreased its average score.

Conclusion

In this section, we have seen a variety of approaches tackling VMWE identification, mainly in the context of the PARSEME shared tasks. We divided these into parsing-based and other (non-parsing-based) approaches. There is no clear picture on what works best, since a variety of methods performed quite well in the shared tasks: parsing-based (Al Saied

[9] Source: `https://d2l.ai/chapter_natural-language-processing-applications/finetuning-bert.html`, [Accessed: 03.05.23]

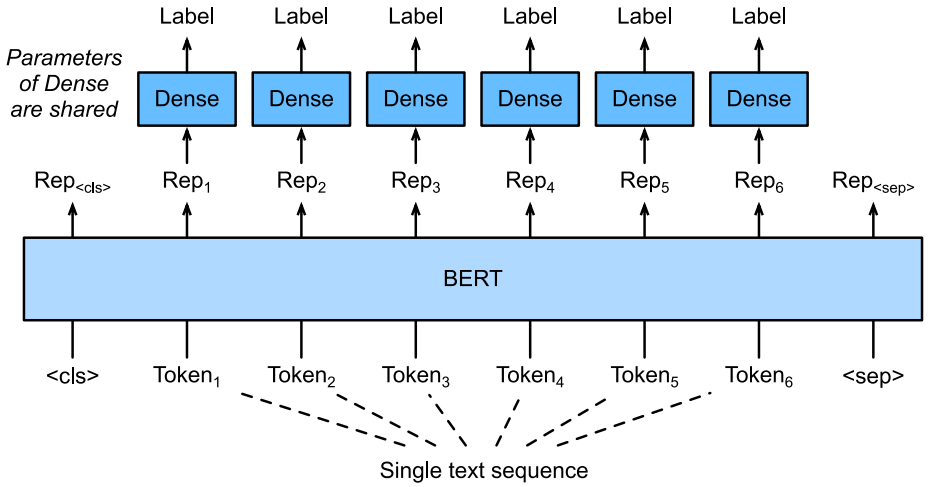


Figure 3.7: Token-level classification with BERT (Zhang et al., 2023).

et al. (2017), Waszczuk (2018)), graph-based (Boroş and Burtica (2018)) sequence labeling (Taslimipoor et al. (2020)) or multi-task (Taslimipoor et al. (2020)). Maybe the division between parsing-based and non-parsing methods is a bit artificial anyway, since most successful systems rely on morphosyntactic features one way or the other.

3.3 MWE Disambiguation

This section is concerned with a subtask of VMWE identification: the disambiguation of VIDs and their literal counterparts. For illustration, consider the following two sentences which contain both the string *kicked the bucket*:

(13) After a long illness he finally **kicked the bucket**.

(14) He kicked the bucket and broke his toe.

In Example (13) *kicked the bucket* is used in its idiomatic sense and means 'to die', i.e. it is an instance of the VID type *kick the bucket*. In (14), however, the string occurs in its literal reading, thus it describes the act of kicking an actual bucket. Due to this ambiguity, a simple approach not considering context – like matching strings to entries in a MWE lexicon – would yield false positives for some cases. However, it is important to note that literal counterparts of VMWE types are quite a rare phenomenon. Savary et al. (2019) found in a corpus study that the idiomaticity rate[10] for all five examined PARSEME 1.1 data sets (Basque,

[10]The term denotes the percentage of idiomatic instances.

German, Greek, Polish and Portuguese) was at least 96%. This result is not that surprising considering the often figurative nature of semantically idiomatic VMWEs. For example, one would be hard-pressed to find a literal occurrence of the colorful expression *talk to Huey on the big white telephone* as seen in (15)[11].

(15) He is in the bathroom and talks to Huey on the big white telephone.

As we will see in Section 5.1, there are of course expressions with significantly lower idiomaticity rates (even below 50%), but these seem to be the minority. Further evidence in this direction will be presented in Section 4.3, when we present a number of dedicated corpora.

Before continuing, we will discuss a bit of terminology which can be tricky when talking about MWE disambiguation. When looking at sentences (13) and (14) only in the former the string *kicked the bucket* can be considered a VID. But how do we denote the instance of the string in (14)? Following Savary et al. (2019), we would call it a *literal occurrence* of a MWE. However, one could argue this is not correct as strictly speaking it is not a MWE at all: A literal combination of words cannot be an occurrence of an idiomatic expression. Haagsma et al. (2020) found a creative solution by introducing the term *potentially idiomatic expression* (PIE) which covers literal and idiomatic instances at the same time. A downside to this expression is that it might evoke the assumption that the kind of expressions we are talking about usually occur literally and only sometimes come in their idiomatic form, whereas in fact, it is usually the other way round. But you cannot have your PIE and eat it, too, so we will settle for the term *PIE* when talking about literal and idiomatic occurrences at the same time, as it is still the most elegant solution[12].

In view of the rarity of literal readings the question arises why we should concern ourselves with the disambiguation task at all. One reason is the magnitude of the errors emerging when ignoring the issue. As an illustration, consider the output of two semantic parsers (AMR and DRS). Figure 3.8 shows two parses of the sentence *After his illness, John kicked the bucket*. On the left is the output of a DRS (Discourse Representation Structure) parser (Liu et al., 2018) and on the right that of an AMR (Abstract Meaning Representation) parser (Damonte et al., 2016). Both incorrectly yield the literal reading for *kick the bucket* which results in

[11]The *big white telephone* is a toilet and *Huey* is an onomatopoetic expression mimicking the sound a person makes when being sick (i.e. vomiting)

[12]And on top of that it allows for a cornucopia of puns which, as we all know, are an integral part of naming conventions for papers in the MWE literature (and beyond that).

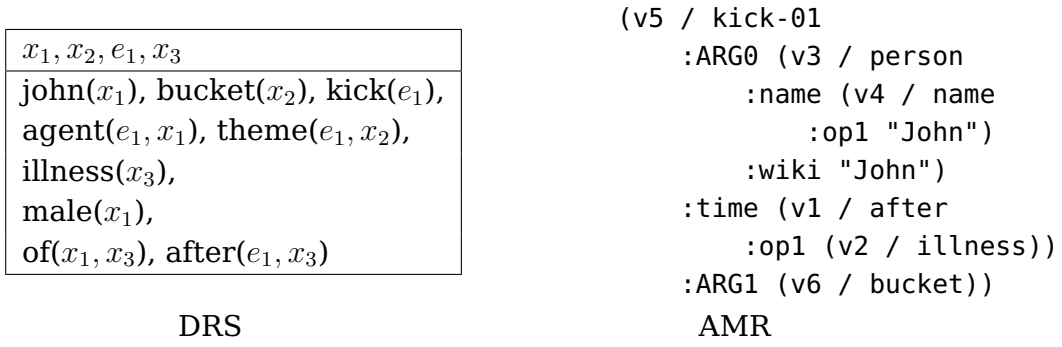


Figure 3.8: DRS and AMR for *After his illness, John kicked the bucket* with the parsers from Liu et al. (2018) and Damonte et al. (2016).

a literal treatment of *kick the bucket*: In the DRS parse *bucket* is a discourse referent (x_2) and in the AMR parse it is an argument (ARG1), which should not be the case for an idiomatic interpretation. The reason for the AMR parser's error can be found in its training data. Even the most recent release of the AMR bank (Knight et al., 2021) does not fully cover the annotation of VIDs. In a sample of 50 VID types, with one example per type, 38 were not annotated with the desired semantic representation.[13] Likewise, the parallel meaning bank (PMB) (Abzianidze et al., 2017) does not contain correct annotations of non-decomposable VIDs.[14] Another application that could benefit from the disambiguation of PIEs is machine translation as shown in Example (9) in Section 3.2.

To conclude, even if literal readings are only a minor issue regarding quantity, qualitatively speaking it is still a major problem for some NLP applications.

Since MWE disambiguation is a subtask of MWE identification one could pose the question why we treat disambiguation as a separate task. After all, identification systems as described above should perform the disambiguation step implicitly by not assigning labels to the literal instances in running text, e.g. *kicked the bucket* in (13) would be annotated, but not in (14). The reason why disambiguation is treated as a separate task is due to the fact that MWE-dedicated corpora often do not contain a lot of literal occurrences of PIEs (cf. Savary et al. (2019). On top of that, literal occurrences are usually not annotated, which makes the evaluation quite difficult. Because of these limitations, special cor-

[13]However, as Bonn et al. (2023) shows, the AMR and UMR (Uniform Meaning Representation) community is aware of the issue by now.

[14]Decomposable VIDs are an issue to a lower degree, since, in theory, they allow for a one-to-one mapping from words to concepts.

pora are created which contain both: a sufficient number of literal occurrences and their respective annotations. These kind of corpora will be discussed at length in Chapter 4, where we present a variety of PIE corpora created by others, and in Chapter 5, where we present our own PIE corpus for German.

3.3.1 Approaches to MWE Disambiguation

Now that we established reasons why one might want to tackle the task of MWE disambiguation, in this section, we will have a look at a variety of different approaches. As with "normal" word sense disambiguation (WSD), the context plays a very important role during classification. E.g. in (13), we can use the context clues *illness* or *finally* to determine the idiomatic reading of *kick the bucket*. In (14), it is *broke his toe* that leads us to favor the interpretation where someone strikes a pail with his foot. In addition to semantic clues we potentially can draw on morphosyntactic features of MWEs to infer the correct class. As was addressed in Section 2.2, some MWE types are said to be more fixed than their literal counterparts. If *bucket* were to appear in plural, for example, this would be a good sign for a literal reading (or a spelling error[15]).

There exist different approaches leveraging semantic features to infer the correct reading of (verbal) PIEs. Katz and Giesbrecht (2006) use static word vectors to identify literal and idiomatic occurrences of the German PIE *ins Wasser fallen* ('fall into the water'⇒'get cancelled'), the assumption being that the contexts of the literal and idiomatic use of this expression differ which should be represented by their distributional vectors. To this end, they annotated 67 occurrences of *ins Wasser fallen*, whether they were used literally or idiomatically, and used this to create two word vectors: one for the literal and one for the idiomatic usage. Test instances whose vectors are computed on the basis of their local context (a 30 word window), are then compared to these vectors in order to classify them with a simple nearest neighbor classification. In a second experiment, the authors build what they call a "compositional meaning vector" by summing up the vectors of the individual components. They then use this to classify preposition-noun-verb "collocation candidates" regarding their compositionality, the idea being that the compositional vector and the vector for a non-compositional collocation as a whole are far apart.

To determine how close a vector is to another, the cosine is used. A

[15] A google search delivers more than a few documents for the string"kicked the buckets" where the reading is clearly idiomatic.

cosine value of 1 signifies maximum similarity, while a value of 0 means the vectors are orthogonal to each other, i.e. they could not be further apart[16] and this accordingly signifies minimum similarity between the words. Figure 3.9 illustrates how the vector for *Löffel* ('spoon') is close to the vector for *essen* ('eat'), while *den Löffel abgeben* ('die') is closer to *sterben* ('die').

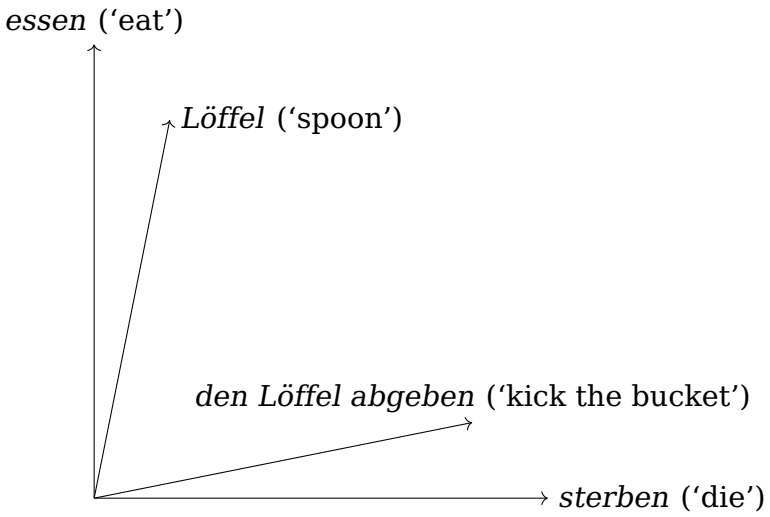


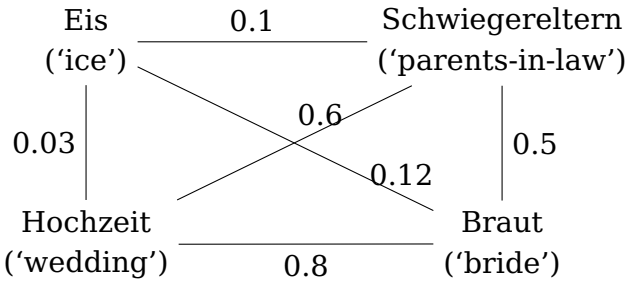
Figure 3.9: Vector-based disambiguation.

For the first experiment, the authors perform a 10-fold cross-validation study and report an average accuracy of 72%, which exceeds the simple maximum-likelihood baseline of 58%. For the second experiment, the authors explore different cosine-thresholds. The best one ($\cos < 0.2$) achieved an F-score of 0.48.

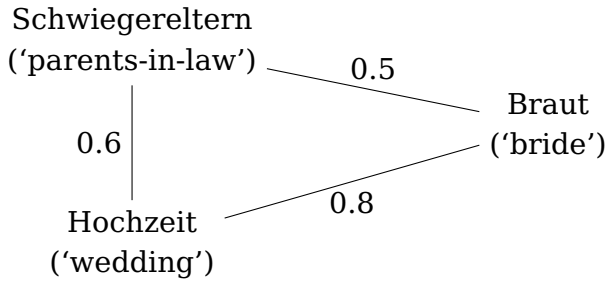
Li and Sporleder (2009) employ the notion of cohesion to approach the disambiguation task by presuming that VIDs disrupt the context they appear in. To classify test instances cohesion-based graphs are built based on a metric called Normalized Google Distance (NGD) and if the mean value inside the graph rises after the removal of the instance, it is classified as idiomatic. This method is evaluated on a data set of 17 PIE types (like *pull the trigger* or *back the wrong horse*) whose 3964 instances were extracted from the English Gigaword corpus. Ehren (2017) follows up on this idea by using the cosine between word embeddings as a metric for semantic similarity instead of NGD. This approach is tested on a German data set consisting of 15 PIE types (e.g *das Eis brechen* or *die Fäden ziehen*) which were extracted from TüPP-D/Z (Tübinger partiell

[16] Please not that this is only the case if the vectors consist solely of positive numbers. If we introduce negative numbers, we have a potential range of 1 to -1.

geparstes Korpus Deutsch/Zeitung). Figure 3.9 shows a graph representing a German sentence, where the expression *das Eis brechen* ('break the ice'⇒'relieve tension in a social situation') appears in the context of a wedding.[17] The top graph still contains the noun *Eis* ('ice'), while it is removed in the bottom one. The cosine mean rises after the removal which we would interpret as a sign for an idiomatic expression disrupting the cohesion before the removal. Accordingly, the instance would be labeled as *idiomatic*. For both, Li and Sporleder (2009) and Ehren (2017), this cohesion-based approach slightly outperforms the majority baseline.



Mean: 0.36



Mean: 0.63

Figure 3.10: Cohesion-based disambiguation.

Haagsma et al. (2018) expand on the cohesion-based approach even further and incorporate idiom literalisations, a method where the figurative sense of a PIE is represented by a literal paraphrase which is inserted into the context instead of the PIE instance. The hypothesis behind this is that the literalisation fits nicely into the original context if the PIE is used figuratively. However, this modification is not able to outperform the original approach described above. The approach is evaluated on a

[17] The graph only contains nouns as they were perceived as the most relevant content words.

corpus consisting of the VNC-Tokens data set (Cook et al., 2008), (a subset of) the IDIX corpus (Sporleder et al., 2010), the SemEval-2013 Task 5b data set (Korkontzelos et al., 2013) and some self-annotated data. The dev set consists of 8,235 instances of 299 PIE types and the test set of 3,073 instances of 146 PIE types.

The approach of Li et al. (2010) is closely related to the cohesion-based approach in that it uses a topic model to disambiguate PIEs. It was able to outperform the majority baseline as well as the original cohesion-based classifier of Sporleder and Li (2009) (on the same data set) by a small margin. Peng et al. (2014) also make use of a topic model, but they add an interesting feature to their system: Following the observation that idiom usage is often accompanied by a non-neutral stance towards the things they denote (Nunberg et al., 1994), they incorporate knowledge about the strength of emotions expressed in a text. To achieve this, they make use of a database which contains annotations for the arousal associated with a word. The authors evaluate their approach on four small datasets comprising of several paragraphs which contain an instance of one of four PIE types: *blow whistle*, *make scene*, *loose head* and *take heart* (so one data set per PIE type). All these instances were extracted from the VNC-Tokens data set (Cook et al., 2008). They find that the topic representation outperforms the baseline, a simple bag-of-words model, and that the arousal feature has a positive effect on the overall performance.

While most works presented in this section are concerned with typical (English) idioms, i.e. verb-noun idiomatic combinations (VNICs), Köper and Schulte im Walde (2016) tackle the task of distinguishing literal and non-literal use of German particle verbs (PVs). To this end, they build a corpus of 6436 sentences containing particle verbs with 4174 literal and 2262 non-literal usages. During their experiments they employ a random tree classifier whose best performing feature is the distributional fit between the PV's base verb and its context and the PV itself and its context, the assumption being that a PV is used literally if the context and the PV's base verb have a high cosine similarity:

(16) **Säge** den Ast lieber **ab**, bevor er während eines Sturms
Saw the branch better off before it during a storm
herunterfällt.
falls down.
'Better saw off the branch before it falls down during a storm.

(17) Sie haben den Abteilungsleiter **abgesägt**, bevor er dem
They have the department head saw off before he the

Unternehmen noch mehr Schaden konnte.
company even more damage could.
'They fired the department head before he could damage the company even more.'

In Example (16), we would expect a high similarity of the base verb *sägen* ('saw') and its context, but a low one in Example (17). Using this feature alone, the classifier achieved an F1-score of 83 for the literal and an F1-score of 61.8 for the non-literal class (majority baseline: 78.7 for the literal class). Other well-performing features were abstractness and concreteness ratings. These were deemed useful as – according to the authors – "non-literal expressions tend to occur with abstract words" (Köper and Schulte im Walde, 2016, p. 355). Only relying on this feature, the system achieved an F1-score of 81.3 for the literal and an F1-score of 60.7 for the non-literal class. Combining these two with the unigrams feature resulted in an even better performance with 88.6 for literal and 77.1 for non-literal.

Salton et al. (2016) explore the use of Skip-Thought sentence embeddings (Kiros et al., 2015) as input for k-nearest neighbors classifiers and SVMs. The motivation behind this is that the surrounding sentences may contain important information for the disambiguation of a PIE and thus should be considered during classification. Since a lot of corpora do not contain more than one sentence per instance, we often do not have access to this kind of context. The solution proposed by the authors is to use an existing Skip-Thought (or Sent2Vec) model to embed a sentence containing a PIE which in turn gives us information about the surrounding sentences. Salton et al. compared their system to that of Peng et al. (2014) by evaluating it on the same four data sets as them. It turns out, the performances of the two systems are very similar: For two of the data sets, the systems have the same F1-score and for the other two, one outperforms the other, respectively. The question also arises whether Salton et al. really perform PIE disambiguation. After all, they embed the whole sentence without giving the classifier the location of the PIE instance, so one could argue, they actually perform text classification. This is something we try to avoid with our own architecture which we present in Section 6.2.1.

As attractive as these kind of approaches are, their reliance on distributional properties bears some pitfalls as well. Consider the cohesion-based approach for example. It depends on the assumption that because of the non-compositionality of VIDs their components usually appear in different contexts when used literally. But it is not uncommon for authors to "play" with these literal meanings and to construct a metaphorical con-

text around the VID instance:

(18) He just couldn't help it, he had to play with fire and now the house burnt to the ground with the ashes of his aspirations flying in the wind.

Although there also is a clue indicating the idiomatic meaning, most content words in (24) suggest a literal reading if we base our decision on shared contexts.

But luckily we are not restricted to semantic features when it comes to PIEs. Hashimoto et al. (2006) use lexical knowledge to disambiguate between literal and non-literal readings of Japanese PIEs. They implement this knowledge in an idiom dictionary which contains a variety of restrictions pertaining to the targeted idioms like adnominal modification constraint or voice constraint. Their idiom recognizer then makes use of this dictionary in order to perform the disambiguation. A particularity of their approach is the prior classification of idioms into three different classes (class A, B and C) which, according to the authors, loosely correspond to the distinction of fixed (class A), semi-fixed (class B) and syntactically-flexible (class C) expressions in Sag et al. (2002). However, among those three classes only members of class C are considered ambiguous, which is not in line with what we established in Section 2.5.5: At least semi-fixed expressions as *kick the bucket* clearly have the potential of being ambiguous. However, Hashimoto et al. state that this is not relevant for their data as virtually no literal instances occur in class A or B. During classification, different disambiguation knowledge is used for the different classes. For class C – the only class relevant for disambiguation – they report an F1-score of 80 for a rather small test set containing only 108 instances.

In similar (but more general) fashion, Cook et al. (2007) explore the use of knowledge about lexicosyntactic fixedness to classify instances of PIEs. They assume that usages of VIDs usually exhibit limited variation in that regard and tend to occur in only a small number of canonical forms, leaning on the statistical method of Fazly and Stevenson (2006) to determine these forms. Literal usages in contrast show more flexibility and occur in a variety of patterns. Thus, if a PIE instance's pattern differs from the canonical form of a VID, it is a strong indication for a literal reading:

(19) The bucket was kicked.

For example, the passivization of *kick the bucket* in (19) could be a clue for a literal reading, as we discussed in earlier chapters, and a method

relying on canonical forms would classify this instance as literal. Cook et al. achieve a macro-averaged accuracy of 72.4 with this approach, outperforming the majority baseline (61.9) substantially.

Boukobza and Rappoport (2009) employ what they call surface and syntactic features as input for a SVM. Surface features include, for example, the distance between PIE components or the information if the expression appears in its canonical form. Syntactic features are dependency-based, like the number of edges in the minimal subtree or a list of dependency types between each component pair. The evaluation was conducted on a data set of 3,350 sentences from the BNC containing instances of 24 verbal PIEs. The authors report a maximum F1-score of 94.86 for one of their supervised models based on surface features which outperformed the syntactic features (87.77) as well as two rule-based baselines leveraging canonical forms (80.70 and 75.53, respectively). The numbers seem impressive, but the authors provide little information on the data set that would facilitate the interpretation of the results, e.g. the idiomaticity rate or how many accidental co-occurrences it contains.

In a similar vein, Diab and Bhutada (2009) use a SVM with shallow features like POS tag, lemma or the last three characters of a token to capture inflectional and derivational morphology. On top of that, information about named entities is incorporated. The advantage of such methods is that we do not have to rely on (much of)[18] the context to make an informed decision on the correct reading. This approach is evaluated on the VNC-tokens data set and achieved an F1-score of 84.58.

It is of course possible and desirable to combine semantic and morphosyntactic features for the disambiguation task. Gharbieh et al. (2016) make use of both kinds of features in their word embedding approach. First, they build a vector for a PIE type by summing up the embeddings of its components' lemmas which is then averaged. Afterwards, they separately select the context words for the verb and noun component of a PIE instance in a given window because those might be discontiguous. In both cases the embeddings for the context words are again summed up and averaged to form two context vectors, one for the verb and one for the noun. These two vectors are summed up and averaged as well. The vector for the relevant PIE type is then subtracted from the context to receive yet another vector which is enriched with a single binary feature giving the information if the PIE instance occurs in it canonical form. Finally, this feature vector is used as input to a SVM classifier. They also employ a second, unsupervised, method which performs k-means clustering using the same vector representation for PIEs. Again, the evaluation

[18] E.g. a noun modifier would be part of the context, strictly speaking.

was conducted on the VNC-Tokens data set. The supervised approach achieved an F1-score of 88.3 on the test set, outperforming the majority (62.1) and canonical form baseline (72.3) substantially. The unsupervised method achieved a significantly lower F1-score of 78.1.

Haagsma (2020) employ a variety of LSTM-based approaches to tackle PIE disambiguation. These systems are trained and evaluated on the MAGPIE corpus (Haagsma et al., 2020) and compared to several baselines. One of these, the most-frequent-sense-per-type (MFS-per-type) baseline, assigns the most frequent label for a PIE type encountered in the train set. It already performs quite strongly (91.99 harmonic mean accuracy) and is way ahead of the normal majority baseline (80.07). Their most basic system consists of a (Bi)LSTM layer whose input is a truncated sequence containing a PIE instance (i.e. a PIE instance and some context) who are encoded as GloVe vectors (Pennington et al., 2014). This input is then encoded into a single vector which is in turn fed into a softmax layer to predict the final label (*literal* or *idiomatic*). This architecture is then enhanced with an attention mechanism as well as indicator features. These indicator features, a 1 or a 0 concatenated to the input embeddings, are supposed to mark the PIE components or the entire PIE span which includes the words in-between the first and last PIE component. Unsurprisingly, the enhanced version of their system performs significantly better then the basic version. This was to be expected as the basic system does not mark the PIE instances and thus, one could argue, does not really perform PIE disambiguation, but text classification, since it cannot really know it is supposed to pay special attention to certain tokens. We probably cannot even be sure it achieves this by virtue of indicator features, since it is hard to grasp whether the system reliably learns to consider one simple integer concatenated to a 300-dimensional vector. However, the fact that by adding indicator features the performance improves by 1.8 points is a good sign it has some effect after all. Haagsma's best system achieves a harmonic mean accuracy of 94.81, outperforming the MFS-per-type baseline. Interestingly, PIE spans perform a bit better than marking individual PIE components. Furthermore, the performance gain using the attention mechanism is only modest (+0.13).

Conclusion

In the previous section, we have seen a variety of PIE disambiguation approaches. Traditionally, a lot of these were unsupervised and based on the assumption that idioms act as semantic outliers and do not fit well into their contexts. Other methods tried to leverage their presumed

morphosyntactic inflexibility to distinguish them from their literal counterparts. Both methods showed some promising results on their own, but were also combined and used in a supervised fashion. Despite some good results, it is clear that the unsupervised approaches are outperformed by these supervised ones. It is reasonable to assume that modern neural architectures are able to draw on semantic and morphosyntactic properties at the same time without the need for careful feature selection and accordingly are best suited to tackle the task. Thus, in the same vein as Haagsma (2020), we employ neural architectures to tackle PIE disambiguation, but we will use word embeddings containing subword information, like fastText (Bojanowski et al., 2017), in order to try and capture morphosyntactic features as well, although – as always with these methods – it is difficult to grasp what they are learning. We will try to explore this further in Chapter 6. In addition, we will design our architecture in a way to unambiguously make sure it performs PIE disambiguation and not text classification.

Chapter 4

Corpora

This chapter is concerned with corpora that were annotated for VMWEs. Since we are interested in the identification task as well as the subtask of disambiguation, we will make the distinction between data sets containing annotation for all VMWE types and those containing only PIE annotation. The former are typically used for identification and the latter for disambiguation. Usually, this distinction is also one between *all-words* and *lexical sample* corpora. The term *all-words* alludes to the fact that all words of interest are annotated in a certain text. That means, the procedure is as follows: We compile a text and then we annotate this text, going through it word by word. On the other hand, for *lexical sample* corpora we first compile a list of target expressions and then extract text which contains those expressions. Next, only these expressions are annotated while their context is not[1]. This distinction is important, since the way we construct a corpus can completely change some of the properties. PIE corpora are often created based on the lexical sample approach precisely to change one of the key properties: the high idiomaticity rate of most PIEs. We will go further into this in Section 4.3.

The data sets which – aside from COLF-VID – are used during the experiments, i.e. the PARSEME data sets, will be covered more extensively. Furthermore, there will be a special focus on PIE corpora, as one of the main contributions of this work is the creation of a PIE corpus for German.

[1]There is of course no rule against annotating the context as well, but it would require going through the whole text looking for new types which typically is not done.

4.1 PARSEME Corpora

Arguably the most prominent VMWE corpora are the ones created for the three editions of the PARSEME shared task on VMWE identification (Savary et al., 2017; Ramisch et al., 2018, 2020) as well as the most recent release of the PARSEME corpus (Savary et al., 2023), which was the first not associated with a shared task. What makes them stand out in comparison with other MWE corpora is their scope and their homogeneity: The PARSEME corpora consist of an impressive amount of data sets from different languages which were all annotated according to the same annotation guidelines (albeit sometimes with language specific modifications). Table 4.1 shows which languages were covered in the different corpus editions. Edition 1.0 comprised data sets of 18 languages, followed by edition 1.1 with 20 languages and 1.2 with 14 languages. The number of languages varies because for every edition new languages were introduced, while others dropped out (or rather paused) due to lack of annotators updating the datasets. Version 1.3 includes all languages of the previous three editions as well as a completely new language (Serbian (SR)). Hence, it is the largest of all the PARSEME corpora with 26 languages covered.

Edition	AR	BG	CS	DE	EL	EN	ES	EU	FA	FR	GA	HE	HI	HU	IT	LT	MT	PL	PT	RO	SL	SR	SV	TR	ZH
1.0	-	✓	✓	✓	✓	-	✓	-	✓	✓	-	✓	-	✓	✓	✓	✓	✓	✓	✓	✓	-	✓	✓	-
1.1	✓	✓	✓	✓	✓	✓	✓	-	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	✓	✓	-
1.2	-	-	-	✓	✓	-	-	✓	-	✓	✓	✓	✓	-	✓	-	-	✓	✓	✓	-	-	✓	✓	✓
1.3	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Table 4.1: Languages covered in the PARSEME corpora.

The declared goal for the creation of these corpora was to model VMWEs which – up to that point – did not receive much attention in this regard, albeit being a major challenge for NLP (Savary et al., 2017). PARSEME defines MWEs the following way (Ramisch et al., 2018, pp. 223-224):

> In particular, we understand multiword expressions as expressions with at least two lexicalized components (i.e. always realised by the same lexemes), including a head word and at least one other syntactically related word. Thus, lexicalized components of MWEs must form a connected dependency graph. Such expressions must display some degree of lexical, morphological, syntactic and/or semantic idiosyncrasy, formalised by the annotation procedures.

As was mentioned earlier, a VMWE is then a “MWE whose head in a canonical form is a verb, and which functions as a verbal phrase”

(Ramisch et al., 2018, p. 224). Although similar, this definition differs from the one by Baldwin and Kim (2010) (discussed in Section 2.1) in two ways: 1. There is no mention of statistical idiomaticity because PARSEME does not consider collocations a type of MWE[2]. 2. Pragmatic idiomaticity is likewise not included in the definition, even though it is not explicitly excluded like collocations. Furthermore, the annotation guidelines state that metaphors, although closely related to idioms, are not in PARSEME's scope as they "are not sufficiently established in the common vocabulary to be considered VMWEs" (Khelil et al., 2022).

The annotation of VMWEs comes with certain challenges, a few of which (discontinuity and overlap) we already discussed in Section 3.2. Other challenges were the assignment of categories to identified VMWEs (categorization), VMWEs which can appear as one token (like German VPCs, e.g. *um|fahren*) and the fact that most text tokens will not be annotated due to the fact that they lie outside the annotation scope, i.e. they are not part of any VMWE (sporadicity) (Savary et al., 2017).

The initial annotation guidelines were adjusted and extended through several pilot annotation phases. At the end, this yielded a typology which differentiates between universal, quasi-universal, language-specific and an optional experimental category. As the name suggests, the universal category comprises VMWE categories present in all included languages. These are the categories LVC and VID. LVCs were subject to a shift in typology which took place between edition 1.0 and 1.1: While in edition 1.0 LVCs were not further divided, the annotation guidelines for edition 1.1 distinguish between LVC.full and LVC.cause. The former category pertains to LVCs whose verb almost adds nothing to the overall semantics (cf. Section 2.5.4), whereas the latter subsumes LVCs whose verbs add causative meaning and accordingly are less idiomatic (e.g. *grant rights, give a headache, provoke a reaction*).

The quasi-universal category is formed by IRVs, VPCs and MVCs and it is valid for "some language groups or languages, but not all" (Ramisch et al., 2018, p. 224). Like LVCs, VPCs are subclassified into two categories since edition 1.1. VPC.full denotes VPCs whose particle completely changes the meaning of the verb, e.g. *do in* ('kill'). VPC.semi describes VPCs whose meaning shift is partly predictable but non-spatial (e.g. *wake up*).

An initial category that fell victim to the earlier mentioned typology shift was the other (OTH) category which was meant for expressions without a unique verbal head like *drink and drive*. In later versions of

[2]Which is debatable given their properties and how much of the MWE literature treats collocations as MWEs.

the PARSEME corpora, OTH was absorbed by the VID category. Thus, with version 1.1 one of the defining criteria of VIDs became the fact that it does not fit in one of the other categories (as we discussed in Section 2.5.5).

One language-specific category was introduced for Italian (again in edition 1.1): inherently clitic verbs (LS.ICV). LS.ICVs are verbs "in which at least one non-reflexive clitic (CLI) either always accompanies a given verb or markedly changes its meaning or its subcategorization frame, e.g. *prenderle* 'take-them' ⇒ 'get beaten up' " (Ramisch et al., 2018, p. 225).

Last but not least, after edition 1.0 an experimental category in the form of inherently adpositional verbs (IAVs) was introduced which included verbs that always require an adposition or whose meaning is changed if the adposition is absent (e.g. *stand for something*). Experimental in this case means that the annotation of IAVs was optional.

To determine whether a certain candidate expression is an actual VMWE the annotation guidelines provide type- and sometimes language-specific decision trees which are supposed to maximise the determinism in an annotator's decision making process. These decision trees comprise examples and linguistic tests as shown in Figure 4.1. These linguistic tests were designed "to approximate[...] semantic non-compositionality of MWE[s] by their lexical and morphosyntactic inflexibility" (Savary et al., 2023, p. 27).

↳Apply test VPC.1 - [**PART-REDUC**: *Can the verb without the particle refer to the same event?*]
↳ **NO** ⇒ It is a **VPC.full**.
↳ **YES** ⇒ Apply test VPC.2 - [**PART-SPATIAL**: *Is the particle spatial?*]
↳ **YES** ⇒ It is **not** a VPC, exit
↳ **NO** ⇒ Apply test VPC.3 - [**PART-SPATIAL-LIT**: *Is the particle spatial in a literal reading?*]
↳ **NO** ⇒ It is a **VPC.semi**
↳ **YES** ⇒ It is **not** a VPC, exit

Figure 4.1: VPC-specific decision tree.

Tables 4.2 and 4.3 show the total statistics of version 1.0 and versions 1.1 - 1.3, respectively[3]. The two different tables reflect the earlier mentioned typology shift: OTH was absorbed by VIDs, the category LVC was split into LVC.full and LVC.cause and the category VPC into VPC.full

[3] *Sent.* stands for the number of sentences, *Tok.* for the number of tokens and *L.* for the average length of a sentence.

and VPC.semi. Furthermore, the new types IAV and LS.ICV were introduced in version 1.1. Table 4.4 shows the ranking of VMWE types with respect to the number of instances in the different PARSEME corpora. However, we have to be careful in treating this as a representative picture of VMWE distribution, since the sizes of the data sets vary greatly and the VMWE distribution can be wildly different depending on the language. To name one extreme: Farsi (FA) has 3453 annotated VMWEs, 3435 of which are LVC.fulls and only 17 are VIDs (and one single IRV). Hence, for a reliable picture on VMWE distribution, we need to look at the languages individually, but this would be outside of the scope of this work. But what these statistics do show is the pervasiveness of VMWEs as a phenomenon, since on average 28% of sentences contain a VMWE (calculated on the basis of version 1.3 as it contains the most languages).

V.	Sent.	Tok.	VMWE	ID	IReflV	LVC	OTH	VPC
1.0	274376	5439204	62218	13755	20621	17523	4802	5517

Table 4.2: PARSEME corpora total statistics (1.0).

V.	Sent.	Tok.	L.	VMWE	VID	IRV	LVC.full	LVC.cause	VPC.full	VPC.semi	IAV	MVC	LS.ICV
1.1	280838	6072331	21.6	79326	18757	16198	28190	2285	8527	1156	3049	1127	37
1.2	279785	5517910	19.7	68503	18553	11571	24574	1809	3018	4204	680	4057	37
1.3	455629	9264811	20.3	127498	26214	29062	40933	3238	9164	6443	7375	5032	37

Table 4.3: PARSEME corpora total statistics (1.1 - 1.3).

1.0	IReflV (20621)	LVC (17523)	ID (13755)	VPC (5517)	OTH (4802)					
1.1	LVC.full (28190)	VID (18757)	IRV (16198)	VPC.full (8527)	IAV (3049)	LVC.cause (28190)	VPC.semi (1156)	MVC (1127)	LS.ICV (37)	
1.2	LVC.full (24574)	VID (18553)	IRV (11571)	VPC.semi (4204)	MVC (4057)	VPC.full (3018)	LVC.cause (1809)	IAV (680)	LS.ICV (37)	
1.3	LVC.full (40933)	IRV (29062)	VID (26214)	VPC.full (9164)	IAV (7375)	VPC.semi (6443)	MVC (5032)	LVC.cause (3238)	LS.ICV (37)	

Table 4.4: Ranking of VMWE types per versions (1.0 - 1.3).

Besides the shift in typology, another important change from version 1.0 to 1.1 pertains to the format of the datasets. In 1.1, the parsemetsv format, which was used in 1.0, was merged with the CoNLL-U format, resulting in the cupt (CoNLL-U + parseme-TSV) format which can be seen in Figure 4.2. It is basically identical to the CoNLL-U format, but it contains an 11th column for the MWE annotation. This was done to align with the Universal Dependencies (UD) community[4] as their objectives of universality and diversity are the same as PARSEME's (Savary et al., 2023). To this end, PARSEME heavily relies on UDPipe (Straka, 2018)[5] to enrich the data with morphosyntactic information. The MWE annotation itself consists of identifiers which are assigned to all components of a

[4] `https://universaldependencies.org/` [Accessed: 04.06.2024]

[5] `https://lindat.mff.cuni.cz/services/udpipe/run.php` [Accessed: 04.06.2024]

MWE (*1* in the example) and a category label (*VID*). In case of overlap, we have multiple identifiers per row, separated by a semicolon.

```
# source_sent_id = . . newscrawl-508
# text = Die Rede ist von Drohnen.
1 Die     der    DET   ART   Case=Nom|... 2 det _ _ 1:VID
2 Rede    Rede   NOUN  NN    Case=Nom|... 5 nsubj _ _ 1
3 ist     sein   AUX   VVFIN Mood=Ind|... 5 cop _ _ 1
4 von     von    ADP   NE    Case=Nom|... 5 case _ _ 1
5 Drohnen Drohne NOUN  NN    Case=Nom|... 0 root _ Spa... *
6 .       .      PUNCT $.    _            5 punct _ _ *
```

Figure 4.2: The PARSEME cupt-format.

One weakness of the PARSEME corpus is the fact that the majority of datasets were not annotated by more than one person. However, a small sample size was double-annotated for every dataset, which formed the basis for the computation of the inter annotator agreement (IAA). The resulting agreement scores vary greatly with regard to the different languages. This is something that might "correlate with the results obtained by participants: the lower the IAA for a given language (i.e. the more difficult the task is for humans), the lower the results of automatic MWE identification" (Ramisch et al., 2018, p. 228). But at the same time it is unclear how reliable the agreement scores are given the small sample size.

In edition 1.0 the data was split into a train and a test set. This changed with edition 1.1 which additionally introduced a dev set. Edition 1.2 of the ST – and the corpus along with it – saw again a bigger change as the focus shifted towards the identification of VMWE types unseen during training. To this end, the split was adjusted in a way to ensure the dev and test set contained a certain number of unseen VMWEs.

We will come back to the PARSEME corpus in Section 6.1, when we present our VMWE identification experiments.

4.2 STREUSLE Corpus

The PARSEME corpus was not the first to cover all types of VMWEs. The aim of STREUSLE (Schneider et al., 2014b; Schneider and Smith, 2015) – a 55,000 word social web corpus for English – was to cover all types of MWEs, which therefore included verbal ones. The annotators were

"simply shown one sentence at a time and asked to mark all combinations that they believe are multiword expressions". MWEs, in the context of STREUSLE, then are defined as "a group of tokens in a sentence that cohere more strongly than ordinary syntactic combinations: that is, they are idiosyncratic in form, function, or frequency" (Schneider et al., 2014b). Thus, the notion of MWEs is not as strict as for PARSEME, since collocations are explicitly included ("idiosyncratic in [...] frequency"). The following shows an example with two annotated MWE instances from STREUSLE which are marked with underscores (_) and tildes (∼), respectively:

(1) They eventually turned_ it _over to a collection agency and now will not even discuss∼the∼matter.

The MWE type *discuss the matter* would not be annotated as a MWE in PARSEME, as it is only statistically idiomatic.[6]. However, STREUSLE distinguishes between *strong* and *weak* MWEs. The former category comprises expressions that lean towards non-compositionality, while the latter is reserved for compositional but statistically idiomatic expressions. In (1), *turn over* constitutes a strong MWE and *discuss the matter* a weak one (the underscore and tilde are used to mark these cases, respectively).

Originally, the annotations did not include MWE-specific labels, but in the current version of STREUSLE categories from PARSEME 1.1 were incorporated. Thus, at least in theory, it should be compatible with the English PARSEME dateset if weak MWEs were to be removed and if we disregard the fact that overlap is handled differently in STREUSLE than in PARSEME (STREUSLE uses the the IOB annotation scheme).

4.3 PIE Corpora

In Section 3.3, we discussed PIE disambiguation as well as approaches suitable to tackle this task. In this section, we present corpora that can be used to train and/or evaluate classifiers capable of distinguishing between MWEs and their literal counterparts. To this end, these corpora not only need to mark PIE instances, but also annotate whether they are used literally or idiomatically. But we will also see how this distinction is not always binary and that we sometimes (albeit not very often) need to

[6]And even that can be questioned when considering that tests for anti-collocations do not yield negative results. E.g. *argue to matter, talk through the matter* or *discuss the topic* seem also perfectly fine. Thus, even from a statistical standpoint, *discuss the matter* seems hardly idiosyncratic.

accommodate for other readings as well.

We compare the presented corpora with respect to key properties like size, number of PIE types, labels used for annotation, etc. As the creation of a PIE corpus is one of the main contributions of this thesis, we will go into more detail at this point in order to be able to later relate our corpus to the work presented here. For easier reference to the original papers we will adapt their terminology, but keep in mind that we would subsume all the expressions discussed in the following under the term PIE.

VNC-Tokens Dataset (2008)

One of the first PIE corpora was the *VNC-Tokens Dataset* (Cook et al., 2008). Like all data presented in this section, it is a lexical sample corpus which means that instances from a pre-chosen set of expressions are extracted from a certain source. In this case, the data set consists of 2894 instances of 53 idiomatic verb noun combination (VNC)[7] types (or their respective literal counterparts). The starting point was a VNC list compiled by Fazly and Stevenson (2006) which was filtered by the authors based on whether the idiomatic meanings were familiar to the annotators and a literal reading of an expression was thought possible. The British National Corpus[8] (BNC) was then searched for verbs with a noun in direct object relation which matched any of these types. For each type, between 20 and 100 examples were collected. The two authors of the paper both independently annotated all examples and attained an agreement rate of 88% and a Kappa of 0.76. Disagreements were discussed among the judges until a joint verdict was reached. The labels used were LITERAL, IDIOMATIC and UNKNOWN. The latter was applied when a judge could not decide on one of the other two labels. The idiomaticity rate, i.e. the ratio of examples labeled as IDIOMATIC, is 67.69% and thus quite low compared to the numbers reported in Savary et al. (2019) where the lowest reported rate is 96%. But we have to keep in mind that we cannot really compare these numbers as the PARSEME corpora are all-words corpora, while the VNC-tokens data was chosen according to its potential to contain a high number of literal readings, so its low idiomaticity rate is not that surprising. What is a bit unusual about this data set is that 13.84% of instances were labeled as UNKNOWN. This is a very large amount, especially if we consider that it is quite close to the number of literal examples (18.47%). The limitation to one sentence only

[7] In an earlier chapter we introduced these as verb noun idiomatic combinations (VNICs).

[8] `http://www.natcorp.ox.ac.uk` [Accessed: 04.06.2024]

during annotation could be an explanation. However, as we we will see, other corpora with the same amount of context have a much lower ratio of undecidable cases. What might be a better explanation is that some types make it much harder than others to infer their correct reading in any context and that the VNC data set contains a lot of them. We will discuss this in more detail at a later point.

Hashimoto and Kawahara (2008)

The largest PIE corpus in terms of number of sentences was created by Hashimoto and Kawahara (2008). It contains 102,846 instances of 146 Japanese idioms[9]. The authors started with a list of Japanese idioms compiled by Sato (2007) and only chose the ambiguous ones. To decide which idioms potentially have literal meaning and which do not, two groups of two native speakers were asked to give their judgement. Based on those judgements a Kappa score of 0.66 was computed, which only amounts to moderate agreement. It is a somewhat interesting result that the intuition of native speakers on this matter is not more in tune. Usage examples for the idiom types were drawn from a web corpus, albeit it is not mentioned which specifically. Since all sentences without enough context were discarded, the set of annotation labels only consisted of the tags LITERAL and IDIOMATIC. A sample of 1421 examples were annotated by multiple judges with a Kappa score of 0.85.

IDIX (2010)

Like the VNC-tokens data set, the IDIX (IDioms In conteXt) corpus by Sporleder et al. (2010) is also based on the BNC. Nevertheless, it differs in a few regards from the former. As the name suggests, the usage examples for the idiom types were not annotated in single sentences but in larger contexts of two paragraphs before and after the instance. Another difference is the number of labels applied during annotation. While we find equivalents for the labels LITERAL, NON-LITERAL and UNCLEAR in the data sets presented before, the labels BOTH, META-LINGUISTICALLY and EMBEDDED are new. The label BOTH is applied to cases where the literal and idiomatic reading are active at the same time. This usually happens in combination with word play. The label META-LINGUISTICALLY is used for examples where the linguistic nature of the target expression is discussed as in (2):

[9]The authors actually never state the exact type, but the 90 examples shown in the paper seem to be VIDs.

(2) The idiom *kick the bucket* is semantically idiomatic.

The label EMBEDDED comes into use if the instance is embedded in a larger metaphorical context. Due to the figurative nature of a lot of idioms, this happens quite regularly. For example as in (3):

(3) She is a fighter who does not pull any punches.

Unsurprisingly, few examples were annotated as BOTH and UNCLEAR, only accounting for 0.69% and 4.15% of the instances, respectively. But the distribution of the labels LITERAL and NON-LITERAL on the other hand is quite unexpected: only 45.61% of instances are labeled as non-literal, while 49.4% where judged to be literal, making it one of the few corpora with an idiomaticity rate of below 50%. That is remarkable given that idiomaticity rates usually are quite high, even for PIE data sets tailored to keep it low. We can only speculate on why that is. The most obvious reason is that the authors did a good job in anticipating which PIE types have a high literality rate.

The annotation quality was determined based on a subset of 1,136 examples which were annotated by two judges who reached an agreement of 93.19% and a Kappa score of 0.87. According to the paper, the total number of instances should be 5836. But as stated in a footnote in Haagsma et al. (2019) where a personal conversation with one of the authors of the IDIX paper is cited, the actual number is much lower with only 4022 instances. Also, the number of types is not 75, as claimed in the paper, but 52. This of course means that the numbers describing the label distribution also have to be taken with a grain of salt, although we assume the tendency at least should be the similar[10].

PNV Dataset (2010)

Fritzinger et al. (2010) were the first to create a German PIE corpus. It consists of 9700 usage examples of 77 preposition-noun-verb (PNV) triplets, a pattern that is very common for German VIDs (e.g. *im Raum stehen* ('stand in the room'⇒'be raised as a problem')). The instances were drawn from the FAZ (Frankfurter Allgemeine Zeitung)[11] and EUROPARL[12]. Three different labels were applied during annotation: LITERAL, IDIOMATIC and AMBIGUOUS, the latter one being used for examples where the context was not enough to decide on the reading. Given that

[10] We would have established the real numbers ourselves, but the IDIX corpus does not seem to be publicly available and multiple requests for access were ignored.

[11] A German newspaper.

[12] `https://www.statmt.org/europarl/` [Accessed: 04.06.2024]

political debates are a peculiar form of speech, the authors computed the label distribution separately for the FAZ and the EUROPARL part. Confirming the intuition of the authors that PNVs are used mainly idiomatically in political speech, the EUROPARL part has an idiomaticity rate of 98.49, while for the FAZ part the rate is *only* 93.63. The combined idiomaticity rate of 95.15% stands in stark contrast to that of the IDIX corpus and shows how much the choice of candidate expressions influences the nature of the resulting data set. It seems the authors have chosen PNVs which are predominantly used idiomatically. This also becomes evident when comparing the ratios of ambiguous instances in the different corpora. Only 0.9% of examples in the PNV data set received the AMBIGUOUS label, whereas 13.84% of all cases in the VNC-tokens data set were annotated as UNKNOWN, although the conditions during annotation were the same in that the annotators had only access to one sentence. We suspect this considerable difference is also due to the choice of candidate expressions. The language and type of source corpus could of course be another factor, but in a later chapter, we will present a German newspaper corpus with a label distribution more similar to the IDIX and VNC-tokens corpora than to the PNV data set (i.e. lower idiomaticity rate and higher rate of ambiguous instances when only considering one sentence).

SemEval 5b (2013)

During the SemEval-2013 shared task on the evaluation of phrasal semantics (task 5), two PIE data sets were created: one for English and one for German (Korkontzelos et al., 2013). In contrast to the corpora presented above, not only verbal types were allowed as candidates, but also nominal ones (e.g. *bread and butter*) and adverbials (e.g. *through the roof*). The English data set consists of 4350 instances of 65 types[13] and the German one of 3408 instances of 41 types. Along with the sentences containing the examples, two surrounding sentences are provided, usually the preceding and succeeding one. According to the paper, each usage consists of 5 sentences, but at least the German data contradicts this claim. Also, during annotation the labels LITERAL, FIGURATIVELY, BOTH and IMPOSSIBLE were used according to the authors. But this only seems to be the case for English as the German data solely contains the tags LITERAL and FIGURATIVELY. The data was annotated by three crowdworkers independently who reached agreements of 90-94%. 5% of instances

[13]These numbers are taken from Haagsma et al. (2019), since they were not provided in the paper and we were not able to obtain the English data set.

with low agreement or marked as *impossible* were removed.

PV Dataset (2016)

Köper and Schulte im Walde (2016) focused on a verbal type of PIE that usually does not get much attention: particle verbs (PVs). When we talk about verbal PIEs we usually think about the prototypical expressions of the form V+NP (*spill the beans*) and V+PP (*jump on the bandwagon*). But some PVs can also have a literal and and a non-literal reading:

(4) Ich **säge** den Ast **ab**.
I **saw** the branch **off**.
I saw off the branch.

(5) Der Minister wurde **abgesägt**.
The minister was **sawn off**.
The minister was removed.

Example (4) contains the literal version of the German PV *absägen* ('saw off'), Example (5) the non-literal reading ('get rid off').

The corpus consists of 165 German PVs across 10 particles in 6436 usage cases drawn from the German web corpus DECOW14AX (Schäfer and Bildhauer, 2012). What sets this work even more apart from the other data sets presented here, is the annotation scheme. To account for the fact that idiomaticity is not a binary property, but falls on a continuum, the judges were given a six-point scale [0, 5]. However, this proved to be problematic, since the agreement of the three annotators was only 43% and the Kappa score only 0.35. Therefore the scale was divided into two disjunctive ranges, where the range from 0-2 was interpreted as literal and the range from 3-5 as non-literal. This increased the agreement to 79% and the Kappa score to 0.70. All disagreements were discarded from the data set based on this binary distinction. This resulted in 4,174 literal and 2262 non-literal uses and an idiomaticity rate of only 35.15% - by far the lowest of all the corpora presented here. This raises the question whether (German) PVs in general have a lower idiomaticity rate or if the authors did a very good job at selecting those PVs that have a particularly high number of literal instances.

Horbach et al. (2016)

Another corpus with a focus on a special type of PIE was created by Horbach et al. (2016). Their object of study are infinitive-verb compounds

(IVCs), i.e. verb groups consisting of an inflected head verb and an infinitive modifier:

(6) Ich **bleibe** auf meinem Stuhl **sitzen**.
I remain on my chair be seated.
I remain seated on my chair. LITERAL

(7) Er **bleibt** am Ende des Jahres **sitzen**.
He remains at end of year be seated.
He will be retained at the end of the year. IDIOMATIC

In (6), we have the literal reading of *sitzen bleiben* ('remain seated'⇒'to be retained') where someone actually remains seated on a chair. In (7), we have the idiomatic reading, where the physical act of remaining at the same place stands for retention in school. Interestingly, IVCs add a new dimension to the set of properties of PIEs, since, in German, the literal usage of an IVC has to be written as two separate words, while the idiomatic one can also be written together. Hence, the spelling in (8) would be correct, but not in (9).

(8) Sie wird auf ihrem Stuhl **sitzen bleiben**.
She will on her chair be seated remain.
She will remain seated on her chair. LITERAL

(9) *Sie wird auf ihrem **Stuhl sitzenleiben**.
She will on her chair be seated remain.
She will remain seated on her chair. LITERAL

So in theory, it could be a sure sign for idiomatic usage if the infinitive and the inflected verb are written together, but in practice, it is questionable if many German speakers have this kind of knowledge. And indeed, this corpus was created with the aim to train tools which monitor if the spelling rules are applied. To this end, 6000 instances of 6 IVC types were extracted from the Wahrig corpus (Krome, 2017) and annotated. The annotation was conducted by two experts who were presented with the sentence containing the instance and one additional sentence to the left and right. IVCs proved to be a challenging type as some types had multiple idiomatic and even multiple literal meanings. Disagreements among the judges were discussed within a group, where for 79% of the unclear cases an agreement could be reached. The Kappa score was computed individually for every type and spelling and lay between 0.6

and 0.9. The labels applied during annotation were LITERAL, IDIOMATIC and ? (for unclear). With an idiomaticity rate of 56.15 it has one of the lowest of all data sets presented in this section.

MAGPIE (2020)

The largest data set with respect to PIE types was created by Haagsma et al. (2020). The English MAGPIE[14] corpus contains an impressive 1,756 different PIE types and 56622 usages in context. Since it was the declared goal to cover as much types as possible, a maximum of 200 instances per type were extracted from the BNC and the PMB. Like the SemEval data sets, MAGPIE was annotated by crowdworkers. They marked instances with the labels LITERAL, IDIOMATIC, UNCLEAR and NON-STANDARD USAGE. The latter was used when an example could "be interpreted but simply [did] not fit the binary sense distinction" (Haagsma et al., 2020, p. 281), i.e. it is the equivalent to the BOTH tag in other data sets. During annotation extensive quality checks were conducted to filter out untrustworthy crowdworkers. The remaining group of 54 workers was still large enough to collect so many judgements till a threshold of 70% agreement was reached. 70.66% of instances in the MAGPIE corpus were labeled as idiomatic and 28.55% as literal. Only 0.01% of examples received the UNCLEAR label which is the the lowest ratio of all corpora presented in this section[15]. An obvious explanation for this is that the majority of PIE types is not ambiguous. This is confirmed by the authors who found only 81 PIE types they considered to be truly ambiguous, i.e. they have an idiomaticity rate between 40% and 60%. And since MAGPIE – in contrast to other corpora – was not tailored to contain as many literal readings as possible but a maximum amount of types, it makes sense the number of ambiguous examples is so low.

Conclusion

Table 4.5 represents a compact overview over the corpora presented above. If the corpus does not have a specific name, we give the name of the main author. For the IDIX corpus, the correct numbers are taken from Haagsma et al. (2019), the numbers from the original papers are given in parentheses.

[14] While the PIE part is obvious, it is never stated in the paper what the MAG in MAGPIE stands for.

[15] Some corpora do not contain any ambiguous examples at all, but only because they were filtered out.

Corpus	Size	#PIE	Lang.	Labels	Agr./ Kappa	I%	Source	Form Type	Avail.
VNC-tokens (2008)	2984	53	EN	literal, idiomatic, unknown	88/0.76	67.69	BNC	V+NP	yes
Hashimoto et al. (2008)	102,846	146	JP	literal, idiomatic	-/0.85	N.A.	N.A.	N.A.	no
IDIX (2010)	4022 (5836)	52 (75)	EN	literal, non-literal, unclear, both, meta-ling. embedded	93.19/0.87	45.61	BNC	V+NP/PP	no
PNV data set (2010)	9700	77	DE	literal, idiomatic, ambiguous	97.9	95.15	FAZ, EUROPARL	V+PP	no
SemEval 5b (2013)	4350	65	EN	figuratively, literal, both	90-94%/-	N.A.	ukWac	all	on request
SemEval 5b (2013)	3408	41	DE	figuratively, literal	N.A.	N.A.	deWac	all	on request
PV data set (2016)	6436	159	DE	0-2 (literal) 3-5 (non-literal)	79/0.7	35.15	DECOW14AX	particle verbs	yes
Horbach et al. (2016)	6000	6	DE	literal, idiomatic, ?	$0.6 < 0.9$	56.15	Wahrig	infinitv-verb compounds	yes
MAGPIE (2020)	56622	1756	EN	idiomatic, literal, unclear, non-standard usage		70.66	BNC, PMB	all	yes

Table 4.5: PIE corpora overview.

What all corpora have in common, is a relatively high agreement or Kappa score between the judges. This suggests that although some PIE types are hard to annotate because it is unclear where on the literal–idiomatic spectrum they fall, the annotation seems to be a an easy task overall. Also, the label sets are very similar as all but one contain tags for the literal and idiomatic reading as well as an option for unclear cases. For some data sets (Hashimoto and Kawahara (2008), SemEval 5b), the labels for the latter are not shown in table 4.5 despite being used during annotation, since instances that received the UNCLEAR label were removed from those corpora. IDIX and SemEval 5b (EN) contain labels for cases in which the literal and idiomatic reading are active at the same time, while IDIX is the only corpus with an explicit marker for meta-linguistic material. However, MAGPIE has an underspecified label for non-standard usage of PIEs, comprising labels such as BOTH and META-LINGUISTIC. Another commonality is how most instances were annotated as either idiomatic or literal – all the other labels were rarely applied. As discussed above, the only outlier is the VNC-tokens data set which contains an unusual high number of unclear cases. Hence, it is safe to say that PIE disambiguation largely remains a binary classification task.

Where the corpora differ greatly is with regard to the idiomaticity rate which ranges from 35.15% (PV data set) to 95.15% (PNV data set), a clear indicator of how much the choice of PIE types influences this

property. On top of that, it emphasizes that lexical sample corpora do not represent the real distribution of literal and idiomatic usages of PIE types.

In the next chapter, we will introduce our own PIE corpus and discuss how it relates to the works presented above.

PART II

Corpus Construction

Chapter 5

COLF-VID

In this work, we address both VMWE identification and PIE disambiguation. In theory, those two tasks should go hand in hand, since the identification of a VMWE instance implies disambiguation from potential literal counterparts. E.g. for a tagger to label *spilled the beans* in Example (1) as a VMWE, but not in Example (2), it has to perform disambiguation first.

(1) After a long interrogation, they finally **spilled the beans**.

(2) When Kevin entered the office, he dropped the pot he was carrying and spilled the beans all over the floor.

So ideally, we would not need dedicated PIE corpora. But – as was discussed extensively in earlier chapters – for most languages there do not exist data sets with an adequate number of PIE instances to counter their usually high idiomaticity rates. VMWE corpora like the PARSEME data sets do not contain enough literal occurrences to train and evaluate a classifier capable of disambiguation in an effective way. To alleviate this issue for German, we built a Corpus of Literal and Figurative[1] Readings of Verbal Idioms[2] (COLF-VID). Please note that there are two different versions of COLF-VID: 1.0 and 1.1. COLF-VID 1.1 is a slightly cleaned up version of 1.0. However, we used version 1.0 for the original experiments and kept on doing so for comparison purposes, so every time we speak about COLF-VID, we mean version 1.0 (even if not explicitly mentioned).

In this chapter we discuss various aspects of (PIE) corpus construction, i.e. the annotation guidelines, how we decided on the type of the corpus (lexical sample vs. all words), how VID types were chosen, etc. Furthermore, we will discuss which issues arose during annotation and a

[1] Here we use *figurative* and *idiomatic* synonymously.

[2] We had not yet adopted the PIE terminology when we named the corpus.

qualitative study on the flexibility of the annotated VID types will be conducted and put into context of what the literature usually claims about VIDs.

Parts of this chapter, namely Section 5.1, heavily rely on our paper *Supervised Disambiguation of German Verbal Idioms with a BiLSTM Architecture* (Ehren et al., 2020).

5.1 COLF-VID

The motivation for the creation of this corpus was the lack of German data to properly train a supervised system capable of disambiguating PIEs (cf. Chapter 4). In the study of Savary et al. (2019), which included five different languages, the German data set was found to have an idiomaticity rate of 98%[3] and even the German SemEval 5b data set (Korkontzelos et al., 2013) which was specifically built for a shared task on the disambiguation of PIEs, has an idiomaticity rate of 93.58%. This begs the question: How do we build a corpus with a sufficient amount of literal instances? It seems clear that we do not get enough data if we just annotate random text as was the case for the PARSEME corpora[4], so this rules out an all-words corpus. However, the German SemEval 5b data set shows that even for a lexical sample corpus we cannot just randomly select some PIE types, since most types (at least for German) seem to have a high idiomaticity rate. Thus, when compiling the list of VID types we chose them according to the perceived potential of delivering a large number of literal instances. These types were partly found by consulting the lexicon *Duden - Redewendungen: Wörterbuch der deutschen Idiomatik*[5] and partly by discussing suitable candidates among ourselves. Table 5.1 constitutes the complete list along with the literal and idiomatic meaning(s). What stands out is that there can be not only ambiguity at the PIE level, but also at the VID level: A VID type can have multiple different, albeit often related, meanings (albeit not very often). E.g. *auf der Straße stehen* ('stand on the street') can have the meanings 'to be unemployed' or 'to be homeless':

(3) Die Abteilung wurde geschlossen und etliche
The department was closed and a large number of
Mitarbeiter **standen auf der Straße**.
employees stood on the street.

[3]Polish had the lowest rate of the five languages with 96%.
[4]Unless we have huge amounts of data, of course.
[5]Transl: *Duden - Idioms: Lexicon of German Idiomatics* (ISBN: 9783411041152)

'The department was closed and a large number of employees was out of the job.'

(4) Er verlor seine Wohnung und **stand** auf einmal
He lost his apartment and stood suddenly
auf der Straße.
on the street.
'He lost his apartment and was homeless all of a sudden.'

The PIE types come in one of two different patterns: 26 types consist of a verb and a PP, while the rest is a combination of a verb and a noun in direct object position. Judging by the literature, the latter pattern received a lot of attention in recent years, as it seems to be a much more common pattern for English PIEs than the former. In contrast, the PP + V pattern seems to be much more common for German (Fritzinger et al., 2010).

5.1.1 Extraction

Sentences containing instances of the pre-selected PIE types were extracted from the TüPP-D/Z[6] corpus which consists of articles of the German newspaper *Die Tageszeitung* (*taz*) from the years 1986 to 1999. With only 204 million tokens, it is a rather small corpus by today's standards, but the fact that it contains whole articles made it attractive for our purposes, since the context needed for disambiguation sometimes exceeds sentence boundaries. The corpus is partially parsed and contains information concerning sentence structure, topological fields and chunks.

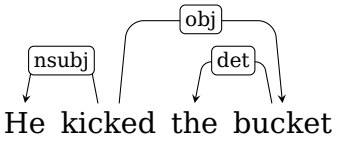
The extraction was conducted by compiling a list of the pre-selected PIE types whose components were represented in lemma form. Subsequently, a very basic string-based extraction method was employed by extracting all the sentences in TüPP-D/Z containing the lemmas of a PIE type (making use of the existing lemmatization). As described in (Haagsma, 2020, p. 65), there are more sophisticated extraction methods which are based on dependency parses:

> 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.

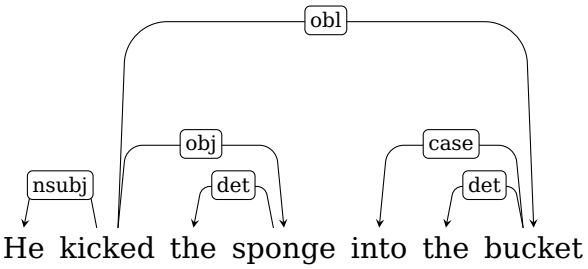
[6] http://hdl.handle.net/11858/00-1778-0000-0007-5E99-D [Accessed: 27.05.24]

PIE type	Literal	Idiomatic
am Boden liegen	lay on the ground	to be in a bad state
an Glanz verlieren	lose luster	to loose attractiveness or beauty
an Land ziehen	pull on land	to secure a good deal/ purchase sth. valuable
am Pranger stehen	stand in the pillory	to be publicly criticized
den Atem anhalten	hold one's breath	to have doubts sth. will end well*
auf dem Abstellgleis stehen	stand on the siding	to be robbed of one's influence
auf den Arm nehmen	pick sb. up	tease so.
auf der Ersatzbank sitzen	sit on the substitutes bench	to be robbed of one's influence
auf der Straße stehen	stand on the street	to be unemployed/to be homeless
auf der Strecke bleiben	stay on the track*	fail/sth. gets thwarted
auf dem Tisch liegen	lay on the table	to be available/to be topic
auf den Zug aufspringen	jump on the train	follow a trend
eine Brücke bauen	build a bridge	establish a connection/find a compromise
die Fäden ziehen	pull strings	(secretly) influence sth.
im Blut haben	have in one's blood	to have an innate talent
in den Keller gehen	go in the basement	to sink (fast and) low
in der Luft hängen	hang in the air	to be without support/ face an uncertain future
im Regen stehen	stand in the rain	to be marooned/ face a difficult situation
ins Rennen gehen	enter a race	begin sth.
in eine Sackgasse geraten	to get into a dead end	to get in an alternativeless situation
im Schatten stehen	stand in the shadow	to receive no attention
in Schieflage geraten	get into imbalance	to be in an alarming situaton
ins Wasser fallen	fall into the water	to be canceled
Luft holen	take a breath	take a break
mit dem Feuer spielen	play with fire	to recklessly take a risk
einen Nerv treffen	hit a nerve	achieve a strong effect
die Notbremse ziehen	pull the emergency brake	terminate a situation
eine Rechnung begleichen	settle a bill	fulfill an obligation/to revenge
von Bord gehen	to go off board	leave a project or a job
vor der Tür stehen	stand in front of the door	to be imminent
ein Zelt aufschlagen	pitch a tent	to settle
über Bord gehen	go overboard	to be abandoned
über Bord werfen	throw overboard	to abandon
über die Bühne gehen	walk on the stage	proceed in a certain way

Table 5.1: COLF-VID PIE Types.



(a) PIE instance.



(b) Accidental co-occurrence.

Figure 5.1: Parsing-based extraction.

Although parsing-based methods should be much more precise, we settled for the string-based method in order to maximize recall because by not relying on parsers we were not subject to error propagation resulting from erroneous parses. Naturally, this gave us way more sentences than we needed because of coincidental matches that contained the lemmas but without them forming a PIE, i.e. the sentence did not contain a subtree that matched the PIE dependencies. For example, both sentences in 5.1 would have been extracted in our, but only 5.1(a) in the parsing-based approach because the dependency relation between *kick* and *bucket* in 5.1(b) is *obl* (for oblique argument) instead of *obj*. If we had used a larger corpus, this approach would of course have been preferable, but, as we will see shortly, due to the small size of TüPP-D/Z even with our recall-focused method there were scarcity-issues for some PIE types which would have been exacerbated even further by potential parsing errors.

Erroneously extracted sentences were dismissed in a manual manner which was feasible because of the small size of the corpus and the correspondingly small number of extracted sentences.

The sixth column (*Total*) in Table 5.2 shows the total number of instances per PIE type that remained after the manual filtering step. We can see right away that there is a large large variance with regard to the number of instances per type, with as little as 5 instances for *am Pranger stehen* ('stand in the pillory'⇒'being criticised in public') and as much as 951 for *auf dem Tisch liegen* ('to lay on the table'⇒'to be available').

At this point, there is one more important issue to address regarding

VID type	Lit.	Idiom.	Und.	Both	Total	I%
am Boden liegen	35	11	0	1	47	23.4
an Glanz verlieren	0	15	1	0	16	93.75
an Land ziehen	25	235	0	0	260	90.38
am Pranger stehen	0	5	0	0	5	100.0
den Atem anhalten	10	30	0	0	40	75.0
auf dem Abstellgleis stehen	15	11	0	0	26	42.31
auf den Arm nehmen	39	50	0	0	89	42.31
auf der Ersatzbank sitzen	16	5	0	0	21	23.81
auf der Straße stehen	93	156	1	0	250	62.4
auf der Strecke bleiben	4	616	1	0	621	99.19
auf dem Tisch liegen	262	678	10	1	951	71.29
auf den Zug aufspringen	5	121	0	0	126	96.03
eine Brücke bauen	109	238	1	0	348	68.39
die Fäden ziehen	9	164	0	0	173	94.8
im Blut haben	29	7	0	0	36	19.44
in den Keller gehen	34	91	0	0	125	72.8
in der Luft hängen	28	256	0	0	284	90.14
im Regen stehen	69	302	4	4	379	79.68
ins Rennen gehen	11	51	0	0	62	82.26
in eine Sackgasse geraten	2	99	0	0	101	98.02
im Schatten stehen	7	52	0	1	60	86.67
in Schieflage geraten	3	40	1	0	44	90.91
ins Wasser fallen	67	186	0	0	253	73.52
Luft holen	100	66	4	0	170	38.82
mit dem Feuer spielen	9	74	2	0	85	87.06
einen Nerv treffen	1	284	0	0	285	99.65
die Notbremse ziehen	51	275	0	0	326	84.36
eine Rechnung begleichen	89	162	0	0	251	64.54
von Bord gehen	45	48	0	0	93	51.61
vor der Tür stehen	189	409	1	1	600	68.17
ein Zelt aufschlagen	53	41	6	0	100	41.0
über Bord gehen	62	52	1	0	115	45.22
über Bord werfen	54	389	0	0	443	87.81
über die Bühne gehen	2	198	0	0	200	99.0
Total	1527	5417	33	8	6985	77.55

Table 5.2: COLF-VID annotation statistics.

the lexical flexibility of an expression. Although we use the canonical form of a PIE when citing it, during extraction, we did not exclude patterns exhibiting lexical variation in the determiner spot. Consider for example the following two instances of *auf dem Tisch liegen* pulled from COLF-VID:

(5) [...] das Mißtrauensvotum **lag auf** dem **Tisch**.
[...] the vote of no confidence lay on the table.
'A vote of no confidence was up for debate.'

(6) [...] vorgestern **lagen** endgültig die Fakten **auf** seinem **Tisch**.
[...] the day before yesterday lay finally the facts on his table.
'The day before yesterday, he finally knew the facts.'

In Example (5), the VID *auf dem Tisch liegen* occurs in its canonical form with the definite article *dem* (*the*). But in (6), it appears with the possessive pronoun *seinem* (his) instead of the definite article. This goes to establish that *auf dem Tisch liegen* can have its idiomatic reading, even if it does not appear in its canonical form. But we would not only exclude some idiomatic instances if we prohibited this kind of flexibility: The decision whether to include or exclude non-canonical forms potentially effects the idiomaticity rate of a PIE type in the resulting corpus. For example, let us assume that for *auf dem Tisch liegen* the literal instances exhibit a larger degree of lexical flexibility in the determiner spot, while the idiomatic occurrences usually are accompanied by the definite article. If we extracted only sentences satisfying the canonical pattern, we would ignore some literal occurrences and this would result in a higher idiomaticity rate. For example, instances such as the following would not have been included in COLF-VID:

(7) Auf seinem Tisch lag ein Buch [...].
On his table lay a book... [].

There are PIE types like *einen Korb geben*[7] ('give a basket'⇒'refuse') where we can be (relatively) sure that they are not subject to this kind of variation as they loose their idiomatic meaning when the determiner is replaced by something else, but we need to be careful with such assumptions or else we might fail to capture the real distribution of literal and idiomatic instances of a PIE type in a given corpus. The same goes for

[7]Not part of the original COLF-VID corpus, but it is included in the shared task corpus described in the next chapter.

morphological flexibility. For example, the noun in the VID *eine Brücke bauen* ('build a bridge'⇒'make connections') can be singular or plural without it loosing its idiomatic meaning.

This raises the question where to draw the line with respect to the flexibility of a PIE type. What degree of flexibility do we allow until we consider an instance to be of a different type? The answer for our case is that we allow for morphological flexibility and lexical variation concerning the determiner but not the preposition (if present), noun or the verb. E.g. *über Bord gehen* ('go overboard'⇒'to be abandoned') and *über Bord werfen* ('throw overboard'⇒'abandon') are considered different types despite their strong similarity in literal and idiomatic meaning because the verbs are different. In the same vein, *über Bord gehen* and *von Bord gehen* ('go off board'⇒'leave a project') are considered different because of the different prepositions and the resulting significant change in meaning:

(8) Maria **ging** nach 30 Jahren im Unternehmen **von Bord**.
Maria went after 30 Years in the company off

board.
'After 30 years, Maria left the company'.

(9) Alle Hemmungen **gingen über Bord**.
All inhibitions went over board.
'All inhibitions were lost'.

Still, the question of how much variation is allowed for a given PIE type is not always easy to answer and there are borderline cases where different arguments can be made. But regardless of whether one allows for minimum or maximum variation, it is important to keep in mind how it affects the idiomaticity rate of the final corpus.

5.1.2 Annotation Scheme

The goal of the annotation was to label every PIE instance whether it was used literally or idiomatically. Initially, the plan was to use a binary scheme with only the labels IDIOMATIC and LITERAL. But during the annotation it soon became clear that these two did not encompass all possibilities, so the labels UNDECIDABLE and BOTH were added to the set. The following gives a short overview of how these labels were supposed to be applied:

- **LITERAL**: This label was assigned to instances which were perceived as compositional, i.e. the overall meaning of an expression is deter-

mined by the most basic meanings of its components without any form of figuration involved. In other words, we equate literality with compositionality.

- **IDIOMATIC**: This label was reserved for instances which show a lack of compositionality, i.e. whose overall meanings are not an amalgamation of the meanings of its components. Thus, this label describes semantic idiomaticity.
- **UNDECIDABLE**: In some cases, it was not possible to decide on one of the two readings described above, even with the whole context available.
- **BOTH**: This label was assigned to instances were the literal and the idiomatic reading are active at the same time.

For better illustration, the following shows examples for the four different labels, extracted from TüPP-D/Z:

(10) Bundesbahn will die **Notbremse ziehen**.
Federal railway wants the **emergency brake pull**.
'Federal railway wants to pull the emergency brake.' **idiomatic**

(11) Der Zug war kurz nach seiner Abfahrt durch das **Ziehen**
The train was shortly after its departure by the **pulling**
der **Notbremse** gestoppt worden.
of **emergency brake** stopped was.
'The train was stopped shortly after its departure by the pulling of the emergency brake.' **literal**

(12) Wer möchte, könnte ihm den **Kopf waschen**, ihm mal **auf** den
Who wants, could him the **head wash**, him once **on** the
Zahn fühlen oder ihn gar **auf** den **Arm nehmen** [...].
tooth feel or him even **on** the **arm take** [...].
'Whoever wants to, can dress him down, see what he's made of or even pull his leg.' **both**

(13) Wochenlang **lag** das Band mit den Gewaltphantasien der
For weeks **lay** the video with the violent fantasies of the
Rekruten **auf dem Tisch**.
recruits **on the table**.
'For weeks, the video with the violent fantasies of the recruits lay on the table.´ **undecided**

Example (10) contains an instance of the PIE type *die Notbremse ziehen* ('pull the emergency brake'⇒'terminate a situation') that was labeled as idiomatic. We can infer this from the institution *Federal railway* being the subject of *ziehen* ('pull') instead of an animate agent that could actually perform the physical action of pulling an emergency brake. In (11), the context informs us that the same expression is used literally. Although strictly speaking, we cannot be 100% sure because there is always the possibility that the expression is embedded in a figurative context and *Zug* ('train') and *Abfahrt* ('departure') are metaphors of some sort, but we would argue the literal reading is far more likely.

Example (12) is especially remarkable as it includes instances of three different PIE types which all describe a bodily action when used literally: *den Kopf waschen* ('wash sb.'s head'⇒'scold sb.'), *jmdm. auf den Zahn fühlen* ('feel sb.'s tooth'⇒'test sb. out') and *auf den Arm nehmen* ('take sb. in their arms'⇒'make fun of sb.'). And since they are used in a report about the demolition of a statue of a historical personality one could argue that both readings are active at the same time, since all the actions denoted by the PIEs could in theory be done to the statue. A good indicator for the appropriateness of the label is the fact that one would loose part of the meaning if the sentence were translated considering only the literal or only the idiomatic senses. The only faithful translation should include analogous English expressions, i.e. PIEs with the same idiomatic meaning that also describe bodily actions.

In Example (11), we have an instance of the PIE type *auf dem Tisch liegen* ('lay on the table'⇒'to be topic') whose reading is hard to establish because both things are possible: There could be a physical copy of a video lying on an actual table or it could be the matter of discussion.

Considering the range of labels presented above one could argue that PIEs should be annotated based on a scale rather than with distinct labels, since it is not always a clear-cut decision whether an expression is compositional or not. E.g. Reddy et al. (2011) and Köper and Schulte im Walde (2016) use a scale from 0 to 5 to annotate the compositionality of noun compounds and particle verbs, respectively. As the latter work shows, this is not without caveats: The annotator agreement for all six categories was only 43%. To counter the low agreement the scale was divided into two disjunctive ranges ([0,2] and [3,5]), resulting in a binary classification after all. One could even argue that this approach resulted in a classification even more restrictive than the one we used, because there is no medium value which could represent a case, where both readings are active at the same time, i.e. there is no equivalent to the label both. Thus, it is unclear whether an annotation based on scales is better

suited for the annotation of PIEs, at least for our purposes.

5.1.3 Annotation

The annotation was conducted individually by three annotators with linguistic background. So each of the roughly 7000 sentences received three annotations on the basis of which we calculated the following Cohen's Kappa scores:

- annotator 1 – annotator 2: 0.9
- annotator 2 – annotator 3: 0.8
- annotator 1 – annotator 3: 0.77

These scores show a relatively strong agreement between the annotators which we could interpret as the annotation task being relatively straightforward.

Figure 5.2 shows an example from the corpus which is in column format. The first column contains the original tokens, the second the lemmas and the third the POS tags. The sentences were lemmatized with GermaLemma[8] and POS tagged with the TreeTagger[9]. Columns 4 to 6 contain the three individual annotations while column 7 represents the final label which was decided upon by the majority of labels. The index 2 represents the label IDIOMATIC. The final label for an instance was determined as follows: If the majority of annotators applied a certain label, we chose this as the final label. If all three annotators applied different labels, UNDECIDABLE was assigned.

Furthermore, two of the annotators were given the instruction to judge whether more than one sentence was needed to disambiguate a PIE instance. In figure 5.2, these judgements are prefaced bye *# context_judgement_1* and *# context_judgement_2*. *0* indicates that no further context was needed, while *1* would indicate otherwise. This was done because at publishing time the original version of COLF-VID 1.0 was restricted to one sentence per instance due to to licensing and these context judgments enable us to filter out certain sentences if we want to. The licensing situation changed when COLF-VID was merged with the SemEval 5b dataset to form the dataset for the Shared Task on the Disambiguation of German Verbal Idioms (Ehren et al., 2021). We will discuss this further in section 5.3.

[8] `https://github.com/WZBSocialScienceCenter/germalemma` [Accessed: 04.06.2024]

[9] `https://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/` [Accessed: 04.06.2024]

```
# global.columns = ID FORM LEMMA
POS ANNO_1 ANNO_2 ANNO_3 MAJORITY_ANNO
# article_id = T890825.128
# text = Bundesbahn will die
Notbremse ziehen
# context_judgement_1 = 0
# context_judgement_2 = 0
1 Bundesbahn Bundesbahn NN * * * *
2 will wollen VMFIN * * * *
3 die die ART * * * *
4 Notbremse Notbremse NN 2 2 2 2
5 ziehen ziehen VVINF 2 2 2 2
```

Figure 5.2: A sample idiomatic instance in COLF-VID

corpus	size	#PIE	lang.	labels	agr./ Kappa	I%	source	form type
COLF-VID 1.0 (2020)	6985	34	DE	literal, idiomatic, undecidable, both	0.77 < 0.9	77.55	TüPP-D/Z	V+NP/PP

Table 5.3: COLF-VID 1.0 overview.

As was stated earlier, we chose PIE types with the goal of maximizing the number of literal instances. But the idiomaticity rate in the last column (*I%*) of table 5.2 shows we were way off the mark with our assumptions for some PIE types: Eleven of the 34 types have an idiomaticity rate above 90%. Still, there is the possibility that this is due to the nature of the corpus rather than the nature of the chosen types. E.g. for *den Nerv treffen* ('hit a nerve'⇒'provoke a reaction') we only found one literal instance, but, as can be verified by a quick Google search, this would look very different in medical forums. The same goes for *auf der Strecke bleiben* ('stay on the track'⇒'to be left behind') which occurs much more often in its literal form in the racing context. Moreover, the total idiomaticity rate of 77.55% – which is considerably lower than that of the German SemEval 5b data set – shows that in general our assumptions about the literality rates were correct.

Another salient property of the label distribution is the low number of UNDECIDABLE and BOTH labels. Only 0.59% of all instances received one of those two labels. This is hardly surprising given that similar corpora (Sporleder et al., 2010; Haagsma et al., 2019) produced a similar picture. Table 5.3 displays an overview of the corpus' key characteristics.

Compared to the corpora discussed in Section 4.3, COLF-VID ranks 4th in size with respect to number of sentences, but only second to last

when it comes to the number of PIE types. The idiomaticity rate of COLF-VID is also higher than that of most other corpora but with 77.55% still far away from the highest value of 95.15% (PNV data set). The agreement is pretty much in line with what was reported for the other corpora. The corpus can be found on GitHub[10].

5.2 Analysis

Till now we discussed the creation of COLF-VID as well as its key statistics. The goal of this section is to provide deeper insights into the corpus by examining the nature of its VID types. To this end, we will see how their behavior fits into common assumptions made about certain types of MWEs in the literature.

5.2.1 Decomposable vs. non-decomposable

In a first step, we want to clarify how the pre-chosen VID types fit into the common classification scheme of decomposable vs. non-decomposable idioms. Out of the 34 VID[11] types we deem 29 types non-decomposable, i.e. it is not possible to map the individual components onto the idiomatic meanings. That leaves the following five for which a more or less clear mapping can be found:

- *auf den Zug aufspringen*: *Zug* ('train') → 'trend'; *aufspringen* ('jump on') → 'follow'
- *die Fäden ziehen*: *Fäden* ('strings') → 'influence'; *ziehen* ('pull') → 'exert'
- *eine Brücke bauen*: *Brücke* ('bridge') → 'connection'; *bauen* ('build') → 'establish'
- *von Bord gehen*: *Bord* ('board') → project, company, etc.; *gehen* ('go') → 'leave'
- *an Glanz verlieren*: *Glanz* ('luster') → 'attractivity'; *verlieren* ('loose') → 'loose'

[10] `https://github.com/rafehr/COLF-VID` [Accessed: 04.06.2024]

[11] We use the term VID instead of PIE here, because the decomposable/non-decomposable classification only makes sense for idiomatic expressions.

What stands out is that most of the VID types in COLF-VID are what (Nunberg et al., 1994, p. 506) would call "synchronically relatively transparent metaphors", i.e. the motivation behind the meaning of a VID seems rather straightforward to us. For example, why *mit dem Feuer spielen* ('play with fire'⇒'take a risk') stands for the act of taking risks is not exactly a mystery. Contrast that with an expression like *shoot the breeze* whose origin is opaque to modern day speakers. One obvious reason for the prevalence of transparent VID types in COLF-VID might be our preference for PIEs with a high ratio of literal readings: If an expression is used a lot literally, the motivation for the respective idiom is less likely to be lost.

5.2.2 Flexibility of non-decomposable VIDs

The (non-)flexibility of idioms is a hotly debated topic in the MWE literature. According to a classic theory by Nunberg et al. (1994), the flexibility of an idiom, or its lack thereof, can be accounted for by its degree of decomposability, i.e. decomposable idioms are more flexible than non-decomposable ones. As stated earlier, decomposability describes the property that the idiomatic meaning can be distributed among the constituents of the idiom. For Nunberg et al., this is a prerequisite for the ability to take part in certain morphosyntactic operations, e.g. modification by adjective or by relative clause: "In order to modify part of the meaning of an idiom by modifying a part of the idiom, it is necessary that the part of the idiom have a meaning which is part of the meaning of the idiom." (Nunberg et al., 1994, p. 500). In reference to Ernst (1981) this is usually termed internal modification and it is illustrated by (14):

(14) Pat got the job by **pulling strings** that weren't available to anyone else.

Pull strings in this context has the meaning to 'exploit one's connections' which can be distributed onto the components of the expression (*pull* = 'exploit', *strings* = 'connections'). Thus, the fact that *strings* carries part of the idiomatic meaning allows for the relative clause to modify it. Since this distribution of meaning over its parts does not occur for non-decomposable idioms, they do not exhibit this kind of syntactic flexibility, or so Nunberg et al. argue. The same goes for other operations like topicalization, passivization or elliptical constructions.

Among others, Bargmann and Sailer (2018) challenge this assumption as there exists data suggesting a considerably higher degree of flexibility. E.g. for *kick the bucket* passivization is attested, or the metalinguistic

modification of *bucket* by adjectives like *proverbial* or *metaphorical*.

(15) **The bucket** will be **kicked**.

(16) He **kicked the** proverbial **bucket**.

Another example for modification is what is termed *external modification*:

(17) He **kicked the** political **bucket**.

Here, *political* is not modifying the idiomatic sense of bucket (since there isn't any) but that of the expression as a whole: The sentence describes the end of someone's political aspirations, i.e. someone "kicked the bucket" in the political domain. Nevertheless, it is an example of syntactic flexibility of a non-decomposable idiom and as such speaks against flexibility being dependent on the distribution of the idiomatic meaning over the idiom's parts. According to Fellbaum (2019), this kind of external modification by means of an adjective is available to all idioms and thus does not tell us much about their syntactic flexibility.

Consequently Fellbaum (2019) presents more challenging data to the theory of Nunberg et al. for German and for English. She lists examples for a variety of syntactic operations involving non-decomposable idioms: passivization, pseudo-clefting, relativization, compounding of an idiom-interal NP, determiner variation, reversal of polarity, quantification, adjectival modification and topicalization.

5.2.3 Flexibility of non-decomposable VIDs in COLF-VID

The goal of this section is to examine how the COLF-VID data fits into the assumptions made about the flexibility of non-decomposable idioms presented above. In the following we will take a closer look at *die Notbremse ziehen* ('pull the emergency brake'⇒'to end a dangerous situation'). We will start with an instance of external modification which - as mentioned above - is quite common:

(18) Die Bundesregierung muß nun schon im zweiten Jahr
The federal government must now already in the second year
hintereinander die finanzielle **Notbremse ziehen**
consecutively the financial **emergency brake pull**
'For the second year in a row, the federal government has had to pull the financial emergency brake.'

In Example (18), the idiom-NP *Notbremse* ('emergency brake') is modified by *finanzelle* ('financial') on a syntactic level, but semantically, the idiom as a whole is modified, since the NP does not carry part of the idioms meaning. So, roughly the sentence means: 'Financially, the state is in trouble and the government has to undertake immediate measures to prevent a further deterioration of the situation.' In other instances of external modification *Notbremse* is modified by the adjectives *haushaltspolitische* ('fiscal') and *politische* ('political').

Furthermore we can find examples for topicalization:

(19) **Die Notbremse gezogen** hat die Regierung der
The emergency brake pulled has the government of the
Republik Zypern [...]
republic Cyprus [...]
'The government of the republic Cyprus has put an end to the situation'.

However, according to Nunberg et al. (1994) German topicalization is different from the English construction in that it does not necessarily emphasize the content of the initial element and consequently is not dependent on the topicalized constituent having a meaning. Even if we agree with this assessment, there are quite a few more examples we can contribute to the catalogue of challenging data, e.g. relative clause constructions, passivization and internal modification by means of a noun:

(20) Die **Notbremse**, die Wedemeier **zog**, zeigte, daß
The **emergency brake,** which Wedemeier **pulled,** showed that
man auch mit hemdsärmeligen Methoden die "Asylantenflut"
one also with down-to-earth methods the "asylee flood"
eindämmen kann.
contain can.

(21) Bei VW muß nach Ansicht Gansäuers angesichts der
At VW must according to opinion Gansäuers in view of the
hohen Verluste nun die **Notbremse gezogen** werden.
high losses now the **emergency brake pulled** be.

(22) Die [...] psychischen Belastungen werden ignoriert. Sie
The [...] psychological stress are ignored. They
führen zu emotionaler Verkürzung, Verrohung und
lead to emotional reduction, brutalization and
Alkoholmißbrauch. **Notbremsen**, die **gezogen**
alcohol abuse. **Emergency brakes,** which **pulled**

werden.
are.

(23) Bei seinem neuen Film scheint er nun die **Notbremse**
With his new film seems he now the **emergency brake**
seiner vielleicht zu internationalen Karriere **gezogen** zu
of his maybe too international career **pulled** to
haben.
have.

These examples seriously undermine Nunberg et al.'s assertion about the syntactic inflexibility of non-decomposable idioms. This is not only relevant from a linguistic point of view, but it also has implications for the automatic processing of VIDs. After all, morphosyntactic properties of VIDs could give the classifier valuable cues on the correct reading. For example, if it were true that non-decomposable VIDs do not passivize, a passive construction like in (21) would mean the classifier should assign the label *literal* – a wrong choice in this case. If, on the other hand, there is no difference in flexibility regarding VIDs, then a classifier has to rely solely on contextual features. However, whether non-decomposable VIDs can show the same flexibility as decomposable ones is only one part of the question when it comes to automatic processing. The other part would be whether this is regular or rather exceptional behavior. Since the models we employ are stochastic in nature, this question is more relevant. E.g. if in 9 of 10 cases a passivized *kick the bucket* had a literal reading and only one was idiomatic, a classifier would do well in predicting the former label. Or take the following examples from COLF-VID:

(24) Darum zwinkerte man sich in der Filiale Am Dobben
That's why winked one each other in the branch Am Dobben
wie auch in der Zentrale bedeutungsvoll zu, wenn die
like also in the head office meaningful to, when the
Jahresrechnungsberichte **auf den Tischen lagen**.
annual financial reports on the table lay.
'That's why both, at the Am Dobben branch and the head office, they winked meaningfully at each other when the annual financial reports were available.' IDIOMATIC

(25) Weiße Taschenbuchausgaben **liegen** überall **auf** den
White paperback editions lay everywhere on the
Tischen.
tables. LITERAL

In its canonical form, the noun of the VID *auf dem Tisch liegen* ('to lay on the table'⇒'to be topic') is in its singular form, but, as can be seen in Example (24), it is possible for the expression to keep its idiomatic meaning when we have the plural form (*Tischen*). However, in COLF-VID there are 29 instances of this PIE type where *Tisch* is in its plural form and 25 of those instances have a literal reading (as in Example (25)), only four are idiomatic. So in this case, the classifier could exploit the morphological flexibility which – if not impossible – is at least atypical for the VID and thus gives a valuable hint for the correct reading. Hence, to examine the implications of morphosyntactic flexibility of VIDs on automatic PIE disambiguation we would have to perform a more extensive evaluation in terms of quantity. It would be too time-consuming to inspect the roughly 7000 instances manually, but we will explore automatic means to compute the flexibility of an expression in the next section.

5.2.4 Computing MWE Variability

In the last section, we were concerned with the question whether, *in principle*, non-decomposable VIDs in COLF-VID are morphosyntactically as flexible as their literal counterparts. We found examples in our data that suggest they are. Now we want to investigate whether they are as flexible *in general*, meaning that there is no difference between idiomatic and literal instances in that regard. Our assumption is that this is not the case and that literal instances still have the tendency to be more flexible, making it possible for classifiers to leverage these differences.

In order to examine this automatically, we use the variability measure for MWEs proposed by (Pasquer et al., 2018a). More specifically, Pasquer et al. define two variability measures: one based on syntactic similarity (S^S) and the other based on what they call linear similarity (S^L). Both are based on the Sørensen–Dice coefficient:

$$S(O_1, O_2) = \frac{2 \times |P(O_1) \cap P(O_2)|}{|P(O_1)| + |P(O_2|} \tag{5.1}$$

where $P(O_1)$ and $P(O_2)$ are the sets of relevant features of two objects O_1 and O_2.

S_L is defined in "terms of the POS of the elements inserted between the lexicalized components" of an expression (Pasquer et al., 2018a, p. 427). So in our case O_1 and O_2 are two instances (or variants) of a VID type or its literal counterpart and $P(O_1)$ and $P(O_2)$ are then the sets[12] of POS of the words in-between components for O_1 and O_2.

[12]Multiple occurrences of the same POS are disregarded.

S_S is defined in terms of the outgoing dependencies of an expression's component and is supposed to "account for long-distance arguments and modifiers not necessarily included between the lexicalized components" (Pasquer et al., 2018a, p. 427). Thus, O_1 and O_2 are the corresponding components of two instances of an expression (for example *bucket* in *kick the bucket*) and $P(O_1)$ and $P(O_2)$ are their outgoing dependencies. The score for the whole MWE is then the weighed sum of the component scores with the weights summing up to 1[13].

The rigidity score, which is supposed to quantify the inflexibility of an expression, is then the average over all pairs of an expression's instances, e.g. we have $\binom{n}{2}$ pairs for n instances. The variability score is then the complement of the rigidity score (i.e. variability score $= 1 -$ rigidity score). Please see §2 in Pasquer et al. (2018a) for a more rigorous definition and detailed examples.

The reason why we go into this much detail at all is to address a certain issue that arises when using the Sørensen–Dice coefficient in this way. For illustration, consider the following two instances of *spill the beans*:

(26) He finally **spilled the beans**.

(27) The conductor **spilled the beans** eventually.

For both examples, we do not have any elements inserted between the lexicalized components, so we have the following situation:

$$S_L(O_1, O_2) = \frac{2 \times |\emptyset|}{|\emptyset| + |\emptyset|} = \frac{2 \times 0}{0 + 0} = \frac{0}{0} \tag{5.2}$$

where O_1 is the instance in Example (26) and O_2 the instance in Example (27). We can see that in cases like these, the Sørensen–Dice coefficient is undefined. In our view, this is undesirable behavior, since both instances exhibit the same property of not having any elements inserted in-between the lexicalized components and accordingly are maximally similar. This is not addressed in Pasquer et al. (2018b)[14], but we need to, since COLF-VID contains a lot of PIE instances that do not have any intervening words in-between components. This is why we implemented S_L in a way to account for this. If two instances do not have any insertions between their lexicalized components, they get a similarity value of 1. The same goes for S_S, when components do not have any outgoing edges.

[13]The weights' values are a hyperparameter.

[14]Maybe they do not have any instances in their data without insertions between lexicalized components.

PIE type	L	I	Lin. Var. L	Lin. Var. I	Syn. Var. L	Syn. Var. I
auf den Arm nehmen	39	50	0.15	**0.2**	**0.35**	0.25
auf der Straße stehen	93	156	**0.46**	0.37	**0.16**	0.11
auf Tisch liegen	262	678	0.34	**0.37**	**0.2**	0.1
eine Brücke bauen	109	238	**0.79**	0.75	**0.64**	0.59
im Regen stehen	69	302	**0.66**	0.36	0.19	**0.23**
in den Keller gehen	34	91	**0.38**	0.36	**0.16**	0.09
ins Wasser fallen	67	186	0.49	**0.63**	**0.14**	0.13
Luft holen	100	66	**0.75**	0.51	**0.48**	0.41
die Notbremse ziehen	51	275	**0.7**	0.66	**0.26**	0.24
eine Rechung begleichen	89	162	0.73	**0.74**	**0.56**	0.54
vor der Tür stehen	189	409	**0.39**	0.31	**0.16**	0.12
über Bord gehen	62	52	**0.59**	0.39	**0.2**	0.14
über Bord werfen	54	389	0.38	**0.52**	**0.17**	0.14
	1218	3054	0.52	0.47	0.28	0.24

Table 5.4: Variability results for a selection of PIEs.

Table 5.4 shows the variability results for a selection of 13 PIE types from COLF-VID. We chose those types with at least 100 instances and an idiomaticity rate of below 90%. The table shows different pictures with respect to linear (*Lin. Var.*) and syntactic variability (*Syn. Var.*). The chosen PIE types exhibit, on average, a greater linear than syntactic variability and the literal instances of the PIE (L) are more variable according to both measures than the VID (I), albeit not by much: 0.52 (L) and 0.47 (I) vs. 0.28 (L) and 0.24 (I). However, if we look at the individual PIE types, the picture is a bit less clear for linear variability as for the five types *auf den Arm nehmen*, *auf dem Tisch liegen*, *ins Wasser fallen*, *eine Rechnung begleichen* and *über Bord werfen* the VID is more flexible than its literal counterpart. By looking at a couple of examples we can get an idea why that is:

(28) Das Comeback des US-amerikanischen Schwimm-Opas Mark
The comeback of the US-American swim grandpa Mark
Spitz **fiel** nach 15 Jahren Pause **ins** **Wasser**
Spietz fell after 15 years break into the water
'The comeback of the US-American swim-grandpa Mark Spitz fell short a 15 year break'

(29) Trotzdem **liegt** noch kein konkreter Vorschlag
Nevertheless lays yet no concrete proposal
auf dem Tisch.
on the table.
'Nevertheless, there is still no concrete proposal.'

The nature of the exertions in Example (28) and (29) gives us no reason to believe these are exceptions. In (28), we have a PP (*nach 15 Jahren*

Pause) modifying the verb of the VID (*fiel*) and with it the whole expression. And there is not really a reason why VIDs should not be able to be modified adverbially. Similar applies to Example (29): There is nothing in German that prohibits the subject and its modifiers (*noch kein konkreter Vorschlag*) from following the verb (*liegt*), even for VIDs. So this kind of behaviour is expected and a certain flexibility not surprising.

For the syntactic variability measure on the other hand, the picture seems much clearer: For all but one PIE type (*im Regen stehen*) the literal counterpart has a higher variability score, albeit the margins being very small, i.e. for most PIE types there is not much difference between the scores for the two classes. Still, this is in line with the assumption that, in general, non-decomposable VIDs are less flexible than their literal counterparts. Thus, even though we have seen counter-examples for this theory in the previous chapter, these results suggest that literal instances at least have a minor tendency to be more flexible syntactically, which is something a classifier could potentially learn.

Please note that the measures presented above do not capture morphological variability like inflection or derivation.[15] Something like the plural form of *Tisch*, as discussed in the examples above, would be disregarded accordingly. For future work, it would be interesting to see what a measure capturing morphological variation would reveal about our data.

5.3 Shared Task Data Set

In order to invite other members of the MWE community to experiment with COLF-VID, we organized the *Shared Task on the Disambiguation of German Verbal Idioms*[16] in conjunction with KONVENS 2021 (Ehren et al., 2021). Due to their similarity, we decided to merge COLF-VID with the German part of the SemEval 2013 5b (SE5b) data set presented in Section 4.3. The SemEval-2013 Task 5 was concerned with the evaluation of phrasal semantics and comprised two different tasks: 5a and 5b. While 5a consisted of computing the similarity of word sequences of different length, 5b was concerned with deciding on the compositionality of phrases in context (Korkontzelos et al., 2013); or in other words, to decide if a certain expression was an instance of a VID or not. Consequently, the SemEval 5b data set has very similar properties to COLF-VID, so it

[15] Pasquer et al. (2018a) did not propose a variability measure that captures morphological flexibility.

[16] We had not yet adopted the term PIE at this point.

was quite straightforward to align them. But there were a few issues to address nevertheless.

The most notable difference between COLF-VID and SE5b is that the latter also includes non-verbal PIEs, like *steif und fest* ('stubbornly') or *zweite Geige* ('second fiddle'), which we filtered out in order to assure the homogeneity of the resulting data set. Furthermore, in contrast to COLF-VID, SE5b included the preceding and succeeding sentence for every PIE instance, thus we also added the same amount of context to our corpus. According to Korkontzelos et al. (2013) even the two preceding and two succeeding sentences for a PIE instance were included, but – at least for German – upon examining the data this actually does not seem to be the case except for a few instances. Another difference is the source of the corpora: While COLF-VID was created from a newspaper corpus, SE5b's instances were drawn from a web corpus which makes it more heterogeneous, but also results in significantly more noise, like superfluous characters and partial or otherwise ungrammatical sentences. Given its amount, we decided to leave the noise in as its removal would have altered the original data set too much.

An issue which we had to give a little more thought to was the fact that in SE5b for some types only the canonical form was annotated. E.g. *mit dem Feuer spielen* ('play with fire'⇒'take a risk') always included the determiner, while – as for reasons discussed above – this was not the case for COLF-VID. Thus, we had to decide whether to merge the respective types or to keep them distinct. We decided on the latter, in order to preserve the integrity of both corpora. After all, the decision to include or exclude certain variations directly influences the idiomaticity rates of a type, so it might not be a coincidence that *mit dem Feuer spielen* has a higher idiomaticity rate in SE5b compared to *mit Feuer spielen* in COLF-VID (96.20% vs. 86.90%, cf. Table 5.5). In general, SE5b has a very high idiomaticity rate which in the end drives up the idiomaticity rate of the whole corpus which is much higher than for COLF-VID, even though SE5b is not even half the size (cf. table 5.6).

Table 5.5 shows the statistics with regard to the annotated readings and idiomaticity rates of the combined data set per PIE type, while Table 5.6 gives the same statistics for the whole data set. It can be found on Zenodo[17] in the same split as was used in the shared task. In Chapter 6, we will come back to the data set when we cover the organization as well as the results of the shared task.

[17] https://zenodo.org/records/5920622 [Accessed: 04.06.2024]

VID type	Lit.	Idiom.	Und.	Both	I%	VID type	Lit.	Idiom.	Und.	Both	I%
am Boden liegen	35	11	0	1	23.40	auf die Nase fallen	7	69	0	0	90.79
an Glanz verlieren	0	14	0	0	100.00	Korb bekommen	12	82	0	0	87.23
an Land ziehen	25	234	0	0	90.35	Auge zudrücken	8	89	0	0	91.75
am Pranger stehen	0	5	0	0	100.00	Dampf ablassen	5	103	0	0	95.37
Atem anhalten	10	30	0	0	75.00	die Stiefel lecken	2	10	0	0	83.33
auf Abstellgleis stehen	15	11	0	0	42.31	einen Korb geben	7	81	0	0	92.05
auf Arm nehmen	39	50	0	0	56.18	gute Karten haben	5	92	0	0	94.85
auf Ersatzbank sitzen	16	5	0	0	23.81	Handtuch werfen	6	99	0	0	94.29
auf Straße stehen	92	156	1	0	62.65	Hose anhaben	2	11	0	0	84.62
auf Strecke bleiben	4	610	1	0	99.19	im gleichen Boot sitzen	0	94	0	0	100.00
auf Tisch liegen	254	677	10	1	71.87	in den Sand setzen	8	87	0	0	91.58
auf Zug aufspringen	5	186	0	0	97.38	in den Schatten stellen	3	92	0	0	96.84
Brücke bauen	108	237	1	0	68.50	keinen Bock haben	0	91	0	0	100.00
Fäden ziehen	36	226	0	0	86.26	Korb kriegen	0	6	0	0	100.00
in Blut haben	29	7	0	0	19.44	mit dem Feuer spielen	3	76	0	0	96.20
in Keller gehen	33	89	0	0	72.95	rote Zahlen schreiben	0	104	0	0	100.00
in Luft hängen	28	256	0	0	90.14	über den Tisch ziehen	2	91	0	0	97.85
in Regen stehen	69	301	4	4	79.63	Braten riechen	6	84	0	0	93.33
in Rennen gehen	11	50	0	0	81.97	die Daumen drücken	0	95	0	0	100.00
in Sackgasse geraten	2	98	0	0	98.00	gegen den Strom schwimmen	0	80	0	0	100.00
in Schatten stehen	7	52	0	1	86.67	Geld zum Fenster hinauswerfen	1	25	0	0	96.15
in Schieflage geraten	3	39	1	0	90.70	Löffel abgeben	1	85	0	0	98.84
in Wasser fallen	66	183	0	0	73.49	heilige Kuh schlachten	1	83	0	0	98.81
Luft holen	99	66	4	0	39.05	Hut nehmen	6	69	0	0	92.00
Nerv treffen	1	282	0	0	99.65	im Geld schwimmen	0	99	0	0	100.00
Notbremse ziehen	57	367	0	1	86.35	ins Gras beißen	3	78	0	0	96.30
Rechnung begleichen	88	160	0	0	64.52	Öl ins Feuer gießen	0	99	0	0	100.00
von Bord gehen	45	48	0	0	51.61	schlechte Karten haben	4	96	0	0	96.00
vor Tür stehen	189	407	1	1	68.06	Rücken stärken	10	81	0	0	89.01
Zelt aufschlagen	52	40	7	1	40.00	Vogel abschießen	11	80	0	0	87.91
über Bord gehen	61	51	1	0	45.13	unter Strom stehen	23	65	0	0	73.86
über Bord werfen	54	389	0	0	87.81	mit Feuer spielen	9	73	2	0	86.90
über Bühne gehen	2	198	0	0	99.00	Frucht tragen	20	70	0	0	77.78
auf dem Schlauch stehen	1	83	0	0	98.81						

Table 5.5: Statistics for the shared task data set.

	Lit.	Idiom.	Und.	Both	I%
COLF-VID	1511	5386	33	10	77.61
SemEval 5b data	190	2771	0	0	93.58
Total	1701	8157	33	10	82.39

Table 5.6: Total data set statistics.

5.4 Lessons Learned

In this section, we discuss the lessons learned from creating a PIE corpus. Or in other words: We will address what we would have done differently in retrospective.

One of the main issues is the imbalance with regard to instances per PIE type. As was addressed earlier, the variance among PIE types is very high, which means that some types have a much greater influence on the overall performance of a classifier trained on the corpus. E.g. it should be relatively easy for a system to figure out that *auf der Strecke bleiben* is used predominantly with its idiomatic meaning and with its 621 instances this one type will have a very large influence on the evaluation. Another problem caused by this imbalance occurs when the data has to be split into training and test data: If the data was split randomly, we could run the risk of PIE types with a lower number of instances not occurring at all in one of the sets. So in order to assure their representation in all splits we have to apply the split ratio not to the data set as a whole but to every individual PIE type. We will go into further detail at a later point.

The main reason for this imbalance is the small size of the source corpus (TüPP-D/Z) and the fact that we wanted the corpus to have a certain size to ensure its suitability for training. That means, if for some types only a small number of instances could be found, other types had to do the heavy lifting. We could have switched to another corpus, but we refrained from doing so, since our goal was to annotate PIEs in context and TüPP-D/Z consists of complete articles. Another, and probably better, alternative would have been to choose a larger corpus like DECOW (Schäfer and Bildhauer, 2012) and just exclude sentences that do not have enough context to decide on the correct reading of a PIE instance.

Another related issue is the low number of PIE types. COLF-VID is second to last in this regard when compared to the PIE corpora presented in Section 4.3. With more PIE types we would not have had to rely on so few PIE types to reach the desired number of instances. But again, the small size of the corpus did not help, as for some of the pre-chosen PIE types we did not find any examples at all. Furthermore, the homogeneity of TüPP-D/Z (only newspaper articles) could have been an issue because some PIE types might rather appear in less formal texts.

Last but not least, our annotation failed to account for the fact that there sometimes is ambiguity on the VID level as well, i.e. a VID can have multiple (albeit often related) meanings. However, this is not irrelevant for some tasks such as machine translation. To our knowledge, this is something no other PIE corpus accounts for as well, but it may be

something to aim for in the future.

This concludes the background part of this thesis. In the next one, we present the PIE disambiguation experiments we conducted on the basis of COLF-VID. But first we concern ourselves with some aspects of the VMWE identification task.

PART III

Experiments

Chapter 6

Experiments

In this chapter, we present experiments on VMWE identification and PIE disambiguation. Chapter 6.1 will be concerned with the former and Chapter 6.2 with the latter. Furthermore, we present the *Shared Task on the Disambiguation of German Verbal Idioms* (Section 6.2.2) we organized in conjunction with KONVENS 2021. Lastly, we discuss our attempt to generate new training data with ChatGPT in order to increase the performance of our PIE disambiguation system (Chapter 6.2.4).

6.1 VMWE Identification

This chapter is concerned with experiments on the task of MWE identification as described in chapter 3.2. More specifically, we limit ourselves to the identification of verbal MWEs (VMWEs) as the most challenging subclass of MWEs. All experiments were conducted on subsets of version 1.1 and 1.2 of the PARSEME corpus (cf. Section 4.1), respectively.

In Section 6.1.1, we present a simple BiLSTM classifier that performed in *Edition 1.1 of the PARSEME Shared Task on Automatic Identification of Verbal Multiword Expressions* in 2018 (Ramisch et al., 2018). We entered both the closed and open track of the competition, since we employed randomly initialized and pretrained embeddings[1]. For the closed track, we submitted results for seven of the 20 languages (BG, DE, EL, ES, FR, PL and PT) which constituted edition 1.1 of the PARSEME corpus, but only one result for the open track (DE) due to time constraints in training the embeddings.

In Section 6.1.2, we examine how it affects performance if we train a classifier for each VMWE type separately instead of one classifier for all

[1]As soon as other resources than the ones supplied by the organizers were used, a system competed in the open instead of the closed track.

VMWE types at once. For these experiments, we employed a fine-tuned multilingual BERT model for 11 of the 14 languages (DE, EU, FR, GA, HI, PL, PT, RO, SV, TR, ZH) of the PARSEME corpus 1.2. Three languages (DE, HE, IT) had to be ruled out due to issues during preprocessing.

6.1.1 BiLSTM Classifier

In Section 3.2.1, we roughly divided approaches to MWE identification into parsing-based and non-parsing-based methods. Given the rise of deep learning and due to the fact that MWE identification can be modeled as a sequence labeling task, the majority of the latter consists of neural architectures proven to be very successful at these kind of tasks, e.g. bidirectional recurrent neural nets (BiRNNs) or, more recently, transformer-based models like BERT. We joined this line of work by employing a BiRNN with long short-term memory (LSTM) (Hochreiter and Schmidhuber, 1997) units in these early experiments on VMWE identification.

This section is heavily based on our work *Mumpitz at PARSEME Shared Task 2018: A Bidirectional LSTM for the Identification of Verbal Multiword Expressions* (Ehren et al., 2018).

Motivation

The reason why BiLSTMs are so popular for sequence labeling tasks is their supposed ability to remember contextual information for a longer time than vanilla RNNs as the latter are more susceptible to the vanishing/exploding gradient problem. This is an especially desirable property for a task such as VMWE identification, since VMWE components need not be adjacent and there can be a lot of material interfering in-between the parts of a discontiguous VMWEs:

(1) Es **scheiden** die Vertreter von Ruanda, Argentinien,
it **drop** the representatives of Rwanda, Argentine,
Oman, Nigeria und Tschechiens **aus**.
Oman, Nigeria and Czechia **out**.
'The representatives of Rwanda, Argentine, Oman, Nigeria and Czechia drop out.'

As Example (1) illustrates, it is very common for German VPCs to appear in split word order, here with nine words in-between the base verb *scheiden* ('drop') and the particle *aus* ('out'). The propensity for long-distance dependencies obviously differs with respect to the language, but – as we have discussed in earlier chapters – for a language like German it is far

from out of the ordinary as the DE data set of the PARSEME 1.0 corpus on average has the longest discontinuities and "30.5% of VMWEs have discontinuities of 4 or more tokens" (Savary et al., 2017, p. 35).

Besides their good memory, the bidirectionality of BiLSTMs is another property that is advantageous for VMWE identification (or sequence labeling in general) because it allows the model at any given time step to incorporate information from the "past" and the "future" into its decision how to label a token. For simplification, we consider a BiRNN without LSTMs:

$$\overrightarrow{h}_t = g(\overrightarrow{W}_{hh}\overrightarrow{h}_{t-1} + \overrightarrow{W}_{hx}x_t + \overrightarrow{b}_h) \tag{6.1}$$

$$\overleftarrow{h}_t = g(\overleftarrow{W}_{hh}\overleftarrow{h}_{t+1} + \overleftarrow{W}_{hx}x_t + \overleftarrow{b}_h) \tag{6.2}$$

$\overrightarrow{h}_t$ denotes the forward pass where we consider the hidden state of the previous time step $t-1$ during the computation of the current time step t, i.e. the input is processed from left to right. $\overleftarrow{h}_t$ denotes the backward pass, where we consider the hidden state of the subsequent time step $t+1$ during the computation of the current time step t, i.e. the input is processed from right to left. The forward and backward pass are computed by two separate LSTMs, so in order to combine information about the past and the future $\overrightarrow{h}_t$ and $\overleftarrow{h}_t$ are concatenated for every time step. The following serves to illustrate the usefulness of this kind of architecture:

(2) The Knight of the Sad Countenance does not **give up** his obsessions.

If a VMWE tagger received a sentence such as (2) as input, it would have to recognize that *give* is the base verb of a VPC and *up* its particle, rather than a preposition. In order to make the latter decision it does well to remember the fact that it saw *give* shortly before. However, just as *give* informs the system about the MWE status of *up*, the appearance of *up* is a good indicator that *give* is not used as standalone verb, but as part of a VPC. Thus, it can only be beneficial if the system already knows about the upcoming *up* when it encounters *give*.

Labeling

In Section 3.2, we already addressed the issue of a specific kind of *overlap* where distinct MWEs share tokens, i.e. the same token is part of multiple MWEs. In the context of classification, this means that a system needs to be able to output more than one label per token:

(3) We should **turn up the heat** until they get nervous and **make a**
* * 1;2 1;2 1 1 * * * * * 3 3
mistake.
3

In Example (3), the tokens *turn* and *up* form a VPC and at the same time are part of the VID *turn up the heat*. Accordingly, they receive two different identifiers (1 for the VPC and 2 for the VID). This is not a trivial issue, since classifiers usually are functions with the elements of the co-domain representing distinct labels. There exist different approaches to this issue, but they all have their drawbacks. For instance, it is possible to always choose the longest sequence, but this would result in loosing the information about the embedded MWE (Constant et al., 2017). With this method we would only tag the VID as a whole without marking the VPC embedded in it:

(4) We should **turn up the heat** until they get nervous and make a
O O B I I I O O O O O B I
mistake.
I

To counter this issue, the multiple levels of the PARSEME annotation have to be preserved. We will see an approach for how this can be done (to an extent) for the IOB format in the next section.

Note that we presuppose a conversion from the PARSEME format to another one like the IOB format because the PARSEME format is not suitable to be used in a classification task for various reasons. Besides the fact that it potentially uses sequences of multiple labels, it also assigns identifiers for VMWE instances per sentence: The distinct integers do not denote different classes with different properties but only the numbering of VMWE instances in a sentence. Accordingly, it would be quite confusing for a classifier if the same VMWE type received different identifiers in different sentences. The conversion of PARSEME style labels to BIO/IOB or other formats is usually not discussed in the literature despite the fact that it is not a trivial question that potentially has great effects on the upper bound performance of the model.

For our experiment, the approach to label conversion was to convert the PARSEME annotation to binary labels which just marked a token if it was part of a MWE or not. The motivation for this was to reduce the complexity of the task and reduce data requirements. However, as it is the case for the IOB format, in doing so we lost information about overlapping VMWEs as well the distinction between different VMWEs:

(5) We should **turn up the heat** until they get nervous and **make a**
0 0 1 1 1 1 0 0 0 0 0 1 1
mistake.
1

In Example (5), *1* stands for 'part of a VMWE' and *0* for 'not part of a VMWE'. Hence, we lost the information about the embedded VPC and the distinction between the three different VMWEs *turn up*, *turn up the heat* and *make a mistake* which in the original annotation all received different identifiers. However, for evaluation it was necessary to regain part of the lost information which we tried to do by applying a heuristic. Differentiating the words tagged as VMWEs in a sentence was done with the following method: First, it identified all the words in a sentence that were labeled as VMWEs and had the universal POS-tag VERB and enumerated them. In the next step, every word that was a direct dependent of an enumerated verb and was tagged as a VMWE received the same identifier as this verb. Finally, every verb not labeled as a VMWE by the classifier that had a dependent labeled as such, also got the VMWE label. The reasoning behind this heuristic was that the PARSEME annotation guidelines define VMWEs as MWEs whose syntactic head in the prototypical form is a verb. To illustrate how it works consider Figure 6.1.

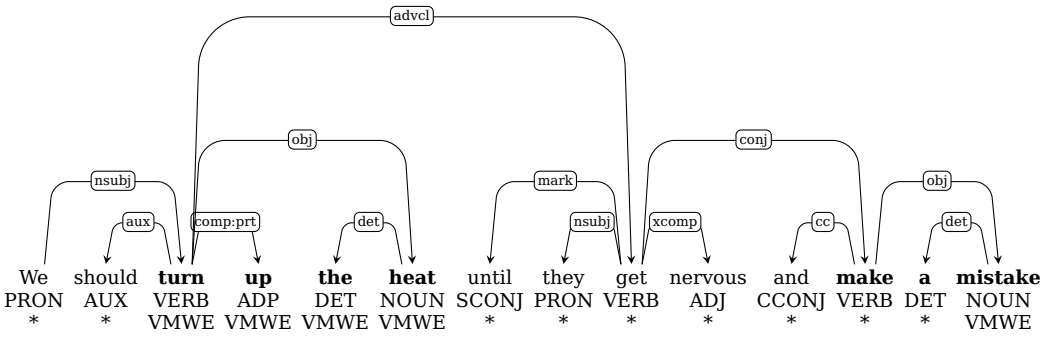


Figure 6.1: Heuristic to regain PARSEME-style labels.

The third row represents a possible output of the system where *turn up the heat* was tagged correctly in its entirety, but *make a mistake* was only identified partially. So during the first step only *turn* receives an identifier as it is the only token with the universal POS tag *VERB* that was labeled a VMWE. Next, every direct dependent of *turn* that is labeled as a VMWE is tagged with the same identifier, i.e. *up* and *heat*. Lastly, as the heuristic searches for non-verbal tokens tagged as VMWEs, *mistake* receives another identifier and its direct head in the dependency graph,

make, is tagged alongside with it. This results in the PARSEME-style labeling seen in Figure (6).

(6) We should **turn up the heat** until they get nervous and **make a mistake**.
* * 1 1 * 1 * * * * * 2 * 2

There are several issues with this approach. The main one being that it only considers direct dependents and thus fails to identify modifiers and determiners that often are lexicalized (like *a* and *the* in Figure 6.1). Also, it is questionable if the strategy to retroactively tag verbs as being part of a VMWE if one of their dependents was tagged does more harm than good. Lastly, with this approach we are not able to regain information about nested VMWEs. In the next section, we will discuss an experimental setup which is able to do so to a certain extent.

Experimental Setup

In the last section, we covered the output of our classifier, the following is concerned with its input and the overall architecture. Since the PARSEME datasets not only contain VMWE annotation but come in a format compatible with CoNLL-U[2], there are already a variety of features ready to be used. We experimented with combinations of the following ones: word form (WF), lemma (L), universal POS (UP), language specific POS (LP), head of the current word (H) and the dependency relation to the head (D). Due to time constraints, the performance on the German dev set was used to decide on the feature combination for all other languages as well. The features with the highest F1 score were a combination of L, LP and D. Hence, despite categorizing our approach as non-parsing-based, we still relied on syntactic information in the form of dependency relations.

We represented the lemmas as embeddings of size 50, while the language-specific POS tag as well as the dependency relation to the head were represented by embeddings of size 20. Then, every token in an input sentence was represented by a concatenation of these three embeddings, i.e. a vector of size 90. Accordingly, sentences were represented as sequences of those vectors and fed into the BiLSTM with a single hidden layer with 100 units, followed by a softmax output layer (cf. Figure 6.2). Furthermore, a dropout layer with a rate of 0.1 was used to counter overfitting.

[2] `https://universaldependencies.org/format.html` [Accessed: 04.06.2024]

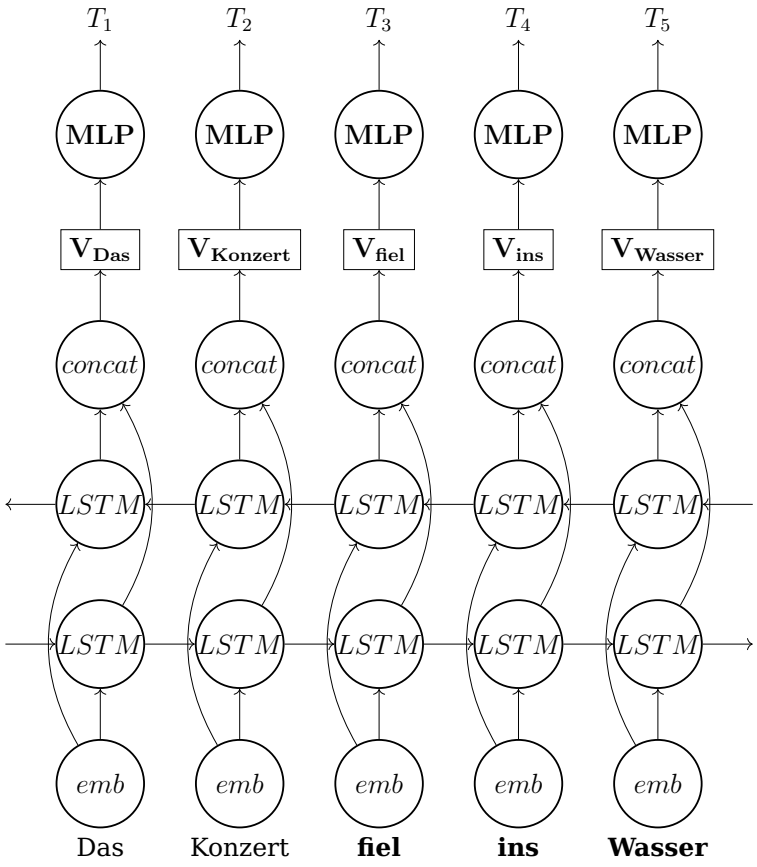


Figure 6.2: Architecture of the BiLSTM model for VMWE identification.

The shared task had two different tracks, an open and a closed one. For the closed track, only resources provided by the organizers were allowed, while the open track permitted additional ones such as lexicons or other corpora (e.g. to compute association measures). We submitted results for seven languages to the closed and for one language to the open track. The systems for both tracks were essentially the same, the only difference being that pretrained embeddings were used to represent the lemmas for the open track in contrast to the randomly initialized ones for the closed track. To that end, we trained a skip-gram model on the German DECOW16 corpus (Schäfer and Bildhauer, 2012) that consists of 11 Billion tokens and shuffled sentences. Those pretrained embeddings were of size 100 which resulted in an input vector with 140 dimensions when concatenated with the LP and D feature vectors. As a consequence, the model was also a bit larger in terms of the number of parameters compared to the model for the closed track.

Results

The results for the closed and open track are shown in Table 6.1. Since our system, which we named *Mumpitz*, did not predict VMWE type labels we will omit the part of the evaluation that was concerned with the type-specific performance.

Mumpitz achieved its best results on the German test set where it ranked first out of eleven teams for the token-based and fifth for the MWE-based evaluation. This is unsurprising given the fact that the features (L, LP, D) were chosen based on the performance on the German dev set. All in all, judging by the token-based evaluation Mumpitz ranks somewhere in the middle field for most of the other languages: BG: 4/10, EL: 6/11, ES: 6/10, FR: 5/13 and PT: 4/13. Polish is the only outlier where it ranks 8th out of eleven teams. However, this changes when we turn to the MWE-based evaluation where Mumpitz ranks significantly lower across the board. One reason might be that the heuristic only considers direct dependents of verbs tagged as VMWEs which, for example, leaves out modifiers and determiners of nouns and thus results only in a partial match not factored in by the MWE-based evaluation.

Our entry for the open track, Mumpitz-preinit, likewise did perform well in the token-based evaluation where it ranked first among 4 teams for German, but last for the MWE-based ranking.

System	Track	Language	MWE-based				Token-based			
			P	R	F1	Rank	P	R	F1	Rank
Mumpitz	closed	BG	75.12	46.42	57.38	6/9	86.99	48.16	62	4/10
Mumpitz	closed	DE	32.15	38.35	34.98	5/11	55.91	48	51.66	1/11
Mumpitz	closed	EL	45	30.54	36.39	8/10	73.21	36.82	49	6/11
Mumpitz	closed	ES	9.66	13	11.08	10/10	31.83	28.87	30.28	6/10
Mumpitz	closed	FR	56.8	33.53	42.17	7/12	81.25	38.86	52.57	5/13
Mumpitz	closed	PL	62.07	38.45	47.48	8/10	80.92	41.34	54.72	8/11
Mumpitz	closed	PT	44.77	47.2	45.95	7/12	63.96	52.37	57.58	4/13
Mumpitz-preinit	open	DE	43.37	36.14	39.43	4/4	70.5	44.62	54.65	1/4

Table 6.1: Language-specific results (Ehren et al., 2018).

Conclusion

Mumpitz treats MWE identification as sequence tagging problem using a BRNN with LSTM units. The features used are lemma, language-specific POS-tag and the dependency relation to the head; feature selection was conducted based on German, for which Mumpitz also obtained the highest F1 measure as to token-based classification compared to competing systems. Within the open track, we used pretrained embeddings, which lead to considerable improvements.

Judging from the the big difference in performance when we compare the token-based and MWE-based rankings, especially with respect to precision, which becomes much worse for the latter, it seems clear that the labeling scheme coupled with our simple heuristic is only a suboptimal solution. And not only the heuristic might be the problem. A binary labeling scheme presupposes that all VMWE types have something in common, since everything receives the same label. But we have seen in Section 2.5 that they have very different properties. This is why in the following identification experiments we opted for the enhanced IOB labeling scheme which also incorporates labels for the different categories.

6.1.2 BERT-based Classifier

In the this section, we describe experiments in which we compare the performance of a classifier that was trained on all VMWE types at once and classifiers which were trained only on individual VMWE types, that is we trained one classifier for VIDs, one for IRVs and so on. The goal was to explore whether performance gains could be made by choosing the latter strategy. From an architectural standpoint, the classifier is very similar to the one presented in section 6.1.1 in that it also approaches the task as sequence labeling. The main difference is that it relies on a pre-trained BERT model (Devlin et al., 2018).

Motivation

The idea to compare the performance of individual classifiers with a classifier that was trained on all VMWE types at once was mainly motivated by a MWE-related phenomenon described in section 3.2 and further elaborated in the section above: the sharing of tokens between multiple MWEs. Example (7) shows an instance for this kind of overlap, taken from the German dataset of the PARSEME ST corpus 1.2:

(7) In den [...] Bussen wurden auf den letzten Kilometern begeistert
* * * * * * * * * *
französische **Lieder angestimmt**.
* 1:VID 1;2:VPC.full
'On the busses, French songs were enthusiastically sung for the last few kilometers.'

Here, the VPC *anstimmen* ('intone') is part of the VID *Lieder anstimmen* ('sing songs')[3] and accordingly receives two distinct labels: *1* and *2:VPC.full* (separated by a semicolon). As we have learned, these PARSEME-style labels need to be converted into another annotation format like the IOB scheme in order to be suitable for supervised training. Converted to IOB format, *1;2:VPC.full* would become *I-VID;B-VPC.full*. The token *angestimmt* would receive the *I-VID* tag because it is the continuation of a VID and the *B-VPC.full* tag because it is the beginning (and end, since it is a single-token VMWE) of a VPC. Please note that this is an 'enhanced' version of the IOB scheme as it includes VMWE categories.

The problem is that even after conversion we still have two distinct labels and if the classifier cannot output more than one label per token, it cannot model this kind of overlap (Constant et al., 2017). At first, it might seem like a solution to just use the tag *I-VID;B-VPC.full* as it is and thereby encode the information that *angestimmt* belongs to two VMWEs in a single label. This method was used by the best performing system of the PARSEME shared task 1.2, MTLB-STRUCT (Taslimipoor et al., 2020). Upon inspection of one of the configuration files for the classifier, one can see that the label as a whole is converted to an integer: "I_VID;B_VPC.full": 18.[4]

The following example (8) illustrates the problem with this approach:

[3]We would contest the status of *Lieder anstimmen* as a VID, but it is an actual example from the German data set.

[4]Source: `https://github.com/shivaat/MTLB-STRUCT/blob/master/code/configs/config_DE_transferDep.json` [Accessed: 05.12.2023]

(8) Bill hoffte , mit einem Pony schneller zum Ziel kommen [sic!] ,
* * * * * * * * * *
kaufte das musikalische , welches sogleich **anstimmte** [...]
* * * * * 1:VPC.full

In Example (8), we have the same VPC *anstimmen*, but occurring on its own. In this case, the label *1:VPC.full* would become *B-VPC.full*. Thus, the two occurrences of *anstimmen* would receive two different labels which results in additional workload for the classifier because it has to learn that the same word receives two different labels depending on the context: *I-VID;B-VPC.full* if *anstimmen* is nested within a VID and *B-VPC.full* if it occurs on its own. The idea is that classifiers individually trained on VIDs and VPCs would remedy this problem to a certain extent and their results could later be merged in order to reconstruct the multi-labels. However, the extent to which this can result in improved performance obviously depends on how many overlapping VMWEs a data set contains. Furthermore, this strategy only works if the nested expressions are from different VMWE categories as the following example illustrates:

(9) Il y a lieu de prévoir une flexibilité suffisante
1:VID;2:VID 1;2 1;2 2 * * * * *
'Sufficient flexibility should be provided'

In (9), we have the two VMWEs *il y a*[5] ('there is') and *il y a lieu* ('there is need'), with the former being embedded into the latter. And since both are considered VIDs it is not possible to get rid of the multi-level tags in this case.

Labeling

While we employed binary labeling for the Mumpitz-experiments, this time we opted for the IOB scheme. This has several reasons. Firstly, using binary labeling would have made it difficult to evaluate the performance of the classifier trained on all VMWE types because the PARSEME evaluation script solely computes the F1-score for the overall performance if category labels are not provided, i.e. the performance for the individual VMWE types is not evaluated. And since we were also interested in a comparison of the performance on the type level, a different annotation scheme seemed more appropriate. A type-level binary scheme, where a label expresses if a token is part of a certain VMWE type or not,

[5]According to the PARSEME annotation guidelines an expression is automatically considered a VID if one dependent of the head verb is a non-reflexive pronoun.

would have solved this problem, but there are other issues that remain with binary labeling. One of these is the need to rely on a heuristic to convert the binary labels back to PARSEME-style labels – the possible shortcoming of this method we have seen above. But the most striking issue in our case: We loose information about overlapping VMWEs if we do not employ multi-level tags. As we have seen, this is not only relevant for the classifier trained on all types as there are also instances of overlapping VMWEs of the same category (cf. example (9)).

Hence, the choice fell naturally on the enhanced IOB scheme presented above since it has multiple levels and includes category labels. So the conversion of Example (7) would look like this:

(10) In den [...] Bussen wurden auf den letzten Kilometern
O O O O O O O O O
begeistert französische **Lieder *angestimmt***.
O O B-VID I-VID;B-VPC.full
'On the busses, French songs were enthusiastically sung for the last few kilometers.'

There is a version of the IOB format which also labels tokens in-between components of a MWE (Taslimipoor et al., 2020):

(11) I would **give** this job **a** **go**
O O B-VID o-VID o-VID I-VID I-VID

Usually, the tokens *this job* would both receive the *O*-tag but in this even more enhanced version of IOB, tokens in-between MWEs are labeled with *o-* followed by the VMWE category. We refrained from using this since the benefit from employing this strategy seems limited, in our view. In addition to a proliferation of labels, the material in-between components of different VMWEs (even if they are of the same type) is unlikely to have enough commonalities to warrant giving them the same labels.

Experimental Setup

For the following experiments we employed a multilingual pretrained BERT model that was fine-tuned for our task, a very popular approach ever since BERT appeared on the scene. As for a number of other tasks, this method proved quite successful for VMWE identification. This is underlined by two fine-tuned BERT models taking first and second place in the last PARSEME shared task on the identification of VMWEs (Taslimipoor et al., 2020; Kurfali, 2020).

As mentioned in the section introduction, the architecture is similar

to our BiLSTM architecture in that it constitutes a sequence labeling approach where every token receives a label whether it is part of a VMWE instance or not. Furthermore, it offers the same advantages as BiLSTMs in terms of bidirectionality. Figure 6.4 shows the architecture together with an example sentence. As can be seen, it is similar to Figure 6.2 except that the BiLSTM-part is exchanged by a BERT model. In addition, the input sequence not only consists of a sentence but also the special tokens <cls> and <sep> which have meaning for the BERT model as they were part of the pre-training. The special token <cls> is always the first token for every input sequence and "[t]he final hidden state corresponding to this token is used as the aggregate sequence representation for classification tasks", e.g. sentiment analysis[6] (Devlin et al., 2018). The special token <sep> on the other hand is used to separate multiple sentences in an input sequence. Both, <cls> and <sep>, are not relevant for us because we perform token-level classification on single sentences. Nevertheless, the BERT model expects them to be present for every input sequence.

We chose to employ MaChAmp (van der Goot et al., 2021) as a framework to implement this architecture. MaChAmp stands for **Ma**ssive **Ch**oice **Amp**le tasks and it is a very convenient framework for multitask learning. However, it also can be used for single-task learning which we did in our case.

```
1 Unter unter ... *
2 anderem ander ... *
3 wurde werden ... *
4-5 ins ... *
4 in in ... 1:VID
5 das der ... 1
6 Auge Auge ... 1
7 gefasst fassen ... 1
8 ... ... ... ...
```

Figure 6.3: Example for VMWE with a contraction.

There is a particularity when it comes to the PARSEME dataset we have not addressed yet, but is very relevant when using MaChAmp as a framework. And that is how contractions like the one shown in Figure

[6] In other words, the final hidden state <cls> is supposed to represent the whole sequence and it alone is fed into a classifier.

6.3 are handled. The example contains the German VID *ins Auge fassen* ('hold in the eye'⇒'consider')[7] which in turn contains *ins*, the contraction of the preposition *in* and the article *das*. Due to the fact that the preprocessing was done with UDPipe[8], these kinds of contractions are dissolved and its subtokens are added to the sentence. The original parse tree is modified to accommodate these new tokens. If originally a contraction was annotated as part of a VMWE, like in this case, the original annotation is distributed over these newly introduced subtokens. Nevertheless, the original token is being kept albeit without a token id. Instead the range of the new subtokens is given. In Example 6.3, *in* and *das* receive the ids 4 and 5 and the original token, *ins*, receives the range *4-5*. This leaves us with three options regarding the input: 1. The sentence is fed into the system as it is, which means that the input will be ungrammatical. 2. We remove the contractions. 3. The subtokens are removed, i.e. the original sentences are restored and the annotation redistributed to the contractions. In our case, the framework rules out option 1 for us, as MaChAmp expects the data to be handled according to option 2 or 3, i.e. it cannot handle the format as PARSEME provides it. We opted for option 3 as it restores the original sentence and thus is more natural. Dissolving the contractions (like in *in das Auge gefasst*) might not result in ungrammaticality, strictly speaking, but it still sounds highly marked in most cases. Option 3 has one caveat, though, which only later revealed itself: There exist official UD conversion tools[9] in order to remove the subtokens, but this introduces an error for some languages that results in erroneous annotation where the original annotation is not correctly restored. Because of this, we had to remove the data sets for Italian (IT), Hebrew (HE) and Greek (EL) from the corpus because we were not able to remove the subtokens without it resulting in erroneous data sets. And since retroactively choosing option 2 would have required training all the classifiers for the other 11 languages again, our decision remained unchanged.

Table 6.2 shows the hyperparameters used during training. These are basically the default settings for MaChAmp. We did not perform any hyperparameter tuning, since our main goal was not a new state-of-the-art but the comparison between the two different approaches.

[7]The idiom has more meanings than that, but in this particular context it means 'consider'.

[8]`https://ufal.mff.cuni.cz/udpipe` [Accessed: 04.06.2024]

[9]`https://github.com/bplank/ud-conversion-tools` [Accessed: 04.06.2024]

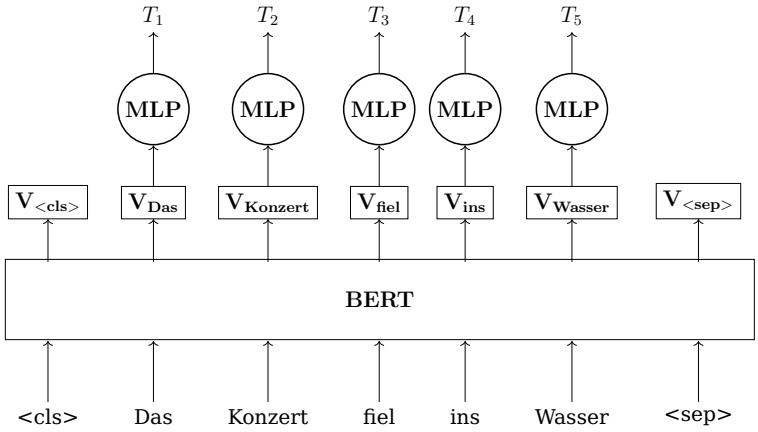


Figure 6.4: Architecture of the BERT-based model for VMWE identification.

Hyperparameter	
# Embedding dimensions	768
# Epochs	20
Objective function	Cross Entropy
Optimizer	Adam
LR schedular	slanted triangular
Batch size	32
Max. input length	128
Dropout	0.2

Table 6.2: Hyperparameters of the BERT-based model.

Results

Table 6.3 shows the overall F1 scores per language for 11 of the 14 languages of the PARSEME 1.2 corpus for the dev and the test set. To avoid cluttering the result tables – we ended up training 52 different models – we content ourselves with reporting the MWE-based F1 scores, since it was the decisive metric for the final ranking in the PARSEME shared task 1.2. The evaluation was conducted with the official PARSEME evaluation script[10]. To be able to compare the performance of the individual classifiers to the ones trained on all types, we merged the results of the former. E.g. for French five different classifiers for the VMWE types VID, IRV, LVC.full, LVC.cause and MVC were trained and their results subsequently merged. Please note that only the types VID and LVC.full were universal in that they appeared in all 11 languages and no language comprised all 8 types. To save on computation we only trained the relevant types for a given language, i.e. if a language dateset only contained very little instances of a VMWE type or no instances at all, we did not train a model for it.

When examining the results, we can see right away that we do not get a clear picture in that one or the other approach performed better across the board. For dev, the individual classifiers (*merged*) achieved better results for 7 of the 11 languages, while for test it was only 5. However, on average, the individual classifiers performed better for both dev and test, but the improvement over the *all types*-classifiers was much smaller on test (65.26 vs. 64.83) than for dev (67.66 vs. 66.00). The largest performance gains were made for Hindi (HI) with 9.7 on dev and 6.58 on test, and for Chinese (ZH) with 5.69 on dev and 3.1 on test.

Table 6.6 breaks down how the different systems performed for each individual VMWE type. If there is no entry in the table (-), then that means that a certain type was not present in the dataset and accordingly no individual model was trained. E.g. there are no particle verbs in French, so we did not train a classifier to identify them.

As before, there is no clear-cut picture as sometimes the individual classifiers and sometimes the *all types* classifiers show a stronger performance. But again, there is a slight tendency towards the individual classifiers being a bit better. For a better overview, Table 6.6 shows which approach has the better ratio per VMWE type for the test set. E.g. the *all types*-approach had a higher F1-score for 6 of 10 languages when it came to VIDs. In addition, it also has a better ratio regarding

[10] `https://gitlab.com/parseme/sharedtask-data/-/tree/master/1.2/bin` [Accessed: 04.06.2024]

	DE	EU	FR	GA	HI	PL	PT	RO	SV	TR	ZH	AVG
dev - all types	71.24	76.66	**79.91**	0.0	61.54	**80.84**	74.56	**87.58**	66.02	**67.93**	59.73	66.00
dev - merged	**71.31**	**77.51**	78.79	**3.01**	**71.24**	79.58	**74.73**	86.92	**68.72**	67.11	**65.42**	**67.66**
test - all types	67.68	**76.14**	**78.98**	0.0	57.24	**80.12**	**72.95**	**87.54**	65.64	**69.09**	57.73	64.83
test - merged	**68.06**	75.75	77.6	**1.75**	**63.82**	78.82	71.48	85.15	**67.31**	67.27	**60.83**	**65.26**

Table 6.3: Per-language and overall results.

	DE	EU	FR	GA	HI	PL	PT	RO	SV	TR	ZH
Travis-Multi	66.75	75.39	76.89	7.17	51.12	79.47	-	86.93	69.11	68.77	**70.03**
This Work	68.06	76.14	78.98	1.75	63.82	80.12	72.95	87.54	67.31	69.09	60.83
MTLB-STRUCT	**76.17**	**80.03**	**79.42**	**30.07**	**73.62**	**81.02**	**73.34**	**90.46**	**71.58**	**69.46**	69.63

Table 6.4: Comparison to similar systems.

LVC.causes (4/7 vs 3/7). But other than that, the individual classifiers outperform the *all-types* systems for every other VMWE type: IRVs (4/6 vs. 2/6), LVC.fulls (6/10 vs. 4/10), VPC.fulls (2/2 vs. 0/2), VPC.semis (2/3 vs. 1/3) and MVCs (3/4 vs 1/4).

If we compare our results to other existing systems, the performance of our models is slightly better for most languages (all but GA, SV, TR and ZH) than the runner-up system of the PARSEME shared task 1.2, Travis-Multi (Kurfali, 2020), but significantly worse than the winning system, MTLB-STRUCT (Taslimipoor et al., 2020), in terms of the Global MWE-based F1 score (cf. Table 6.4). We cannot really compare the results to those of Mumpitz because the two systems were trained on different versions of the PARSEME corpus (1.1 vs. 1.2). However, this is only a side note anyway, since our main interest lay in the comparison of our two different approaches.

Conclusion

In this section, we again treat VMWE identification as a sequence tagging problem, but this time we employ a fine-tuned multilingual BERT model instead of BiLSTMs and the IOB tagging scheme instead of a binary one. We chose this approach to compare the performance of identification models that were trained on individual VMWE types with those that were trained on all VMWE types at once. On average, the results for the models trained on individual VMWE types were slightly better with respect to the MWE-based F1 score and most VMWE types (all but VIDs and LVCs.cause) seemed to profit from the individual treatment. This was the case both for the dev and test set. However, for this quite modest improvement we had to train 52 individual classifiers compared to the 11 all-types-classifiers. Hence, it is questionable whether the huge increase in computation is justified.

This concludes our experiments in VMWE identification. For the re-

		VID	IRV	LVC.full	LVC.cause	VPC.full	VPC.semi	IAV	MVC
DE	dev - all types	**0.5650**	0.6207	0.4681	**0.6667**	0.7969	0.2727	-	-
	dev - merged	0.5411	**0.7097**	**0.4912**	0.4000	**0.8143**	0.2727	-	-
	test - all types	0.5098	0.4874	0.3597	**0.4615**	0.7473	**0.5556**	-	-
	test - merged	**0.5444**	**0.5203**	**0.4331**	0.3529	**0.7768**	0.4262	-	-
EU	dev - all types	**0.6160**	-	0.7844	**0.6809**	-	-	-	-
	dev - merged	0.6099	-	**0.8116**	0.6122	-	-	-	-
	test - all types	**0.5916**	-	0.7786	0.5134	-	-	-	-
	test - merged	0.5874	-	**0.7795**	**0.6064**	-	-	-	-
FR	dev - all types	**0.8065**	**0.8430**	**0.7439**	**0.7143**	-	-	-	**0.8000**
	dev - merged	0.7900	0.8425	0.7351	0.6667	-	-	-	0.6667
	test - all types	0.7611	**0.8741**	**0.7294**	**0.5581**	-	-	-	**0.7500**
	test - merged	**0.7825**	0.8384	0.7109	0.4444	-	-	-	0.6667
GA	dev - all types	0.0000	-	0.0000	0.0000	0.0000	0.0000	0.0000	-
	dev - merged	0.0000	-	0.0000	0.0000	0.0000	0.0000	**0.0816**	-
	test - all types	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	-
	test - merged	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	**0.0580**	-
HI	dev - all types	0.0000	-	0.6255	-	-	-	-	0.6087
	dev - merged	0.0000	-	**0.6940**	-	-	-	-	**0.7800**
	test - all types	**0.1277**	-	0.5551	0.0000	-	-	-	0.6269
	test - merged	0.0000	-	**0.6070**	0.0000	-	-	-	**0.7506**
PL	dev - all types	0.5517	**0.9019**	**0.7398**	**0.5926**	-	-	-	-
	dev - merged	**0.5647**	0.8899	0.7284	0.5714	-	-	-	-
	test - all types	0.5441	0.9100	**0.7100**	**0.5091**	-	-	-	-
	test - merged	**0.5475**	**0.9116**	0.6943	0.4086	-	-	-	-
PT	dev - all types	**0.6080**	0.7778	**0.7732**	0.2222	-	-	-	0.0000
	dev - merged	0.5839	**0.8414**	0.7556	**0.2500**	-	-	-	**0.8000**
	test - all types	**0.5966**	0.7739	**0.7604**	**0.3902**	-	-	-	0.4444
	test - merged	0.5653	**0.7891**	0.7445	0.3636	-	-	-	**0.5714**
RO	dev - all types	**0.8246**	**0.8949**	0.8704	0.8197	-	-	-	-
	dev - merged	0.8216	0.8818	**0.8889**	**0.9492**	-	-	-	-
	test - all types	**0.8483**	**0.8873**	**0.8353**	0.8880	-	-	-	-
	test - merged	0.8393	0.8547	0.8296	**0.9196**	-	-	-	-
SV	dev - all types	**0.2424**	0.7308	0.5238	-	0.6904	0.4874	-	-
	dev - merged	0.2222	**0.8235**	**0.5854**	-	**0.7379**	**0.5510**	-	-
	test - all types	**0.3755**	0.6047	0.5017	0.0000	0.7014	0.4644	-	-
	test - merged	0.2857	**0.6271**	**0.5649**	0.0000	**0.7178**	**0.5722**	-	-
TR	dev - all types	0.6404	-	0.6755	-	-	-	-	0.0000
	dev - merged	**0.6434**	-	**0.6774**	-	-	-	-	0.0000
	test - all types	**0.6482**	-	0.7097	-	-	-	-	0.0000
	test - merged	0.6117	-	**0.7206**	-	-	-	-	0.0000
ZH	dev - all types	0.0000	-	0.5714	0.2222	-	0.6379	-	0.5882
	dev - merged	**0.5161**	-	**0.5806**	**0.5714**	-	**0.6491**	-	**0.6700**
	test - all types	0.0312	-	0.5198	0.2105	-	0.5257	-	0.6542
	test - merged	**0.4646**	-	**0.5306**	**0.3448**	-	**0.5437**	-	**0.6851**

Table 6.5: Results per VMWE tpye.

	VID	IRV	LVC.full	LVC.cause	VPC.full	VPC.semi	IAV	MVC
all types	**6/10**	2/6	4/10	**4/7**	0/2	1/3	0/1	1/4
merged	4/10	**4/6**	**6/10**	3/7	**2/2**	**2/3**	1/1	**3/4**

Table 6.6: Results per VMWE type (Test).

mainder of this work, we will concern ourselves with a subtask of identification: PIE disambiguation as described in Section 3.3.

6.2 PIE Disambiguation

This chapter is concerned with the VMWE identification subtask of PIE disambiguation. For all of the following experiments we assume that another process has already identified the PIE instances and we only need to decide on the correct reading of the expression. Usually, it would make sense to also approach this task in terms of sequence labeling with an IOB labeling scheme. So the task would be to label every token in the sentence whether it is part of a PIE instance or not:

(12) The accomplice **spilled the beans**.
O O B-IDI I-IDI I-IDI

The issue with this approach in our case would be that we use lexical sample corpora which were created by extracting sentences containing instances of a pre-chosen list of candidate expressions. That means, there could be instances of other PIE types in the corpus which were not part of the candidate set. It could potentially be confusing for a system we expect to learn how to generalize well if we feed it instances of unseen PIEs not annotated as such. Consider the following example from COLF-VID:

(13) Hochqualifizierte Wissenschaftlerinnen **stehen auf der Straße**
Highly qualified female scientiests stand on the street
, ihre männlichen Kollegen **schlagen sich** munter **auf die**
, their male colleagues pat self cheerful on the
Schulter [...]
shoulder [...]
'Highly qualified female scientists are unemployed, while their male colleagues happily congratulate themselves [...].'

In this sentence, we have the two PIEs *auf der Straße stehen* ('stand on the street'⇒'to be unemployed') and *sich auf die Schulter schlagen* ('pat oneself on the back'⇒'to be proud of oneself'), but only the former was annotated because it was part of the candidate PIEs, while the latter was not. We actually tried to tackle this issue for COLF-VID by adding a fourth annotation layer which was supposed to include all the instances of unseen PIEs in the corpus. Unfortunately, the annotations proved to be of questionable quality, so we leave this for future work. Thus, the architectures presented in the following sections will all be given a head start in the sense that they always will exactly know which expressions to disambiguate.

6.2.1 BiLSTM Classifier

In this section, we describe our first experiments on PIE disambiguation with a BiLSTM-based architecture trained on COLF-VID 1.0. It is heavily based on our publication *Supervised Disambiguation of German Verbal Idioms with a BiLSTM Architecture* (Ehren et al., 2020).

Motivation

Our first model tries to leverage the fact that usually the context of a PIE instance gives valuable clues on the correct reading. As was discussed in Section 6.1.1, a straightforward choice to capture contextual information are Long short-term memory classifiers (LSTMs) (Hochreiter and Schmidhuber, 1997), which have proven well-suited for processing long sequences in a variety of tasks. Since the preceding as well as the succeeding context can give important hints, we rely on a bidirectional LSTM (BiLSTM) to contextualize our input embeddings. By contextualization we mean that we enrich the word representation given as input for a certain time-step by information concerning its left and right context, i.e. a contextualized representation ideally contains (the relevant) information which words precede and succeed it. For illustration, consider the following example:

(14) If you do not stop rocking the boat, it will CAPSIZE and we'll have to SWIM back.

In this case, our expectation would be that the embeddings associated with the components of *rock the boat* would be enriched with information that it saw *capsize* and *swim* in the preceding context as these words give valuable hints on the correct reading of this PIE instance.

System Architecture

The task is the following: Given a sentence containing a PIE instance, the system needs to predict one of four labels for this instance: LITERAL, IDIOMATIC, UNDECIDABLE or BOTH.

To put things more formally, we start with a sequence of n words $w_1, ..., w_n$ and associate every word w_i with a pretrained word embedding x_i:

$$x_i = e_{static}(w_i) \tag{6.3}$$

where e_{static} is the embedding function that gives us a static vector representation for a given word w_i. This function represents the case when we

use static word embeddings such as word2vec (Mikolov et al., 2013a,b) and fastText (Bojanowski et al., 2017).

The function $e_{context}$, on the other hand, gives us an already contextualized representation of w_i:

$$x_i = e_{context}(w_{1:n}, i) \tag{6.4}$$

In contrast to e_{static}, $e_{context}$ takes the whole sequence $w_{1:n}$ as input as well as the index for the target token. This function represents the case when we use ELMo (Peters et al., 2018) to "pre-contextualize" our input tokens.

For both cases, we receive a sequence of embeddings $x_{1:n}$ which will serve as input to the $BiLSTM$ function. Besides $x_{1:n}$, the function takes as input the index i of the word representation we want to contextualize:

$$BiLSTM(x_{1:n}, i) = LSTM_F(x_{1:i}) \circ LSTM_B(x_{i:n}) \tag{6.5}$$

where $LSTM_{\theta_F}(x_{1:i})$ is a forward LSTM that reads the input from w_1 to w_i, $LSTM_{\theta_B}(x_{i:n})$ is a backwards LSTM that reads the input from w_n to w_i and $BiLSTM(x_{1:n}, i)$ is the composition of the two. In other words, the hidden states of $LSTM_{\theta_F}(x_{1:i})$ and $LSTM_{\theta_B}(x_{i:n})$ are concatenated to form the contextualized embedding v_i, thus $BiLSTM(x_{i:n}, i) = v_i$.

After that, the contextualized representations for the noun and verb component of the PIE are concatenated and fed into a MLP with one hidden layer:

$$SCORE(v_i \circ v_j) = MLP(v_i \circ v_j)$$

where v_i and v_j are the contextualized representations for the noun and the verb, respectively. We did not include the representation of the preposition for PIE types that included a PP (for example *auf* in *auf den Zug aufspringen* ('jump on the train'⇒'follow a trend') was omitted) because in contrast to LSTMs the input size cannot vary for MLPs and some types do not contain a preposition.[11] The output of the MLP are then the scores for the four different classes.

Figure 6.5 illustrates the process with an example. First, the tokens of the sentence *Das Konzert fiel ins Wasser* ('The concert fell into the water.'⇒'The concert was canceled.') are embedded, either with e_{static} or $e_{context}$. Then this sequence of vectors is fed into the forward and backward LSTMs to produce contextualized representations. From these,

[11] We found a more elegant solution for this issue which was employed in another architecture described later on.

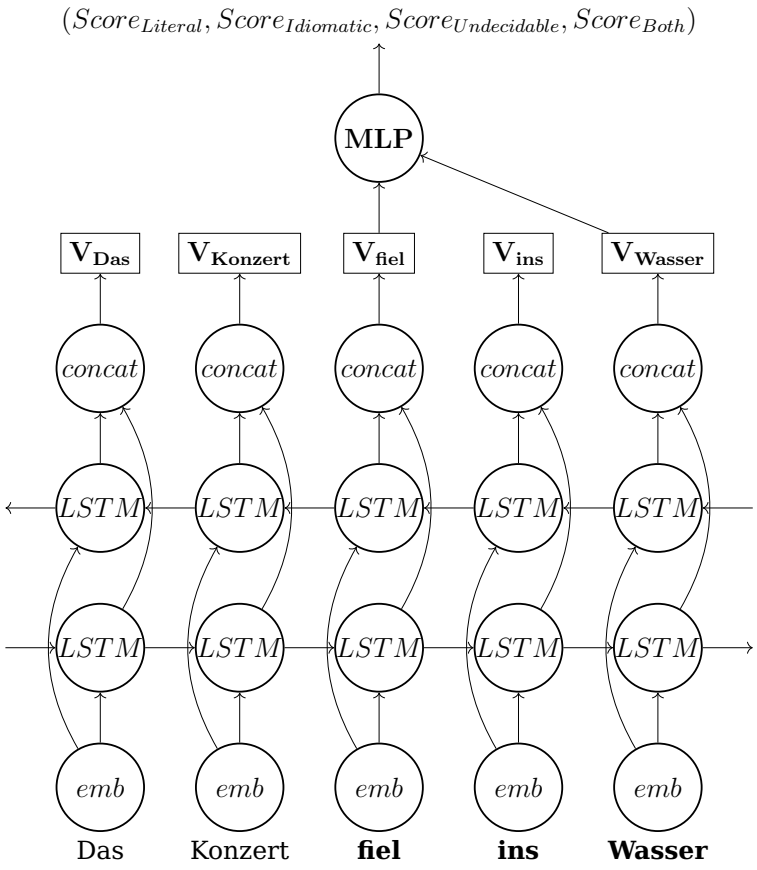


Figure 6.5: Architecture of the BiLSTM model (Ehren et al., 2020).

only the representations for the noun (*Wasser*) and verb (*fallen*) component are selected, concatenated and fed into the MLP which gives us the scores for the four classes LITERAL, IDIOMATIC, UNDECIDABLE and BOTH. This example also illustrates something else: We wanted to make unambiguously sure that our architecture performs PIE disambiguation which we achieve by basing the classifier's decisions on the contextualized embeddings of the PIE components only. In Section 3.3.1, we have seen a BiLSTM-based architecture where, we would argue, exists some uncertainty in that regard (Haagsma, 2020). We will see a further example for a similar architecture in Section 6.2.2.

We employed three different kinds of pretrained word embeddings as input to the BiLSTM: Word2vec, fastText and ELMo. The Word2vec embeddings we trained ourselves on the DECOW16 corpus (Schäfer and Bildhauer, 2012), a web corpus of shuffled sentences with over 11 billion tokens. For the fastText[12] and ELMo[13] embeddings we used existing

[12] `https://fasttext.cc/docs/en/crawl-vectors.html` [Accessed: 04.06.2024]
[13] `https://github.com/t-systems-on-site-services-gmbh/german-elmo-model`

models. The motivation behind using different embedding types was to test their impact on the results. While the word2vec embeddings were trained on lemmas and thus should not contain any information regarding morphology, the fastText representations contain subword information. This – so the hypothesis – should allow the system to pick up on the potential flexibility exhibited by literal PIE instances which could be important to determine the correct reading (cf. Section 5.2.4). The same goes for ELMo embeddings as they are character-based, but additionally they introduce contextual information already at the input level.

For the training we split the COLF-VID data according to a 70/15/15 split, hence the train set comprised 70% of the overall instances, while the dev and test set contained 15% each. One particularity of the data set we had to account for is the high variance in number of instances per PIE type. If we do no account for this, there is a high possibility that PIE types with a small number of instances are not represented in the dev or test set. This is why we had to ensure that for every type the same ratio of instances is included in the different splits. E.g. if PIE type A had a total number of 1000 instances and PIE type B only 100, then the resulting test set would contain 150 instances of type A and 15 of type B.

We trained three different models corresponding to the three different embedding types. Table 6.7 shows the hyperparameters used during training.

	Word2vec	**fastText**	**ELMo**
#Embedding dimensions	100	300	1024
#Epochs	15	15	18
Objective function	Cross Entropy		
Optimizer	Adam		
Learning rate	0.01		
Batch size	30		
#Hidden layers BiLSTM	1		
Hidden size LSTM	100		
#Hidden layers MLP	1		
Hidden size MLP	100		

Table 6.7: Hyperparameters of the BiLSTM model.

For all three embedding types the hyperparameters were the same except for the number of embedding dimensions and epochs. Of course, varying input sizes have quite a large influence on the model, since the number of parameters of the model grows with increasing input size.

[Accessed: 04.06.2024]

Dev set:

	class idiomatic			class literal			weighted macro average			
Model	**Pre**	**Rec**	**F1**	**Pre**	**Rec**	**F1**	**Pre**	**Rec**	**F1**	**Acc**
Majority baseline	75.39	100.00	85.97	0	0	0	56.78	75.32	64.75	75.32
Word2vec+LSTM+MLP	90.60	90.25	90.42	70.47	72.76	71.60	85.30	85.59	85.44	85.59
fastText+LSTM+MLP	91.77	92.85	92.31	77.41	75.20	76.29	87.86	88.14	87.99	88.14
ELMo+LSTM+MLP	90.70	96.36	93.44	85.71	70.73	77.51	89.05	89.71	89.14	89.71

Test set:

	class idiomatic			class literal			weighted macro average			
Model	**Pre**	**Rec**	**F1**	**Pre**	**Rec**	**F1**	**Pre**	**Rec**	**F1**	**Acc**
Majority baseline	76.95	100.00	86.98	0	0	0	59.22	76.95	66.93	76.95
Word2vec+LSTM+MLP	90.40	87.38	88.86	61.05	69.66	65.07	83.17	82.76	82.88	82.76
fastText+LSTM+MLP	91.23	93.94	92.56	77.42	71.79	74.50	87.45	88.29	87.83	88.29
ELMo+LSTM+MLP	93.70	93.94	93.82	78.24	79.91	79.07	89.54	90.10	89.82	90.10

Table 6.8: Evaluation results: BiLSTM Model (Ehren et al., 2020).

The dimensionality of the word representations results in a considerable difference in the number of trainable parameters for each model. While the matrix mapping the word2vec input vectors to the hidden layer has 10,000 (100×100) parameters, it has 30,000 (100×300) for the fastText-based and 102,400 for the ELMo-based model. Nevertheless, since the hidden size of the LSTMs (forward and backwards respectively) was kept constant, the input and thus parameter size of the MLP always stayed the same. The implementation of the model is available on GitHub[14].

Results

Although there were in theory four classes to predict, the models more or less completely ignored the labels UNDECIDABLE and BOTH, which is hardly surprising given the low number of instances annotated as such. So in the end, it was basically treated as a binary task by the systems, i.e. they predicted whether an instance was used literally or idiomatically.

Table 6.8 shows the precision, recall and F1 score as well as the weighted macro average (WMA) for the dev and the test set. The WMA was chosen to account for the class imbalance in the data set. The numbers for the two minority classes are not reported, since they were never predicted by the systems. As a baseline, a simple majority class classifier was used, i.e. it always predicted the label IDIOMATIC.

All three systems performed significantly better than the majority baseline. Predictably, the ELMo-based model performed the best with an F1-score of 89.14 on the dev and 89.82 on the test set. The fastText-based model follows on second place (87.99 on dev, 87.83 on test), while

[14] `https://github.com/rafehr/colf-bilstm-classifier` [Accessed: 04.06.2024]

the word2vec-based model shows the weakest performance (85.44 on dev, 82.88 on test). Our intuition was that the classifier can benefit from embeddings capturing morphosyntactic and contextual information. The former should give it the capability to (potentially) react to clues which indicate a flexibility not exhibited by VIDs. The latter resolves homonymy already at the input level which gives the system the opportunity to learn at a very early stage whether the PIE components carry their idiomatic or literal reading. However, since we did not control for dimensionality which could have a great impact as the size of the embeddings ranges from 100 to 1024 we cannot be sure if the additional number of learnable parameters is not mainly responsible for the performance gains. Additionally, the embeddings were trained on corpora of different sizes and genres which could also have a major effect on the quality of the resulting word representations. The same obviously goes for other hyperparameters like window size, epochs trained, etc. Thus, in order to investigate the impact of subword information in our embeddings we conducted a manual inspection of the results and examined whether the instances that were correctly classified by the fastText but not by the word2vec model exhibited some morphosyntactic clues which could indicate a certain reading. E.g. in Section 5.2.3 we discussed how the plural form of *Tisch* in *auf dem Tisch liegen* ('lay on the table'⇒'to be topic') is a good indicator for a literal reading, so we looked for these kinds of instances in the test set. However, in all 5 cases the word2Vec model predicted them correctly as well. Examining the 107 cases where the word2vec model made incorrect predictions and the fastText model correct ones did not yield any evidence that leads us to suspect the latter model did leverage its access to subword information, since in almost all cases the PIE instances appeared in their canonical form. Therefore, we cannot really substantiate our hypothesis that the fastText model performed better because it picked up on morphosyntactic clues.

A further observation we can make is that all models have a propensity to predict the class idiomatic, i.e. the F1 scores for the idiomatic class are much higher than for the literal class. This too is hardly surprising given the stark imbalance in classes. The systems probably learned during training that they fare better when predicting the majority class more often. Nonetheless, the F1 scores of 77.51 and 79.07 achieved by the best ELMo-based system are a respectable performance which shows that the system actually learned to distinguish the two classes and is not just a better majority baseline.

PIE-specific Results

Further proof of this can be found when examining the performance of a system per PIE type. Table 6.9 shows the precision, recall and F1 score of the ELMo-based model on the test set.
These are the 10 types with at least 10 instances the system achieved the best scores for:

- an Land ziehen: I%: 90.38 , F1: 100.00
- in eine Sackgasse geraten: I%: 98.02, F1: 100.00
- einen Nerv treffen: I%: 99.65, F1: 100.00
- über Bord werfen: I%: 87.81, F1: 98.54
- auf der Strecke bleiben: I%: 99.19, F1: 98.41
- auf den Zug aufspringen: I%: 96.03, F1: 97.30
- in den Keller gehen: I%: 72.8, F1: 94.68
- auf den Arm nehmen: I%: 42.31, F1: 92.89
- eine Brücke bauen: I%: 68.39, F1: 92.45
- die Notbremse ziehen: I%: 84.36, F1: 90.51

Of these, the six best results were achieved for types with a high idiomaticity rate (> 85), which was to be expected. However, the last four types in the top 10 have a considerably higher ratio of literal instances, which would make it much harder to achieve high scores if the system relied to heavily on the majority label. E.g. for *auf den Arm nehmen* and *eine Brücke bauen* the scores are above 90, although the idiomaticity rates are quite low with 42.31% and 68.39%, respectively. Furthermore, the fact that the relatively few literal instances of *an Land ziehen*, *im Blut haben* and *über Bord werfen* were almost all classified correctly is a good sign the system does not just apply one label all the time, even if a type has a high idiomaticity rate.

Still, arguably the most interesting VID types with respect to the disambiguation task are those with a (relatively speaking) more balanced distribution of classes, like *auf der Straße stehen, auf dem Tisch liegen, eine Brücke bauen, in den Keller gehen, im Regen stehen ins Wasser fallen, Luft holen, eine Rechnung begleichen, von Bord gehen, vor der Tür stehen, ein Zelt aufschlagen* or *über Bord gehen*, all of which have idiomaticity rates between 38.82% and 79.68%. For all but four of those expressions, the system achieves F1-scores between 82.54 and 94.45. For *ein Zelt aufschlagen* (65.08), *von Bord gehen* (70.24), *Luft holen*

Test set:

			ELMo+LSTM+MLP		
PIE	**#**	I%	**Pre**	**Rec**	**F1**
am Boden liegen	8	23.4	77.50	87.50	81.94
an Glanz verlieren	3	93.75	100.00	100.00	100.00
an Land ziehen	39	90.38	100.00	100.00	100.00
am Pranger stehen	1	100.0	100.00	100.00	100.00
den Atem anhalten	6	75.0	88.89	83.33	83.81
auf dem Abstellgleis stehen	4	42.31	56.25	75.00	64.29
auf den Arm nehmen	14	42.31	93.88	92.86	92.89
auf der Ersatzbank sitzen	4	23.81	50.00	50.00	50.00
auf der Straße stehen	38	62.4	87.30	86.84	87.00
auf der Strecke bleiben	94	99.19	97.88	98.94	98.41
auf dem Tisch liegen	143	71.29	89.13	90.21	89.44
auf den Zug aufspringen	19	96.03	100.00	94.74	97.30
eine Brücke bauen	53	68.39	92.45	92.45	92.45
die Fäden ziehen	26	94.8	90.86	88.46	89.49
im Blut haben	6	19.44	100.00	100.00	100.00
in den Keller gehen	19	72.8	95.18	94.74	94.68
in der Luft hängen	43	90.14	89.24	88.37	88.74
im Regen stehen	57	79.68	82.42	85.96	84.09
ins Rennen gehen	10	82.26	64.00	80.00	71.11
in eine Sackgasse geraten	16	98.02	100.00	100.00	100.00
im Schatten stehen	9	86.67	100.00	100.00	100.00
in Schieflage geraten	7	90.91	100.00	100.00	100.00
ins Wasser fallen	38	73.52	92.98	89.47	90.15
Luft holen	26	38.82	83.85	76.92	75.11
mit dem Feuer spielen	13	87.06	85.21	92.31	88.62
einen Nerv treffen	43	99.65	100.00	100.00	100.00
die Notbremse ziehen	49	84.36	92.09	89.80	90.51
eine Rechnung begleichen	38	64.54	78.95	78.95	78.95
von Bord gehen	14	51.61	82.86	71.43	70.24
vor der Tür stehen	90	68.17	83.20	82.22	82.54
ein Zelt aufschlagen	15	41.0	76.67	66.67	65.08
über Bord gehen	18	45.22	84.03	88.89	86.30
über Bord werfen	67	87.81	98.66	98.51	98.54
über die Bühne gehen	20	99.0	90.25	95.00	92.56

Table 6.9: Evaluation results (weighted macro) per PIE on the test set (Ehren et al., 2020).

(75.11) and *eine Rechnung begleichen* (78.95), the F1-scores are below 80. It would be interesting to investigate whether the difference in performance for the various VID types correlates with the inter-annotator agreement (IAA). We leave this question to future work.

Conclusion

In this section, we present our efforts to set a first meaningful baseline for COLF-VID. Our architecture consists of a BiLSTM whose role it is to contextualize the input and a MLP which conducts the classification by taking as input the contextualized embeddings of the verb and noun components of a PIE instance. We test three different kinds of input embeddings with ELMo achieving the best results, followed by fastText and then word2vec. However, contrary to our hypothesis that subword information could give the classifier valuable clues on the correct reading by enabling it to perceive morphological flexibility mainly exhibited by the literal counterparts of VIDs, we did not find any evidence that this is actually the case. The classifier performed way better for the idiomatic class than for the literal class, which is not surprising given the skewness of the data set. But it nevertheless achieved some good results for PIE types with a lower idiomaticity rate.

6.2.2 Shared Task

In order to invite other members of the MWE community to experiment with COLF-VID, we organized the *Shared Task on the Disambiguation of German Verbal Idioms*[15] that was held in conjunction with KONVENS 2021. The following is based on our publication *Shared task on the disambiguation of German verbal idioms at KONVENS 2021* (Ehren et al., 2021).

Related Shared Tasks

Another shared task that was also concerned with PIE disambiguation was the SemEval 2013 task on the evaluation of phrasal semantics (Korkontzelos et al., 2013). More precisely, subtask 5b, which was to decide on the compositionality of phrases in a given context. We presented the data set employed during this competition in Section 5.3. Subtask 5b had two different tracks: one for PIEs seen during training and one for PIEs unseen during training. As expected, the results for unknown phrases

[15]Again, the title was chosen before we before we adopted the term *PIE*.

were much worse and barely beat the majority baseline. Although an English and a German corpus were provided for the competition, only results for English were reported. Thus it appears, no results for the German dataset were submitted. Therefore it made a lot of sense to incorporate this data in our shared task by combining it with COLF-VID. The details of that process were also discussed in Section 5.3.

Preprocessing

After combining the COLF-VID and SemEval 5b data, it was split according to a 70/15/15 ratio with 70% of the data for training and 15% each for the dev and test set. As earlier, we had to ensure that the same ratio was applied to each type and not to the dataset as a whole in order to prevent an imbalance of types in the split data sets, since the number of examples per PIE type varies strongly. Furthermore, to align ourselves with the most recent edition of the PARSEME shared task and to challenge the ability of the systems to generalize, we added instances of 3 unseen VID types to the dev and the test set, respectively (270 to test, 268 to dev). This resulted in a train set with 6902, a dev set with 1488 and a test set with 1511 instances.

In terms of format, we decided to go with the same one the SE5b data came in, i.e. a format where every instance is represented by one tab-separated row. Figure 6.6 shows an example from the data set.

```
T890202.28.4077 \t in wasser fallen \t figuratively \t
Der Streit ums Hormonfleisch zwischen USA und EG provozierte den
Polizeieinsatz. Aber nicht nur der Steakverkauf, auch die Aktionen
gegen den Hormonstand, auf die sich Gruppen der Bauernopposition
schon vorbereitet hatten, <b>fielen</b> <b>ins</b> <b>Wasser</b>.
Die Fleischexporteure der USA wollten ihrerseits die
"Gruene Woche" zur "Aufklaerung" nutzen.
```

Figure 6.6: A sample from the shared task corpus.

The first column contains the sentence ID, followed by the PIE type, the reading and lastly the PIE instance together with its context sentences. We employed a column format in the case of COLF-VID because we had additional information in the form of lemmas and POS tags to add which is much more convenient to do in this format. The SE5b data did not come with additional information, so we thought about adding it automatically (by using UDPipe, for example), but as the SemEval data came with a lot of noise, we feared it would impede parsing and decided

	User	F1-all	F1-unseen
1.	FranziskaPannach	76.19	73.81
2.	**BiLSTM-Class. (Baseline)**	71.46	52.05
3.	JeanWayne	58.78	41.98
4.	PeterFankhauser	45.08	29.79
5.	rusaya	30.84	25.00
6.	alisentas	28.95	00.00

Table 6.10: Shared task results (Ehren et al., 2021).

against it. Since we deemed the SemEval format more convenient for relatively raw data, we decided to adopt it for the shared task and converted the COLF-VID data instead of the other way round.

Organization

In terms of evaluation we decided to rank the participating teams according to their performance with respect to the F1-score on the literal class. Furthermore, we evaluated how good the systems were in predicting the reading of the unseen PIE types that were added to the dev and test sets.

The shared task was organized on CodaLab[16], an open-source web-based platform that is widely used for machine learning competitions. Since CodaLab ran low on storage space during our shared task, we hosted the data separately on GitHub[17]. CodaLab allows for two different submission modes: either participants submit their systems or only their system outputs, where in both cases the evaluation is conducted automatically. We opted for the latter submission mechanism. A modified version of our evaluation script can be found in the GitHub repository. The training phase went from May 15 to June 23, and the evaluation phase, during which participants could submit results for up to three systems, went from June 23 to June 30.

Results

Five teams participated in the shared task and they submitted a total of 13 system prediction files. Table 6.10 shows the final results. Three of the participating teams submitted a system description paper. The

[16] https://competitions.codalab.org/competitions/31715 [Accessed: 04.06.2024]

[17] https://github.com/rafehr/vid-disambiguation-sharedtask [Accessed: 04.06.2024]

highest ranking system, *FranziskaPannach*, employed XLM-RoBERTa and a semi-automatic approach to extend the training data (Pannach and Dönicke, 2021). Surprisingly, it was the only deep learning architecture that entered the competition. The team in second place, *JeanWayne*, used a decision tree-based classifier relying on features based on the notions of similarity and concreteness (Charbonnier and Wartena, 2021). The third placed team, *PeterFankhauser*, applied a shallow, statistics-based pipeline that was previously used to detect idioms in another corpus (Amin et al., 2021). The two teams ranking last did not submit a system description paper.

When examining the results, it is salient that all systems lost a lot of ground on unseen VID types, except team *FranziskaPannach*. This was to be expected, since systems cannot rely on what types they memorized during training, but have to be able to generalize well. The same can be seen in the results of the last PARSEME shared task, where the systems were additionally evaluated based on unseen data. The winning system, MTLB-STRUCT (Taslimipoor et al., 2020), achieved a Global MWE-based F1-score of 70.14 (averaged over all languages), but only 38.53 for unseen data. The two shared tasks are designed quite differently, but we still think it highlights the challenges we face when working with small data sets. Furthermore, as discussed in Section 5.4, one of the shortcomings of COLF-VID is its low number of PIE types, which is suboptimal when we want to train a system for generalization. Even after adding the SemEval data the corpus still only has 67 PIE types. However, the winning system shows that it is possible. It was the only one whose performance on the unseen types (73.81) came close to the performance on all types (76.19). Actually, the margin is surprisingly small and another testament to the strength of architectures based on fine-tuning a large language model. To establish a baseline and to compare it to another system that leveraged information from a language model, we trained the BiLSTM-MLP architecture with ELMo embeddings presented in section 6.2.1 on the shared task data. The only difference to the architecture presented above is that, instead of concatenating the contextualized embedding for the noun and verb component of the PIE, we added and averaged the embeddings of all PIE components before feeding it to the MLP. We did this in order to tackle the shortcoming that we had to omit the prepositions for some PIEs in order to keep the input length constant. This system achieved an F1-all score of 71.46 and an F1-unseen score of 52.05 on the test set. So, although it was quite close to the winning system in terms of F1-score, its generalizing capabilities are much weaker than those of the XLM-RoBERTa system. The main reason might be that

the system of *FranziskaPannach* was fine-tuned for the task, while we only took the embeddings from ELMo to feed them into our architecture. Also, BiLSTMs have the reputation to have trouble with very long sequences (see Luong et al. (2015)) which might be a disadvantage, since every input in the shared task usually consisted of three full sentences. An attention based model as XLM-RoBERTa might be better suited for such long inputs.

There is one particularity about the winning system we should mention, however: In contrast to our system, the input to the final classifier not only consisted of the contextualized embeddings representing the components of the PIE, but the whole sequence, i.e. the sentence containing the PIE and its context sentences. This begs the question, what the system actually learned to classify: Did it really classify the PIE instances or the whole sequence instead? The former is only possible if the system learned to pay attention to the tags which mark the PIE components (). This is not completely unreasonable to assume, since they are present in every sentence. However, if it did not learn to do that, this would mean it classified the whole sequence. And this would be another explanation for the small gap between the performances on seen and unseen data: If the classifier bases its decision on the whole sequence instead of only the PIE instances, then a lot of the tokens might not really be unseen after all. But this is only speculation at this point. One possibility to examine this would be to train the system without the tags and see whether the performance changes much. If it did not change, this would of course have some interesting implications for the task as a whole.

Conclusion

In this section, we presented the *Shared Task on the Disambiguation of German Verbal Idioms* which we organized in conjunction with KONVENS 2021. We discussed details of the organization as well as the results. Five systems participated and were evaluated based on the performance on the literal class, both for all and unseen test data. Unsurprisingly, the winning system was BERT-based and outperformed all other participating systems by a large margin. It is quite remarkable, however, how small the margin between the F1-all (76.19) and the F1-unseen score (73.81) is. On the other hand, there remains the question if the winning system actually classified the PIE instances or the whole sequence. We leave the examination of this question for future work.

6.2.3 Attention model

In this section, we present our extended BiLSTM architecture which was equipped with an attention mechanism in order to gain insights into which tokens the classifier focuses the most on during PIE disambiguation. This section is based heavily on our work *An Analysis of Attention in German Verbal Idiom Disambiguation* (Ehren et al., 2022).

Motivation

The concept of attention is pervasive in deep learning at the time of this writing. In NLP, it made its first big splash in the form of the so-called Bahdanau attention (Bahdanau et al., 2014) which improved the encoder-decoder architecture for machine translation by incorporating a mechanism allowing the system to direct its attention to specific parts of the input sequence while translating. Then came the transformer (Vaswani et al., 2017), an architecture mainly based on attention, and it quickly became a landmark for deep learning. In particular, transformers are employed as a basis for large language models like BERT or, more recently, the GPT (generative pre-trained transformer) foundation models for ChatGPT (Radford et al., 2018, 2019; Brown et al., 2020). BERT's popularity even seems to have given rise to its own research field ("BERTology" (Rogers et al., 2020)) which is concerned with exploring how it works. But besides considerable performance gains, attention also allows us to attain some level of interpretability with respect to neural models we did not have before. And since for the following experiments our goal was to gain insights in what our disambiguation system is learning, we equipped it with an attention mechanism.

We would be remiss not to briefly mention the discussion to what extend attention patterns can be leveraged to make neural models "explainable". E.g. Jain and Wallace (2019) argue against it in their tellingly titled paper "Attention is not explanation". In a series of experiments on binary text classification and question answering, using BiLSTMs coupled with Bahdanau Attention, they found only a weak correlation between attention weights and other, gradient-based measures of feature importance. Furthermore, they were able to find attention distributions very different from the learned ones, which nevertheless yielded nearly identical prediction scores. From this, they conclude that attention does not provide "faithful" explanations of a model's decisions and therefore is not suitable to explain its behavior.

This is countered by Wiegreffe and Pinter (2019) who relativize these findings in their likewise tellingly titled paper "Attention is not not Expla-

nation". They reject the assumption that an attention distribution needs to be exclusive to serve as explanation. In addition, they show that even when adversarial attention distributions can be found, they do not perform as well on a simple diagnostic as their learned counterparts. They conclude that explainability depends on the definition and distinguish between plausible and faithful explanations, with the former not being invalidated by the work of Jain and Wallace.

We will not dive deeper into this discussion as it would be outside of the scope of this work, but since we are equipping our model with an attention mechanism to interpret it (to some extend), we obviously believe that attention can serve as some kind of explanation. More specifically, we believe it can answer the question which tokens have the strongest influence on the classifier's decisions. We will argue that the constraints we subject our system to are strong enough to allow for meaningful interpretation and that the resulting attention distributions support this.

Attention Terminology

Before presenting our architecture, we first establish a little bit of terminology introduced in the "Attention is all you need" paper:

> An attention function can be described as mapping a query and a set of key-value pairs to an output, where the query, keys, values, and output are all vectors. The output is computed as a weighted sum of the values, where the weight assigned to each value is computed by a compatibility function of the query with the corresponding key. (Vaswani et al., 2017, p. 3)

This terminology supposedly comes from the field of information retrieval (IR)[18] and in this context the *query* could be search terms entered by a user, the *keys* could be tags associated with documents of some kind and the *values* represent the documents themselves. It admittedly is not the best of analogies, but we can think of the query as the question where the system's attention should be directed to, while the keys constitute the potential recipients of this attention. E. g. during machine translation the query can be the previous hidden layer of the decoder representing the question, which words the model should focus on during a given time step in the translation. Naturally, the representations of the input words are then the keys. When it comes to the term "values",

[18] "Supposedly", because although many web sources claim the terminology comes from IR, it is not that easy to find IR sources that actually use this it.

the analogy gets a bit bumpy, since keys and values can actually be the same things, e.g. the keys and values in the model of Bahdanau et al. (2014) are both the contextualized input word representations. We emphasize this point because the same goes for the model presented in this section.

Attention Architecture

In this section, a model will be presented that incorporates an attention mechanism into an architecture very similar to the one introduced in section 6.2.1. The motivation for this is twofold: Most importantly, we want to explore which tokens the system is focusing on the most, when determining the reading of a PIE. Furthermore, we hope for performance gains quite similar to the ones presented in Bahdanau et al. (2014), where the proposed model significantly increased BLEU scores for longer sentences. Although we do not deem long-distance dependencies as important for our case as it is for machine translation, there are nevertheless very long sentences in COLF-VID with large gaps between the PIE components which may be problematic for a conventional LSTM.

By now there are quite a few different attention mechanisms which primarily differ in their choice of the scoring function. We opted for additive attention also employed by Bahdanau et al. (2014). Given a query $\mathbf{q} \in \mathbb{R}^q$ and a key $\mathbf{k} \in \mathbb{R}^k$ we define the following function:

$$score(\mathbf{q}, \mathbf{k}) = \mathbf{w}_v^\top \tanh(\mathbf{W}_q\mathbf{q} + \mathbf{W}_k\mathbf{k}) \in \mathbb{R} \tag{6.6}$$

where $\mathbf{W}_q \in \mathbb{R}^{h\times q}$, $\mathbf{W}_k \in \mathbb{R}^{h\times k}$, and $\mathbf{w}_v \in \mathbb{R}^h$, i.e. $\mathbf{W}_q$ and $\mathbf{W}_k$ are linear transformations that map $\mathbf{q}$ and $\mathbf{k}$ into the same space allowing us to add them. The result is then put through the $tanh$ activation before it is multiplied by $\mathbf{w}_v^\top$ so we receive a single score.

Our goal is to compute a weighted sum of the key vectors, so we have to ensure that our attentions weights sum up to one. To achieve this, we feed the scores into a softmax function:

$$\alpha(\mathbf{q}, \mathbf{k}_i) = \mathrm{softmax}(score(\mathbf{q}, \mathbf{k}_i)) = \frac{\exp(score(\mathbf{q}, \mathbf{k}_i))}{\sum_{j=1}^{m} \exp(score(\mathbf{q}, \mathbf{k}_j))} \in \mathbb{R}. \tag{6.7}$$

where m is the number of keys.

Now that we defined the way to compute the attention weights, we need to choose our keys and queries. Since the goal is to decide on the correct reading given a PIE instance in a sentence and the basis of our model is very similar to the architecture shown in figure 6.5, it makes

sense to choose the contextualized representations of the input words as keys. After all, we are interested in which input words play the most important role during the disambiguation. And because we want to know which items in the context are most important with respect to the PIE instance, it seems only natural to choose its components embeddings as the query. We chose to average those embeddings in order to always receive a vector of the same length.

Figure 6.7 illustrates the architecture with an example. As before the tokens of the sentence *Das Konzert fiel ins Wasser* ('The concert fell into the water.'⇒'The concert was canceled.') are embedded and contextualized. Then the query is build by picking the (static) embeddings of the PIE components ($e_{static}(fiel)$, $e_{static}(ins)$, $e_{static}(Wasser)$) and averaging them. This way we get a vector of the same size, no matter how many PIE components there are. This takes care of the issue that was addressed in section 6.2.1 where the representations of the prepositions where not included in the input to the MLP to keep its size constant. After building the query it is used to compute the attention score for the contextualized vectors (i.e. the hidden layers of the BiLSTM) v_i (cf. Equation 6.6). These scores are then fed into a softmax function producing attention weights that sum up to one. Next, the contextualized embeddings v_i are multiplied by the attention weights and added, i.e. we compute a weighted average:

$$c = \sum_{i=0}^{n} a_i v_i \tag{6.8}$$

The resulting vector represents the context. This context vector is concatenated with the query (the same as before) and fed into an MLP computing one score per class. So the idea for this example would be that the contextualized representation for the input *Konzert* ('concert') receives the most attention, i. e. the highest weight, because it is the only informative context word present.

The alert reader will have noticed one peculiarity about the the context vector: All contextualized embeddings are included, even the ones representing the PIE components, although they do not really belong to the context, but form the target expression itself. So why include them? Our first instinct was indeed to exclude them by setting their scores to $-\infty$ which would have resulted in their corresponding attention weights being set to zero[19]. But, as was addressed already multiple times earlier, it might not only be the context providing clues on the correct reading, but also the PIE constituents themselves by exhibiting morphosyntactic

[19]This was done to the padding tokens.

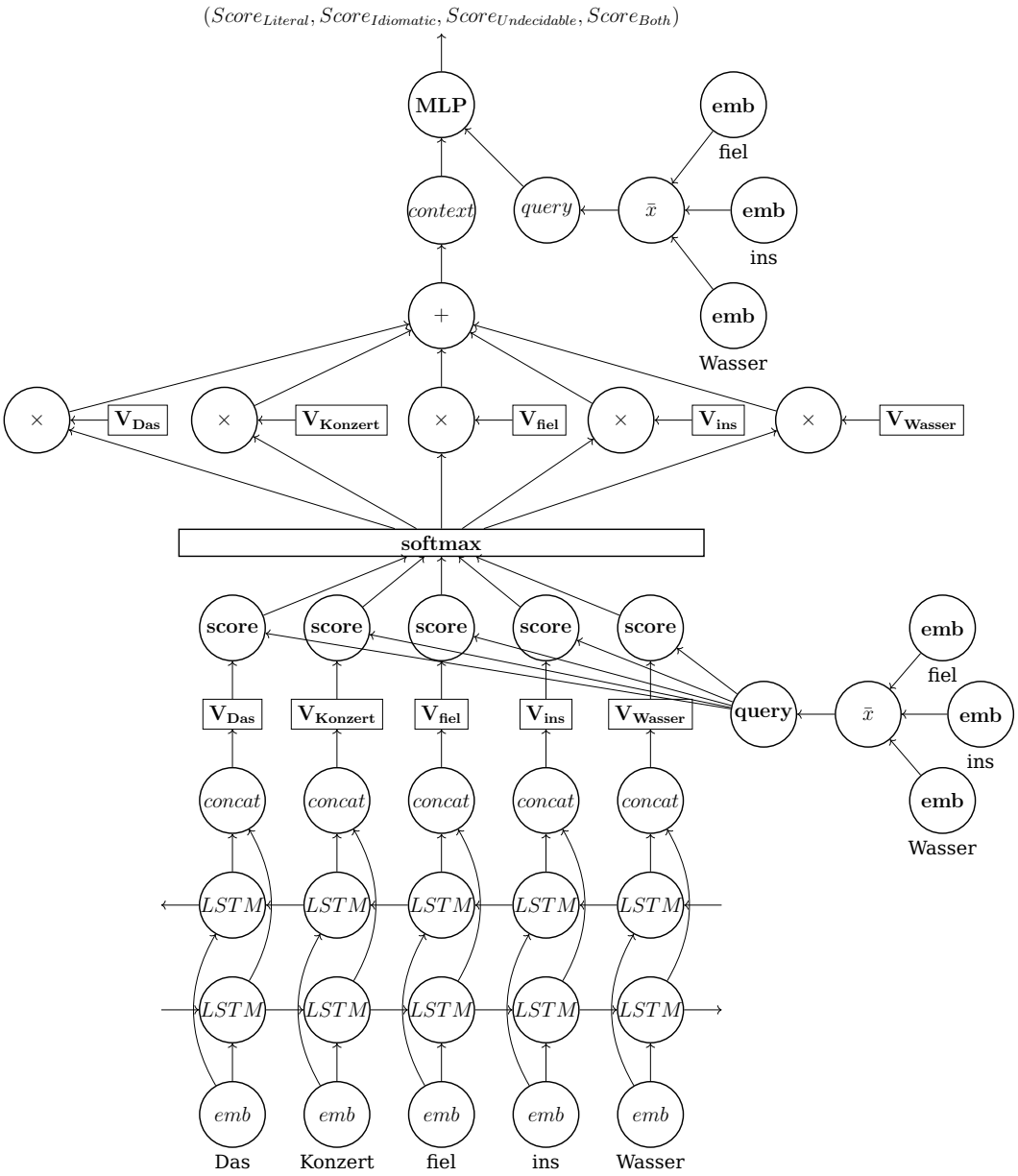


Figure 6.7: Architecture of the attention model.

#Embedding (query) size	300
#Epochs	30
Objective function	Cross Entropy
Optimizer	Adam
Learning rate	0.01
Batch size	30
#Hidden layers BiLSTM	1
Hidden size LSTM/key and value size	100/200
#Hidden layers MLP	1
Hidden size MLP	100

Table 6.11: Hyperparameters of the BiLSTM-attention model.

Weighted macro average			
Split	**Pre**	**Rec**	**F1**
Validation	87.44	87.88	87.66
Test	86.83	86.89	86.85

Table 6.12: Evaluation results of the attention model (Ehren et al., 2022).

flexibility atypical for the respective VID. Hence, if the system focuses on the PIE components instead of the context, it might be evidence the form itself gives a clue on the correct reading.

Classification Results

During the experiments presented in Section 6.2.1 we employed three different kinds of word embeddings: word2vec, fastText and ELMo. However, for now we limit ourselves to fastText embeddings because we want to make sure our model produces sensible results before moving on to more powerful word representations in future work. Furthermore, we suspect the attention distributions to be similar, even if different embeddings are used.

Table 6.11 shows the hyperparameters used during training. Since the query consists of the averaged embeddings of the PIE's components, they share the same number of dimensions, i.e. 300. The keys and values are of size 200, because the hidden layers of the forward and backward LSTM both have 100 neurons whose outputs are concatenated. Hence, the two weight matrices W_q and W_k (cf. equation 6.6), which make up the attention scoring function are of size 100×300 and 100×200, respectively. The results on the dev and test set can be seen in Table 6.12.

To our surprise, the performance was slightly worse than that of the base

model with an F1 score of 87.66 against 87.99 on the dev set and 86.89 against 87.83 on the test set. We can only speculate to why that is. One assumption is that the problem lies in the introduction of an additional 50.000 parameters in the form of the attention scoring function. Our system already has to content itself with a rather small data set, so adding new parameters might exacerbate the issue. This is supported by the fact that other parameter increasing measures during hyperparameter tuning also resulted in a considerable drop in performance. We consider this to be only a small setback, anyway, since the performance is still good enough so that it produces similar results as before and for these experiments we were more interested in the attention distributions.

The implementation of the model can be found on GitHub[20].

Extracting properties of tokens with high attention

In order to examine which parts of the input the system focuses on the most during classification, we monitor how the context vector is composed for every instance during evaluation. That means, we keep track of the attention weights a_i associated with the contextual representations v_i of the input x_i (cf. Equation 6.8). Because of the way we designed our architecture we can interpret the contextualization v_i associated with the highest weight to be of significant importance for the classification decision: Everything the MLP receives at the end are the fixed, static embedding representation (the fastText embeddings are not changed during training) and the context vector. Thus, the context vector and by extension its components with the highest weights must have considerable influence on the classifiers decision.

But what did we look for when analysing attention? For every sentence we extracted the token whose (contextualized) representation received the highest attention during evaluation. Henceforth, we will call this token the MAT (*Maximum Attention Token*). We extracted the following properties for the MAT:

- its attention weight
- its POS tag
- the label of the first arc on the dependency path between the verb component (respectively the noun component) of the PIE and the MAT

[20] `https://github.com/rafehr/PIE-attention` [Accessed: 04.06.2024]

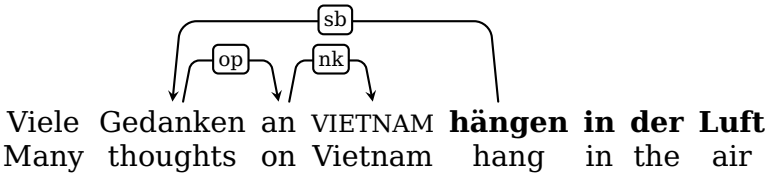


Figure 6.8: Subject (SB) relation.

The third property requires more elaboration. Our goal is to explore how the MAT relates to the verb and noun component of the PIE in terms of dependency structure. Obviously, a direct arc between the MAT and one of the PIE components does not exist in some cases, but we always have a dependency path, provided parsing was successful and did not yield a forest. We assume that the label of the first arc on this path, starting from the PIE component, does a good job characterizing the relationship between the two words. For illustration consider Figure 6.8, which shows an idiomatic instance of *in der Luft hängen* ('hang in the air'⇒'be present')[21] along with the dependency arcs of interest to us. The PIE is written in bold, the MAT is capitalized and the rest of the sentence is in normal font. There is no arc from the PIE verb, *hängen*, to the MAT (Vietnam), but there is one to the subject (*Gedanken*) which in turn governs the MAT as it is part of a PP modifying the subject. And since the MAT is part of the subject NP, we find it is appropriate to register cases like these as subject relations.

However, there was one issue regarding the way we registered dependency relations we had to attend to: In 20.38% of the cases the first arc in the undirected path from the PIE verb to the MAT is labeled *oc*, which stands for *object clause*. E.g. in Figure 6.9 the head of the PIE verb is the auxiliary *haben*, which also is the head of the subject. In such cases, we ignore all *oc* arcs and trace back, until we reach a non-*oc* label (sometimes there are multiple consecutive *oc* arcs). So for the sentence in Figure 6.9, we registered the relation *sb*. This was our way of mimicking the principle of 'primacy of content words' which is applied by the Universal Dependency (UD) community. In short, this principle states that direct dependency relations should hold between content words instead of indirect relations mediated by function words such as prepositions or auxiliaries. In hindsight, it would of course been better to use a parser trained on a UD corpus, like UDPipe, instead of using this 'work-around'. We intend to do that for future work.

In order to have those properties at our disposal, the sentences were

[21] Another meaning of *in der Luft hängen* is 'to be uncertain'.

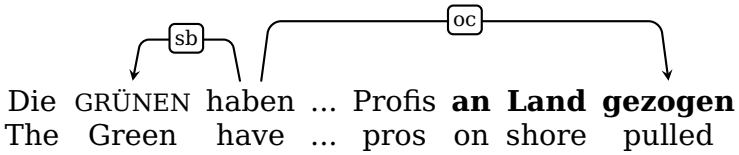


Figure 6.9: OC (object clause) relation.

parsed with spaCy[22], more specifically with the German transformer pipeline[23] which is based on BERT. It was trained on the TIGER corpus[24] and it is reported that it achieved an unlabeled attachment score (UAS) of 96 and a labeled attachment score (LAS) of 95. Even though we have to factor in some error propagation during the analysis, these performance benchmarks indicate that it should not be too big of an issue. The POS tagging was conducted with the treetagger (Schmid, 1999), which uses the STTS tag set.

Now that we know which properties we look for, what do we expect to find? For one thing, it is not unreasonable to assume the model focuses more on content words then on function words, so the strongly weighted words should have POS tags that indicate they are nouns, main verbs, adjectives, etc. Furthermore we presume that focus words have a close connection to the PIE, i. e. they are directly connected by a dependency arc. Since during the annotation process selectional preference violation was identified as one of the key factors to inform the decision whether a PIE instance was idiomatic or not, we assume *subject* and *object* could be the most frequent relations between the focus word and its head. Consider the following examples:

(15) Washington is **playing with fire**.

(16) He **rubbed** the administration **the wrong way**.

In Example (15), it is *Washington*, the subject of the sentence, that violates the selectional preferences of the verb *play*, since we are expecting an animate agent and not an institution. In Example (16), it is *administration*, the direct object, which is the giveaway because one cannot physically rub an institution.

We collected the attention statistics on the test set. A question of par-

[22] `https://spacy.io/` [Accessed: 04.06.2024]

[23] `https://spacy.io/models/de#de_dep_news_trf` [Accessed: 04.06.2024]

[24] `https://www.ims.uni-stuttgart.de/forschung/ressourcen/korpora/tiger/` [Accessed: 04.06.2024]

ticular interest to us during evaluation was whether there is a noticeable difference between the attention distribution for instances where the system predicted the label FIGURATIVE[25] (FIG) and for instances where it predicted the label LITERAL[26] (LIT). Note that we were only interested in the predictions and not the actual label because even if the prediction is wrong, the system could still have good reasons for its decision.

Finally, we performed an experiment on a subset of the data where sentences containing a nominal or a adverbial/adjectival MAT were manually manipulated. In the former case, the noun was replaced by a pronoun, while in the latter case the adverb/adjective was just deleted. We took care that the remaining sentences were still grammatical. That included replacing the whole NP the MAT belonged to and filtering out sentences, where the adjective is the predicate. Then, we let the system make predictions on these sentences and computed the resulting attention statistics, in order to see how the attention changed.

Attention Analysis

Table 6.13 shows a selection of the global attention statistics. The first column contains the numbers for cases in which the class figurative was predicted, the second column for those in which literal was predicted. The last column refers to all cases combined.

As expected, the focus of the model lies on content rather than on function words as the large majority of MATs have POS tags NN, NE, ADJA, ADJD, etc. The lion's share falls on normal nouns (NN), followed by named entities (NE) and attributive (ADJA) and adverbial/predicative adjectives (ADJD). However, there is is a considerable difference between the two classes: LIT has a significantly larger preference for ADJD than FIG (8.75 % vs. 3.07 %) and a lower preference for NN (67.92 % vs. 74.69 %). Also, the focus on NE is much higher for FIG (7.37 % vs. 3.33 %). The ratio of ADJA on the other hand is pretty much the same (6.14 % vs. 6.25 %) for both classes. What is notable is how little attention is given to verbs with only 2.36 % overall going to finite full verbs (VVFIN). We would have assumed for them to have a considerably greater influence on the classifier's decision.

Another striking difference is how more often one of the PIE components was the MAT in LIT sentences compared to FIG. A possible explanation is that the model picked up on morphosyntactic clues given by

[25] Again, we use *figurative* and *idiomatic* synonymously.

[26] The other two labels were barely predicted at all, so we did not include those in the statistics.

	FIG	LIT	overall
average MaxAttn	0.52	0.46	0.51
STD	0.2	0.18	0.2
MaxAttn on PIE verb (%)	1.23	2.92	1.6
MaxAttn on PIE noun (%)	6.51	13.75	8.11
MaxAttn on NN (%)	74.69	67.92	73.11
MaxAttn on NE (%)	7.37	3.33	6.42
MaxAttn on ADJA (%)	6.14	6.25	6.32
MaxAttn on ADJD (%)	3.07	8.75	4.34
MaxAttn on VVFIN (%)	1.84	4.17	2.36
MaxAttn on sb (%)	39.8	17.08	34.62
MaxAttn on mo (%)	25.8	41.67	29.43

Table 6.13: Selection of global attention statistics (Ehren et al., 2022).

PIEs in literal context. But this needs to be investigated further to rule out the possibility that the the PIE components are used as a fallback of some sorts, whenever the context does not contain enough information for the model to make a decision. Given the fact that we did not find any evidence for the system learning about morphosyntactic clues, the latter seems more likely.

Concerning dependency relations to the PIE verb, our hypothesis that subjects are more likely to contain a MAT holds true for FIG, but not for LIT. They are more or less mirror images of each other as for FIG, the subject clearly dominates, while with LIT, the attention model seems to favour modifying expressions to a much larger degree. The reason might be that the semantic features of the subject in idiomatic contexts are different from those usually exhibited by the PIE verb's subjects. In other words, the choice of the subject filler is more marked in figurative readings than in literal ones. For illustration, consider Example (17) which shows a usage of the VID *ins Wasser fallen* ('fall into the water'⇒'to get cancelled') with the MAT *Gewinn* ('profit')). We would assume *fallen* ('fall') usually occurs more often literally and thus with subjects that are physical entities subjected to gravity rather than an abstract entity as *profit* which is much more marked in this context.

(17) Damit **fiel** auch der GEWINN für die Verkäufer **ins**
With this **fell** also the profit for the sellers **into the**
Wasser.
water.
With this, the profit for the sellers also fell into the water .

But what do we make of the fact that for LIT we find the MAT much more often in modifying expressions than in subjects? This could be an indicator for the model picking up on morphosyntactic properties exhibited by some PIEs, e.g. the presence of a modifier indicates a literal reading because instances of the respective VID type are less flexible. A good example, where the model picked up on a modifier very informative for the classification decision is the following:

(18) **Auf** dem **Tisch** im RESTAURANT "Zeus" **liegen** dann
On the table in the restaurant "Zeus" lay then
Türschlösser und Handschellen.
door locks and handcuffs.
On the table in the restaurant "Zeus " then lie door locks and handcuffs.

Here, the modification of *to lay on the table* ('to lay on the table'⇒'to be available') by the PP *im Restaurant* is a tell-tale sign for a literal reading, since it would be odd to specify the location of a table that is not actually present. And indeed, if we search for the strings *Tisch im* ('table in the') and *Tisch in* ('table in') in the COLF-VID file for *auf dem Tisch liegen*, 9 of the 10 cases[27] were annotated as literal.

The same goes for example (19). Here we have the PIE *von Bord gehen* ('to go off board'⇒'to leave a project') which is modified with the PP *im Hafen* ('in the port'). Specifying the port where passengers go off board, if there was not actually a ship to go off board off, would not make much sense.

(19) Aber **von Bord gehen** im HAFEN von Altwarp nicht etwa
But **off board go** in the port of Altwarp not -
Besucher aus dem benachbarten Nowe Warpno.
visitors from the neighboring Nowe Warpno.
But it is not visitors from neighboring Nowe Warpno who disembark in the port of Altwarp.

Another salient observation is the magnitude of the attention given to the MAT by the system: the mean attention is 0.51 with a standard deviation of 0.2. This indicates that the attention is rather not distributed between multiple tokens. On the contrary, the model seems to pick one

[27] Literally speaking. There were exactly ten instances that matched those strings.

target that clearly stands out in terms of attention score, since, on average, MaxAttention differs considerably from the second highest attention score. The minority class LIT has a smaller MaxAttention than the majority class FIG, which seems to reflect the uncertainty of the classifier and the difficulties to identify clear indicators of LIT instances.

A further noticeable difference can be observed in the ratio of cases in which the MaxAttention is on PIE elements: again, this could be taken to speak for the uncertainty of the classifier regarding LIT instances.

Since the features we investigated above can vary considerably depending on the size of the sentence, we also plotted them against sentence length, distinguishing again between FIG and LIT.

Figures 6.10 and 6.11 show how the maximal, second highest and average attention (RestAttention, not counting maximal attention) scores develop with increasing sentence length for the two classes, respectively. The solid line is the mean, while the area surrounding it represents the 95% confidence interval. In both LIT and FIG, MaxAttention decreases with increasing sentence length, albeit Pearson's correlation coefficient is only weakly negative (overall -0.267 for sentences up to 30 tokens). Second highest attention and RestAttention remain rather stable, and in both LIT and FIG, the difference between MaxAttention and second highest attention seems pronounced, while in LIT the confidence interval almost overlaps in some areas, which is clearly not the case for FIG. Generally, second highest attention and RestAttention are relatively close. Again, the larger confidence area and the slightly (but not significantly) lower MaxAttention mean for LIT seems to suggest that the classifier is struggling more to find good indicators for LIT than for FIG, regardless of the sentence size.

The development of syntactic properties of the MAT (POS tag and dependency label) is plotted against sentence length in Figure 6.12 for FIG and in Figure 6.13 for LIT. Again, we observe very different patterns in the two cases. First, as already mentioned above in connection with Table 6.13, we see again that the subjects in figurative PIEs contain more MAT, compared to literal PIEs; for longer sentences the difference is even more striking than the overall values from Table 6.13.

A second observation is that, for LIT, modifiers (relation *mo*) quickly become more important than subjects. It is only natural that we rarely registered the *mo* relation in short sentences because, unlike the subject, which is almost always present, modifying PPs or adjectives occur only in longer sentences. For longer sentences in LIT, modifiers seem to be rather indicative for the label. This is partly due to the fact that some of the data points involve the adverbial *tief* ('deep'), which is a strong

Figure 6.10: Attention and sentence length for FIG (Ehren et al., 2022).

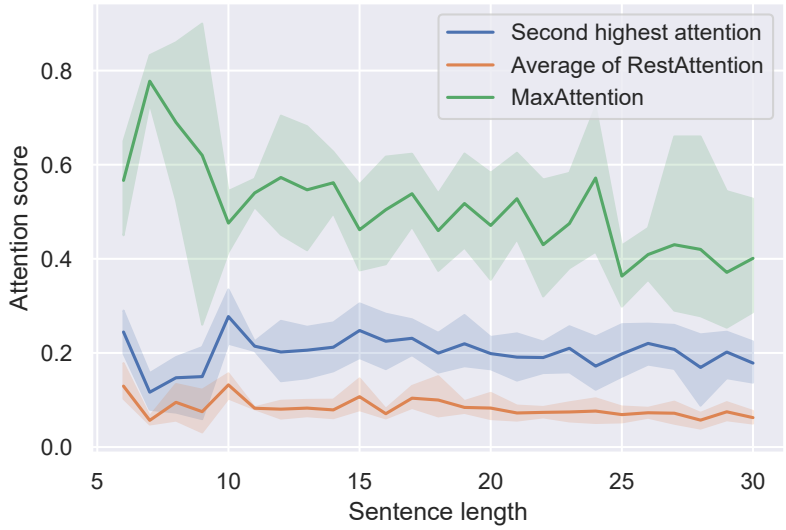


Figure 6.11: Attention and sentence length for LIT (Ehren et al., 2022).

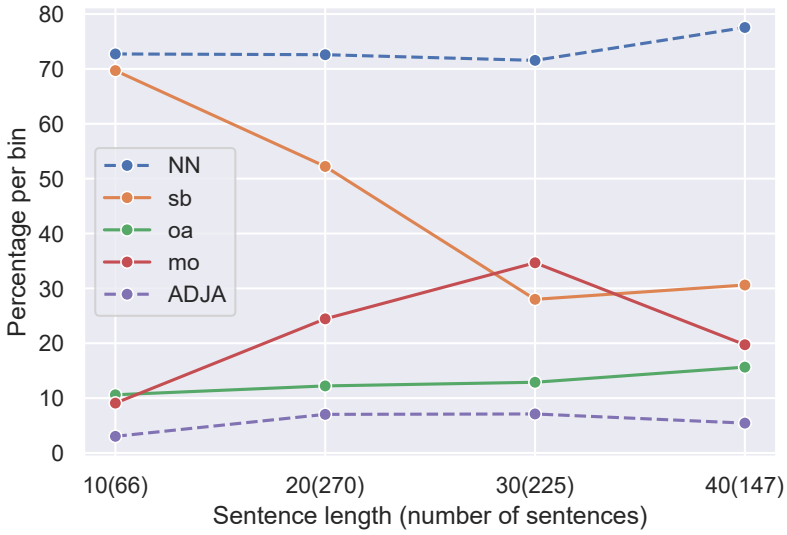


Figure 6.12: POS/dep. labels and sentence length for FIG (Ehren et al., 2022)

.

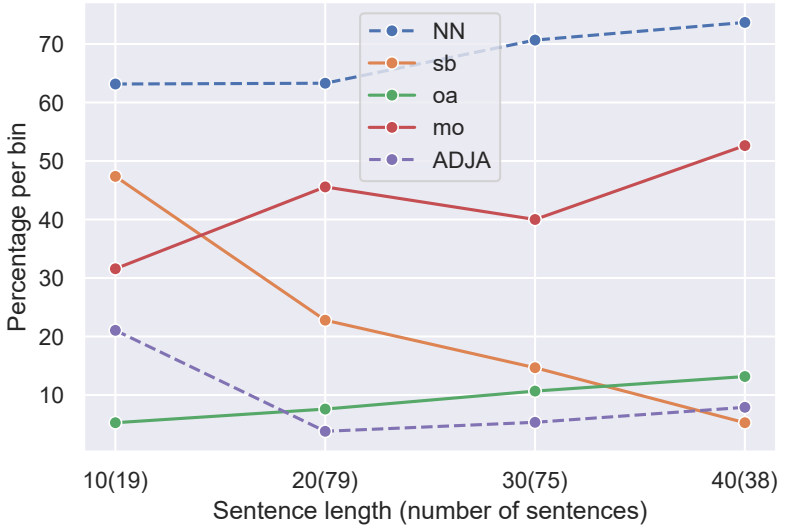


Figure 6.13: POS/dep. labels and sentence length for LIT (Ehren et al., 2022).

indicator for a literal reading when occurring with Luft holen (‘take a breath’). In 9 of 12 of those cases the system gave the highest attention to *tief*, predicting the class LIT seven times. But, as can be seen from Figure 6.13, the most frequent POS tag for MATs is NN, while ADJAs are less frequent. A manual inspection of the data suggests that NN MATs with a modifying relation to the PIE verb are often the heads of locative PPs.

Qualitative Analysis

To gain a better intuition for the attention preferences of the model, we now turn to a qualitative analysis of some of the data. We will look into examples from the perspective of an annotator in order to explore whether the system’s attention falls on tokens a human would also consider important for their decision to annotate a PIE instance in a certain way. The example sentences below are equipped with a heatmap indicating the weight distribution - the higher the attention, the more intense the color.

Example (20) shows an instance of the PIE *auf dem Tisch liegen* (‘lay on the table’⇒‘be available/be known’) with *Zahlen* (‘numbers’) as subject:

(20) *Diese Zahlen* **lagen** *am Morgen danach bereits* **auf** *Erich Honeckers* **Tisch.**
These numbers **lay** on the morning after already **on** Erich Honecker’s **table**.
‘These numbers were already reported to Erich Honecker the following morning.’

We can interpret the abstractness of the subject as an indicator for the idiomatic reading, since numbers (usually)[28] cannot be placed on a table. The model set the same focus and in four of four cases, in which *Zahlen* was the subject of *auf dem Tisch liegen*, it received the highest weight and the label *FIG* was predicted.

In (21), we have one of eight instances of the PIE *eine Brücke bauen* (‘build a bridge’), where *Brücke* (‘bridge’) was modified with the adjec-

[28] We could of course construct a context with physical representations of numbers, but this is obviously not the case here. A bigger problem is that we can interpret it metonymically with *numbers* standing for a physical report lying on someone’s table. But the annotators of COLF-VID did not follow this route and usually judged these type of instances to be figurative.

tive *goldene* (‘golden’) which gives rise to the idiomatic meaning ‘give someone an easy way to retreat’.

(21) *So werden dem künftigen Bankkunden goldene*
This way will be the future bank customer golden
Brücken *bis zu Zinssparen und Dispokredit*
bridges including interest saving and overdraft credit
gebaut.
built.
‘This way, golden bridges will be built for the future bank customer as far as interest savings and overdraft facilities.’

Since bridges are rarely built from gold, the presence of the adjective is very informative to establish the correct reading. The model did pick up on that fact as *goldene* is in the top 3 of tokens with the highest attention in seven of eight cases, predicting FIG six times.

Another adjective attracting a lot of attention is *tief* (‘deep’), when used adverbially with Luft holen (‘’take a breath’⇒‘to take a break’) as shown in (22):

(22) *Wer dort tief* **Luft holt,** *kann den Duft des Newlands*
Who there deeply **air takes**, can the smell of the Newlands
Stadium in Kapstadt einatmen [...] .
Stadium in Cape Town breathe in [...].
‘If one takes a deep breath, one can breathe in the smell of the Newlands Stadium in Cape Town.’

In 9 of 12 of those cases the system gave the highest attention to *tief*, predicting the class LIT eight times. But in contrast to the examples above, it actually is not a sure sign for a literal reading because it can just as well modify the idiomatic reading (*take a deep breath* ⇒ *take a long break*), as is represented in the test set, since 6 of the 12 instances were actually labeled as idiomatic. But since roughly 70% of instances in the training set occurring with *tief* were labeled as literal, the model reasonably predicted the label LIT.

More examples in which the model paid attention to tokens that a human annotator would also consider highly relevant for the disambiguation task can be found when examining the four literal instances of *im Blut haben* (‘have in one’s blood’⇒‘have a predisposition for sth.’) in the test set. In each of these cases, the object of the PIE, that represented a substance a person can actually have in their blood, was given the second or third highest attention (*Schadstoffe* (‘pollutants’), *Cholesterinkonzentrationen* (‘cholesterol concentration’), *Kokain* (‘cocaine’), *Alkohol* (‘al-

cohol')), while always predicting the correct reading.

Example (23) gives an example that was misclassified by the model since LIT was predicted although FIG would have been correct:

(23) *Wer hat die größte, die schönste* **Brücke gebaut?**
Who has the biggest, the most beautiful bridge built?
'Who has established the best connection?'

However, the error is understandable; without context, a human annotator would also classify (23) as LIT because of the attributes *größte* ('biggest') and *schönste* ('most beautiful') which modify *Brücke* ('bridge') (and which the attention model also focuses on).

Even though we could present many more of these types of examples, we of course do not claim that our model's decisions always correspond to the way humans would decide between LIT and FIG concerning the role that the different input tokens play for this decision. There are a lot of instances to be found where the highest weights are associated with input tokens, that – from a human perspective – do not seem to be informative for the disambiguation. This is partly due to biases from training data which distinguish our system from a human native speaker. And we cannot stress enough that we do not claim our model has "human qualities" in the sense that it understands what it is doing. But with our experiments we were able to show two things: (1) The attention distribution is not arbitrary. This is not only supported by the statistics presented above but also by a qualitative analysis of the data. (2) The relationship between the input and the output tends to be tangible and straightforward, i.e. a human can comprehend why the model focused on certain tokens. This is not self-evident, since with contextualizing models, like a BiLSTM, we cannot automatically assume that the hidden states are still faithful representations of the input tokens. It would be interesting to see whether a BERT-based encoder with its many layers would still allow for such a straightforward interpretation.

Ablation study

The goal of replacing MAT nouns with pronouns and removing MAT adverbials and adjectives is to test whether it is the grammatical function which the model likes to pay attention to, or rather some token in the context of the PIE by virtue of being a content word. Let us now inspect the attention statistics for the manipulated data. First let us look at the overall attention scores for LIT and FIG respectively (Figures 6.15 and 6.14). We can observe that the MaxAttention decreases, compared to the

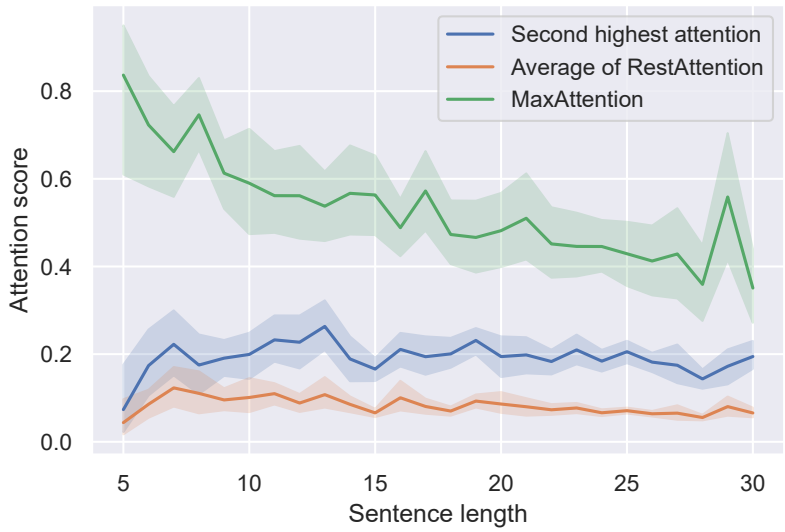


Figure 6.14: Attention and sentence length for FIG after pronominalization (Ehren et al., 2022).

original data, but the pattern basically remains intact.

Figures 6.17 and 6.16 plot the MAT's syntactic features against sentence length for the manipulated data, again for LIT and FIG respectively. A general observation in both cases is that NN POS tags and *sb* dependencies receive less attention than before, due to the pronominalization; i.e., the MaxAttention does not tend to remain on the role filled with the new pronoun (PPER is at 0.92% overall). Modifiers (*mo*), on the other hand, receive more frequently the highest attention, in particular for short sentences, probably due to the absence of other content words.

The decrease concerning *sb*, compared to the original data, is particularly marked in the FIG case. Interestingly, the attention on *oa* dependents also decreases, and attention seems to shift to modifiers. This indicates that the model pays attention to combinations of argument dependency label and content word and, in the absence of this, tends to turn to modifiers. It supports our hypothesis that, when paying attention to the subject, the model actually pays attention to semantic properties of the filler and detects mismatches between these and selectional preferences for the literal reading.

Concerning LIT, the importance of *mo* for the classifier's decision seems to increase at the cost of *sb* in shorter sentences, even though the most discriminative modifiers had been removed as well. Thus, the relatively high attention to modifiers seems to be robust, while content

Figure 6.15: Attention and sentence length for LIT after pronominalization (Ehren et al., 2022).

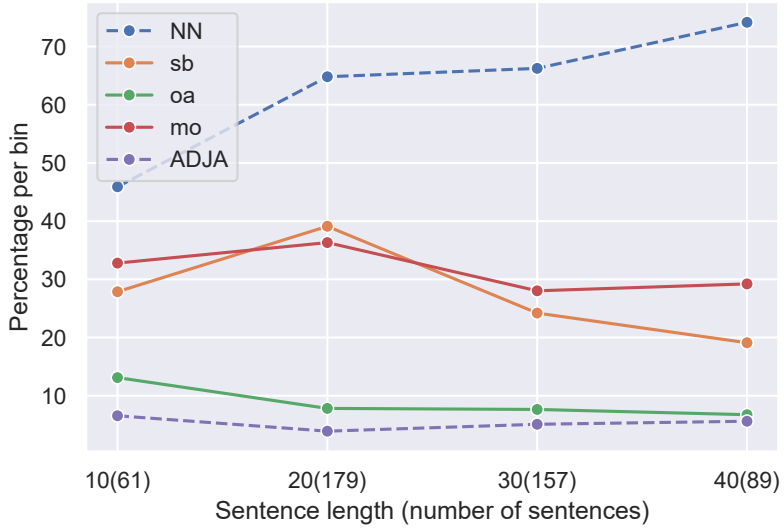


Figure 6.16: POS/dep. relation vs. sentence length for FIG after pronominalization (Ehren et al., 2022).

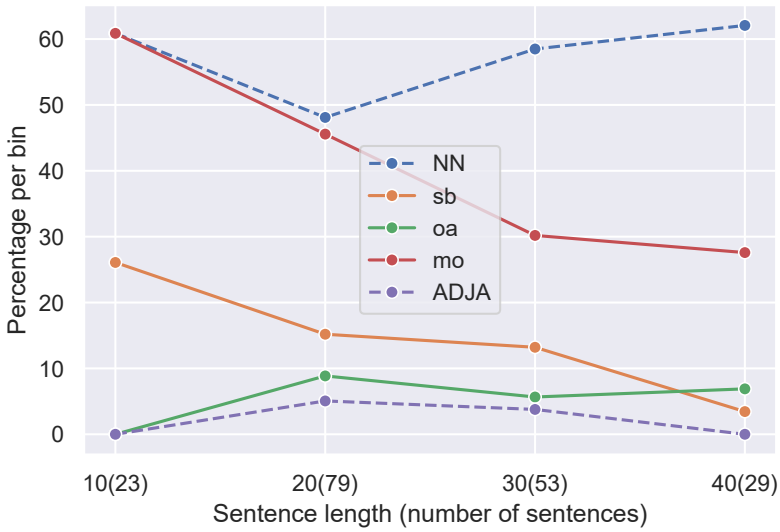


Figure 6.17: POS/dep. relation vs. sentence length for LIT after pronominalization (Ehren et al., 2022).

words are generally preferred.

(24) Nun aber ist eine Generation angetreten , die schon
Now but is a generation stepped up , that already
beim leichtesten Seegang **über Bord** zu **gehen** droht
during the slightest swell over board to go threatens
.
.
.
But now a generation has stepped up that threatens to go overboard at the slightest swell.

Conclusion

In the context of PIE disambiguation, we have provided strong evidence in support of the view that, for certain deep learning architectures, attention can be leveraged to uncover the influence of input tokens on the classifier's decision. Strikingly, regardless of classes and ablation measures, the attention model seems to pick exactly one pivotal target that clearly stands out compared to other tokens in the sentence in terms of attention scores. It would be interesting to explore, whether adversarial attention distributions in the same vein as for Jain and Wallace (2019) can be found and, if so, which properties they would reveal compared to the one presented in this paper. Regardless of the outcome of such

experiments, we would maintain that the results presented here are a valid, because plausible, explanation for the model's behaviour, since we do not agree that an attention distribution needs to be *exclusive* to serve as explanation.

Furthermore, the statistical behaviour of the studied attention model can be motivated with specific properties of the classes LIT and FIG, which differ considerably with respect to the syntactic categories that the model assigns MaxAttention to. This is even more apparent when taking sentence length into account, and also supported by an ablation test using pronominalization that we conducted. This work leaves many interesting options for future work, for example, the consideration of further linguistic features and ablation tests, crosslingual comparisons, and last but not least the comparison to other attention models such as BERT's self attention.

6.2.4 Data Augmentation via Prompting a LLM

In this last section, we want to explore the use of ChatGPT to increase our training data for classifiers trained on COLF-VID. We will present a pre-study that shows that, while having potential for creating correctly labeled additional data, there are also some shortcomings of the output produced by a GPT model such as ChatGPT concerning variability and semantic well-formedness. The result is that our very preliminary efforts of data augmentation did not lead to an improvement of the classifier's performance, on the contrary. At the end of this section, we will discuss ideas for improving on this.

Motivation

Since more data usually means better models when it comes to supervised machine learning and COLF-VID is quite modest in size, we want to explore data augmentation to enhance our corpus with more training instances.

Textual data augmentation methods range from simple techniques, such as lexical substitution or word scrambling (Wei and Zou, 2019), to the use of powerful pre-trained transformer-based models, such as BERT (Kumar et al., 2021). The newest approach is to employ ChatGPT, a transformer-based chatbot, that, at the time of this writing, is making huge waves in the field of AI because of its very impressive performance regarding text generation. E.g. (Dai et al., 2023) use ChatGPT to rephrase training instances "into multiple conceptually similar but semantically different samples". According to the authors, their results

on few-shot learning text classification tasks indicate a superior performance over other state-of-the-art text data augmentation methods.

In the same vein, we prompt ChatGPT to generate more training examples for the BiLSTM-based classifier presented in Section 6.2.1. We will discuss in the next section how we approached designing prompts that resulted in the desired output.

Prompts

Writing prompts or, in a more technical sounding term, *prompt engineering* is not (yet) an exact science. When one asks ChatGPT itself on how to write the best prompts, it gives a list of more or less general advice:

- Be clear and specific
- Provide context
- Ask direct questions
- Avoid complex or compound questions
- Use examples or scenarios
- Consider tone and style
- Be respectful and ethical
- Iterate and refine

This is not exactly a precise manual on how to write the best prompts as some of this advice is just common sense in any communication scenario ("Be clear and specific", "Avoid complex or compound questions"), but it nevertheless gives a few valuable hints which we tried to implement.

The task at hand consisted of the following subtasks:

- Generate a certain number of sentences
- All instances should contain an instance of a certain PIE type.
- These instances should either be exclusively literal or exclusively idiomatic.
- The components of the PIE type should be marked.[29]

[29]This step was necessary because our classifier needs to know the positions of the PIE components.

- The sentences should be at least 15 tokens long.

This is quite a challenging task because multiple things have to be performed at the same time and it requires a certain amount of precision (the marking of the components). Originally, ChatGPT was asked to generate literal and idiomatic instances at the same time, but this seemed to be one subtask to many, so we opted for generating exclusively literal or exclusively idiomatic instances at a time. The last condition, that the sentences should be at least 15 tokens long, was introduced when it became clear that ChatGPT preferred to generate very short and very simple sentences. Even after introducing this condition, it often was ignored.

Another issue touches on the “Iterate and refine” pointer from above. It seemed to confuse ChatGPT considerably if the same prompt was used multiple time with slight variations. It tried to adapt its behaviour, but often for the worse as it not only tried to correct the things it did wrong, but also the things it did right.

After a long period of trial and error, we opted for the following prompt (English translation below)[30]:

> Bitte gib mir 50 Sätze, in denen der Ausdruck "an Land ziehen" nicht wörtlich gebraucht wird. Dabei sollen die einzelnen Bestandteile von "an Land ziehen" mit eckigen Klammern gekennzeichnet sein. Zum Beispiel: "Nach zähem Ringen konnten sie letztendlich doch noch den Deal [an] [Land] [ziehen]." Die Sätze sollten mindestens 15 Wörter enthalten.
>
> *English translation*:
>
> Please give me 50 sentences, in which the expression "to pull ashore" is used not literally. The individual components of "to pull ashore" should be marked with brackets. For example: "After a tough fight they eventually were able to [pull] the deal [ashore]." The sentences should consist of at least 15 words.

For the generation of literal instances, we had an analogous prompt where the *not* in *not literally* was omitted. Originally, we used the term *idiomatic* instead of *not literally*, but this seemed to yield worse results.

For all of the 34 PIE types we constructed an example with a literal and an idiomatic reading, respectively, and used these to enrich the prompts.

In the next section, we will look at the data we generated from these prompts.

[30]The experiments were performed on the beginning of February 2024.

Generated Data

We collected 50 literal and 50 idiomatic instances per PIE type to gather a total of 3400 new sentences that we later added to the training data. The process of generating this data was not completely straightforward in that we could run the 68 prompts without supervision to then collect and use the data. The results were often inconsistent and some refinement was necessary to get the desired data.

A very common issue was that ChatGPT started to generate sentences in the desired reading, e.g. literal, but then all of a sudden "changed its mind" and continued generating only sentences with the other reading. In these cases, we intervened and re-started the process by repeating the prompt. Often this was enough for ChatGPT to adjust its behavior. Another less common but recurring error was that ChatGPT generated sentences, but instead of using the desired PIE type, it generated sentences with a different expression for every sentence. But none of these expressions were part of the prompt. Even when trying to correct its behavior multiple times, it did not adjust the output in any form. This was especially puzzling because this error occurred when ChatGPT already had proven for many other PIE types that it could perform the task. Even more puzzling: When we tried again with the original prompt without any modification some hours later, ChatGPT performed the task without problems.

A task-specific challenge was ChatGPT's incapability to generate examples not in canonical form. For example, we already discussed how, during the extraction phase for COLF-VID (cf. Section 5.1), we did not make it a condition that the determiner *dem* in the canonical form of *auf dem Tisch liegen* ('lay on the table'⇒'be topic') had to be present because it could have influenced the distribution of literal/idiomatic instances. Thus, in the COLF-VID data we also find a lot of examples not in canonical form, e.g. with a possessive pronoun instead the determiner like in *auf seinem/ihrem Tisch* (on his/her table). In order to get as similar as possible data, we would have preferred to not only generate examples in canonical form, but it was not possible to convey this to ChatGPT. For example, it would just ignore instructions that the determiner should not always be present in sentences generated for *auf dem Tisch liegen*. In fact, it would include the determiner, even if we excluded it in the prompt. Similar issues arose for other PIE types so we gave up on this matter at some point.

As was discussed above, we had to intervene multiple times to correct ChatGPT's behavior. The question is: When do we intervene and when do we not? If considerable efforts have to be made intervening all the time,

the suitability for data augmentation purposes comes into question. As a rule of thumb, we decided to intervene (or try again) when we knew from experience that ChatGPT could do better and we did not have to write more than five more prompts to get to the desired result. Other than that, we contended ourselves with the received output.

At first glance, the data produced by ChatGPT seems very promising. Consider the following examples for *auf dem Tisch liegen*:

(25) Die Entscheidung **liegt** nicht **auf dem Tisch**, bis alle
The decision lay not on the table, until all
relevanten Informationen vorliegen.
relevant information is available.
'No decision will be made until all relevant information is available'. IDIOMATIC

(26) Die handschriftlichen Notizen **liegen** verstreut **auf dem**
The handwritten notes lay scrambled on the
Tisch, während der Student seine Gedanken sortiert.
table, while the student his thoughts sorts.
'The handwritten notes lay scrambled on the table, while the student is sorting his thoughts.' LITERAL

Example (25) shows an idiomatic and Example (26) a literal instance for *auf dem Tisch liegen* and both resemble very much the data that can be found in COLF-VID.

However, one caveat is that the generated sentences most of the time have very similar structures. E.g. basically all 100 sentences collected for *auf dem Tisch liegen* begin the following way: DET + (ADJ) + NOUN + *lay* + ADV + *on the table*. Obviously, we would have preferred more structural variation, but ChatGPT was not able to implement the respective instructions. We also attempted to leave out the example in case ChatGPT followed it to closely, but this had little to no positive effect.

Another greater issue is that ChatGPT sometimes produces sentences which are grammatically correct, but are very marked otherwise. Consider this example for *am Boden liegen* ('lay on the ground'⇒'to be in bad shape'):

(27) Die gescheiterte Mission **lag** nach dem Fehlschlag **am**
The failed mission lies after the failure on the
Boden, aber das Team gab nicht auf.
ground, but the team gives not up.
'The failed mission is lying on the ground after the failure, but

the team does not give up.' IDIOMATIC

Although not completely unthinkable, a *mission* usually is not something one would couple with *am Boden liegen*. Usually, this expression occurs with terms such as *country* or *economy*.

Even more problematic is the following example:

(28) Die verlorene Liebe **lag** schmerzhaft **am** **Boden**, während
The lost love lay painful on the ground, while
die Erinnerungen weiterhin präsent waren.
the memories still present were.
'The lost love was lying on the floor, while the memories were still present'. IDIOMATIC

Here, we have *love* which is coupled with *am Boden liegen* and this is already very marked. But additionally, we have the adverb *schmerzhaft* (*painful*) which is something we would associate with an actual physical event (like falling to the ground) and thus could potentially lead a classifier to believe it were a literal instance. It almost seems like ChatGPT used a template for a literal sentence containing *am Boden liegen* and used it with a subject it deemed fitting.

Experiments and Results

In order to extrinsically evaluate the quality of the data produced by ChatGPT, we repeated the experiments from section 6.2.1, but added the generated sentences to the training data. More specifically, we trained the model that employed fastText embeddings as input. Before adding the data to the training split, we had to perform a bit of postprocessing because in some cases not only the PIE components in a sentence were marked. These erroneous markings were removed in a semi-automatic manner by first identifying them manually and then removing them with the help of a script[31]. After the sentences were added, the number of training instances rose from 4877 to 8277. The results can be seen in Table 6.14.

We can see that both for dev and test the (weighted macro average) F1 score drops considerably compared to before: from 87.99 to 86.82 for dev and from 87.83 to 86.14 for test. Thus, obviously the extra data did not help the classifier to increase its performance. The fact that it got considerably worse makes us conclude that the data generated by

[31]The faulty markings were quite easy to remove because ChatGPT was very consistent in what it additionally marked.

Model (split)	Pre	Rec	F1
fastText+LSTM+MLP+aug-data (dev)	86.99	87.5	86.82
fastText+LSTM+MLP (dev)	**87.56**	**88.14**	**87.99**
fastText+LSTM+MLP+aug-data (test)	86.09	86.7	86.14
fastText+LSTM+MLP (test)	**87.45**	**88.29**	**87.83**

Table 6.14: Evaluation results for data augmentation: BiLSTM Model

ChatGPT does not have the same quality actual new data would have. To some extent, we have already discussed above what its deficits are.

Although the number of instances rose by more than 50% (50.11) by adding the generated data, the number of distinct tokens in the vocabulary only increased by 15.77%, strenthening our suspicion that ChatGPT did not produce the most diverse data.

Conclusion/Future Work

Although the generated data has its obvious flaws and therefore did not lead to an increased performance, it is still quite impressive that ChatGPT is able to produce the data it did. A lot of the examples are very similar to what can be found in COLF-VID and it is pretty astonishing how it was able to even mark the PIE components in a sentence. With further improvement of these kind of models, they could be valuable sources for augmented data in the future. Or maybe more extensive prompt engineering would have resulted in better data, already. We limited ourselves to one example per PIE type, but a larger number of examples could have been helpful to the model in order to generate sentences with more syntactic variability and semantic well-formedness.

However, another problem asides from the quality of the data right now is scalability. The way things are now, ChatGPT cannot be trusted to generate the desired data without intervening once in a while. For very large amounts of data, this could prove problematic. It already would have been if the number of PIE types in COLF-VID would not have been so small. A similar approach to MAGPIE (cf. Section 4.3) with its over 1,700 PIE types is not really feasible right now.

This marks the end of the experiments section. In the last chapter, we will summarize the conclusions drawn from our work and discuss possible future work.

PART IV

Conclusions

Chapter 7

Conclusions

In this thesis, we tackled both VMWE identification and PIE disambiguation.

Concerning VMWE identification, we presented two different, albeit similar, architectures which tackled the task as a sequence labeling problem. Our first approach was based on BiLSTMs to leverage their ability to keep track of long distance dependencies which are a challenge in the context of VMWEs due to potential discontinuities. A binary labeling was used (either VMWE or not) coupled with a heuristic based on dependency graphs to detect the span of individual VMWE instances. This system entered the closed and open track of the PARSEME shared task 1.1, the main difference being that for the latter we used pretrained instead of randomly initialized embeddings. Both systems performed considerably better in the token-based evaluation than the MWE-based one. For the open track, our system even ranked first (1/4) for token- and last for MWE-based evaluation (4/4). This led us to conclude that the heuristic we used was less than ideal, so we switched to the IOB labeling scheme for our next approach. Also, we employed a fine-tuned BERT-model instead of BiLSTMs and used it to compare the effect of training individual classifiers per VMWE type instead of one classifier for all. The reason for this is that we tried to counter the issue of overlap where one token needs to receive multiple labels because some VMWE instances share tokens. On average, the results for the individual classifier-approach were slightly better with respect to the MWE-based F1-score and all VMWE categories but VIDs and LVC.causes seemed to profit from the individual treatment. However, the performance gains were so modest that the huge increase in computation seems hardly justified.

One of the main contributions of this thesis is the creation of a German corpus of PIEs (COLF-VID). For this, we annotated 6985 instances of 34 different PIE types resulting in an idiomaticity rate of 77.55% which

makes it suitable for training a classifier for PIE disambiguation. We also examined the data with respect to how it fits into the common conceptions about the flexibility of non-decomposable VIDs. We found examples to suggest that non-decomposable VIDs can show a high amount of flexibility which contradicts claims by Nunberg et al. (1994). However, from an NLP-perspective the real question is whether there is at least a tendency to be more fixed then their literal counterparts which is something that could be learned by a classifier. Since a manual inspection of the complete data would not have been feasible we tried to quantify this by employing the two variability measures proposed by Pasquer et al. (2018a). These are supposed to measure linear and syntactic variability. Regarding the former, both literal and idiomatic instances seem to be quite flexible in that they allow a lot of material in-between PIE components. Concerning syntactic similarity, almost for all examined PIE types the literal instances scored higher, suggesting that the literal instances indeed have the tendency to be more flexible.

In our first experiments on COLF-VID we again used a BiLSTM architecture. Its role was to contextualize the noun and verb components of the respective PIE types which were then fed into a MLP to perform the classification. We experimented with different types of embeddings: Word2Vec, fastText and ELMo. Unsurprisingly, ELMo performed best before fastText and Word2Vec. We hypothesized that embeddings including subword information, namely fastText and ELMo, would help capturing morphosyntactic flexibility of literal instances (like the plural form of the noun component). We did not find any evidence for this, however. The classifier performed considerably better for the idiomatic class than for the literal, but it nevertheless achieved some good results for PIE types with lower idiomaticity rates.

A slightly different version of this classifier was then later used as a baseline in the *Shared task on the disambiguation of German verbal idioms* we organized in conjunction with KONVENS 2021. For this shared task, COLF-VID was merged with parts of the German SemEval 2013 data set (Korkontzelos et al., 2013) which resulted in a higher idiomaticity rate of the final corpus. The participating systems were evaluated according to their performance on the literal class, and in a similar vein as the PARSEME shared task 1.2 (Ramisch et al., 2020) we also evaluated the performance on unseen PIE types. Predictably, a fine-tuned, BERT-based system won (Pannach and Dönicke, 2021) and interestingly showed very similar performances on the seen and the unseen data. However, there remains a question mark on what the system is actually learning, since it is a many-to-one architecture similar to text classification and we do not

have a guarantee that it actually learned PIE disambiguation. Be that as it may, it was the only participating system which was able to best our baseline classifier. However, our own classifier did perform quite poorly on the unseen data.

In our concluding experiment on PIE disambiguation, we enhanced this classifier with an attention mechanism. The goal was to explore which parts of the input our system focuses most on during classification. We evaluated the attention distribution statistically by tracking certain properties of the token with the highest attention value in a sentence, the maximum attention token (MAT). We evaluated the instances in which our system predicted the class LITERAL and IDIOMATIC, separately. We found that regardless of classes and an ablation study where we replaced MAT nouns with pronouns and removed MAT adverbials and adjectives, the attention model picks one pivotal target that stands out in terms of attention score. This pivotal target tends to be noun in most cases, again regardless of classes. For the idiomatic class, the MAT is most often part of the subject, followed by modifiers. For the literal class, it is the other way round. These results coupled with a qualitative investigation of the data, where we found that often the classifier focuses on parts of the sentence an annotator would focus on, too, leads us to conclude that the attention distribution is not arbitrary and that the relationship between the input and output tends to be tangible and straightforward. So we think, contrary to what some authors believe (Jain and Wallace, 2019), attention potentially can serve as a plausible explanation for the classifier's decisions.

Finally, we conducted a pre-study which consisted of leveraging ChatGPT to generate new training examples for a classifier trained on COLF-VID. The generated data had some flaws with respect to syntactic variability and semantic well-formedness and it did not lead to an improvement, but on the contrary, the performance of our classifier declined using this data. Also, scalability is an issue because it often was necessary to intervene during the generation process in order to stop ChatGPT from generating false data. Nevertheless, the generated data was quite impressive and often close to examples one would find in COLF-VID.

Future Work

There are multiple avenues for future work. Given the data-centric state of deep learning, one obvious one would be to extend COLF-VID with new annotated data by introducing new PIE types as well as balancing out the number of instances of the existing ones. This, of course, is

easier said than done as it requires time and money, but Haagsma (2020) showed how, during the creation of MAGPIE, it was possible to manage crowdworkers in such a way to ensure the quality of the annotations. It would be interesting to see whether this improves the performance on unseen data where our classifier performs quite poorly. However, even MAGPIE, despite its size in terms of instances and PIE types, did not perform very well on unseen data (the VNC-tokens data set; cf. Section 4.3). One reason for this might be that the MAGPIE data is quite skewed in that most PIE types are not ambiguous and classifiers trained on it will be affected by that. Thus, the goal would be to try to avoid this kind of skewness when extending the corpus.

With respect to performance on unseen data it would also make sense to examine the classifier of Pannach and Dönicke (2021) (cf. Section 6.2.2) more closely in order to determine why there is such a small gap between the performance on seen and unseen data and if their system really performs PIE disambiguation or something else. In this context, we will try and adapt our own architecture (cf. Section 6.2.1) by replacing the BiLSTM with a BERT-model and fine-tune it. This way we know for sure that we perform PIE disambiguation and nothing else as the classification decision is purely based on the PIE's components.

The relatively modest results on unseen data in the last PARSEME shared task (Ramisch et al., 2020) highlight that there is still a lot that can be improved when it comes to VMWE identification. One avenue we want to explore is the influence of parsing on VMWE identification and vice versa. To that end, we will train individual classifiers in the same vein as for our experiments in Section 6.1.2 in order to compare the influence of different VMWE types.

Bibliography

Lasha Abzianidze, Johannes Bjerva, Kilian Evang, Hessel Haagsma, Rik van Noord, Pierre Ludmann, Duc-Duy Nguyen, and Johan Bos. The Parallel Meaning Bank: Towards a Multilingual Corpus of Translations Annotated with Compositional Meaning Representations. In Mirella Lapata, Phil Blunsom, and Alexander Koller, editors, *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 242–247, Valencia, Spain, April 2017. Association for Computational Linguistics. URL `https://aclanthology.org/E17-2039`.

Alexandra Y. Aikhenvald. *Serial Verbs*. Oxford University Press, October 2018. ISBN 978-0-19-879126-3 978-0-19-183370-0.

Hazem Al Saied, Marie Candito, and Matthieu Constant. The ATILF-LLF system for parseme shared task: A transition-based verbal multiword expression tagger. In *The European Chapter of the Association for Computational Linguistics EACL 2017*, pages 127–132, 2017.

Artemis Alexiadou and Florian Schäfer. Towards a non-uniform analysis of naturally reflexive verbs. In *Proceedings of WCCFL*, volume 31, pages 1–10. Citeseer, 2014.

Miriam Amin, Peter Fankhauser, Marc Kupietz, and Roman Schneider. Shallow context analysis for german idiom detection. In *Proceedings of the shared task on the disambiguation of German verbal idioms at KONVENS 2021, Düsseldorf, Germany*. Zenodo, 2021.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

Timothy Baldwin and Su Nam Kim. Multiword expressions. *Handbook of natural language processing*, 2:267–292, 2010.

Timothy Baldwin, John Beavers, Leonoor van der Beek, Francis Bond, Dan Flickinger, and Ivan A. Sag. In Search of a Systematic Treatment of Determinerless PPs. In Patrick Saint-Dizier, editor, *Syntax and Semantics of Prepositions*, volume 29, pages 163–179. Kluwer Academic Publishers, Dordrecht, 2006. ISBN 978-1-4020-3849-5. doi: 10.1007/1-4020-3873-9_11. URL http://link.springer.com/10.1007/1-4020-3873-9_11. Series Title: Text, Speech and Language Technology.

Sascha Bargmann and Manfred Sailer. The syntactic flexibility of semantically non-decomposable idioms. *Multiword expressions: Insights from a multi-lingual perspective*, 1:1–29, 2018. Publisher: Language Science Press.

Sascha Bargmann, Berit Gehrke, and Frank Richter. Modification of literal meanings in semantically non-decomposable idioms. *Onetomany relations in morphology, syntax, and semantics*, page 245, 2021. Publisher: BoD–Books on Demand.

Bernd Bohnet. Top accuracy and fast dependency parsing is not a contradiction. In *Proceedings of the 23rd international conference on computational linguistics (coling 2010)*, pages 89–97, 2010.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017. Publisher: MIT Press.

Julia Bonn, Andrew Cowell, Jan Hajic, Alexis Palmer, Martha Palmer, James Pustejovsky, Haibo Sun, Zdenka Urešová, Shira Wein, and Nianwen Xue. UMR Annotation of Multiword Expressions. In *Proceedings of the Fourth International Workshop on Designing Meaning Representations*, pages 99–109, 2023. URL https://aclanthology.org/2023.dmr-1.10/.

Tiberiu Boroş and Ruxandra Burtica. GBD-NER at PARSEME shared task 2018: Multi-word expression detection using bidirectional long-short-term memory networks and graph-based decoding. In *Proceedings of the Joint Workshop on Linguistic Annotation, Multiword Expressions and Constructions (LAW-MWE-CxG-2018)*, pages 254–260, 2018.

Ram Boukobza and Ari Rappoport. Multi-word expression identification using sentence surface features. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 468–477, 2009.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D. Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, and Amanda Askell. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html?utm_medium=email&utm_source=transaction.

Miriam Butt. The light verb jungle: Still hacking away. *Complex predicates in cross-linguistic perspective*, pages 48–78, 2010. Publisher: Cambridge University Press Cambridge, MA.

Miriam Butt and Wilhelm Geuder. On the (semi) lexical status of light verbs. *Semilexical Categories: On the content of function words and the function of content words*, pages 323–370, 2001. Publisher: Mouton Berlin.

Miriam Butt and Aditi Lahiri. Diachronic pertinacity of light verbs. *Lingua*, 135:7–29, 2013. Publisher: Elsevier.

Marie Candito and Mathieu Constant. Strategies for contiguous multiword expression analysis and dependency parsing. In *ACL 14-The 52nd Annual Meeting of the Association for Computational Linguistics*. ACL, 2014.

Scott William Carter. *A Cold and Shallow Shore: An Oregon Coast Mystery*. Flying Raven Press, January 2022. Google-Books-ID: kmtbEAAAQBAJ.

Wallace L. Chafe. Idiomaticity as an anomaly in the Chomskyan paradigm. *Foundations of language*, pages 109–127, 1968. Publisher: JSTOR.

Jean Charbonnier and Christian Wartena. Verbal Idioms: Concrete Nouns in Abstract Contexts. In *KONVENS 2021, Düsseldorf, Germany, 06–09 September 2021*, 2021.

Mathieu Constant, Gülşen Eryiğit, Johanna Monti, Lonneke Van Der Plas, Carlos Ramisch, Michael Rosner, and Amalia Todirascu. Multiword expression processing: A survey. *Computational Linguistics*, 43(4):837–892, 2017. Publisher: MIT Press One Rogers Street, Cambridge, MA 02142-1209, USA journals-info . . . .

Matthieu Constant and Joakim Nivre. A transition-based system for joint lexical and syntactic analysis. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 161–171, 2016.

Paul Cook, Afsaneh Fazly, and Suzanne Stevenson. Pulling their weight: Exploiting syntactic forms for the automatic identification of idiomatic expressions in context. In *Proceedings of the workshop on a broader perspective on multiword expressions*, pages 41–48, 2007.

Paul Cook, Afsaneh Fazly, and Suzanne Stevenson. The VNC-tokens dataset. In *Proceedings of the LREC Workshop Towards a Shared Task for Multiword Expressions (MWE 2008)*, pages 19–22, 2008.

H. Dai, Z. Liu, W. Liao, X. Huang, Y. Cao, Z. Wu, L. Zhao, S. Xu, W. Liu, and N. Liu. AugGPT: Leveraging ChatGPT for Text Data Augmentation. arXiv 2023. *arXiv preprint arXiv:2302.13007*, 10, 2023.

Marco Damonte, Shay B. Cohen, and Giorgio Satta. An incremental parser for abstract meaning representation. *arXiv preprint arXiv:1608.06111*, 2016.

Nicole Dehé. Particle verbs in Germanic, 2015.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Mona Diab and Pravin Bhutada. Verb noun construction MWE token classification. In *Proceedings of the Workshop on Multiword Expressions: Identification, Interpretation, Disambiguation and Applications (MWE 2009)*, pages 17–22, 2009.

Laura Domine. Serial verb constructions in Japanese. page 7, 2019.

Rafael Ehren. 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, 2017.

Rafael Ehren, Timm Lichte, and Younes Samih. Mumpitz at PARSEME shared task 2018: A bidirectional LSTM for the identification of verbal multiword expressions. In *Proceedings of the Joint Workshop on Linguistic Annotation, Multiword Expressions and Constructions (LAW-MWE-CxG-2018)*, pages 261–267, 2018.

Rafael Ehren, Timm Lichte, Laura Kallmeyer, and Jakub Waszczuk. Supervised Disambiguation of German Verbal Idioms with a BiLSTM Architecture. In *Proceedings of the Second Workshop on Figurative Language Processing*, pages 211–220, 2020.

Rafael Ehren, Timm Lichte, Jakub Waszczuk, and Laura Kallmeyer. Shared task on the disambiguation of German verbal idioms at KONVENS 2021. *Proceedings of the Shared Task on the Disambiguation of German Verbal Idioms at KONVENS*, 2021.

Rafael Ehren, Laura Kallmeyer, and Timm Lichte. An Analysis of Attention in German Verbal Idiom Disambiguation. In Archna Bhatia, Paul Cook, Shiva Taslimipoor, Marcos Garcia, and Carlos Ramisch, editors, *Proceedings of the 18th Workshop on Multiword Expressions @LREC2022*, pages 16–25, Marseille, France, June 2022. European Language Resources Association. URL https://aclanthology.org/2022.mwe-1.5.

Nicholas J. Enfield. Serial verb constructions: A cross-linguistic typology. *Language*, 85(2):445–451, 2009. Publisher: Linguistic Society of America.

Thomas Ernst. Grist for the linguistic mill: Idioms and 'extra'adjectives. *Journal of Linguistic Research*, 1(3):51–68, 1981.

Stefan Evert. The statistics of word cooccurrences: word pairs and collocations. 2005.

Afsaneh Fazly and Suzanne Stevenson. Automatically constructing a lexicon of verb phrase idiomatic combinations. In *11th Conference of the European Chapter of the Association for Computational Linguistics*, 2006.

Christiane Fellbaum. How flexible are idioms? A corpus-based study. *Linguistics*, 57(4):735–767, July 2019. ISSN 1613-396X. doi: 10.1515/ling-2019-0015. URL https://www.degruyter.com/document/doi/10.1515/ling-2019-0015/html?lang=de. Publisher: De Gruyter Mouton.

Jens Fleischhauer and Thomas Gamerschlag. Deriving the meaning of light verb constructions–a frame account of German stehen 'stand'. *Yearbook of the German Cognitive Linguistics Association*, 7(1):137–156, 2019. Publisher: De Gruyter Mouton.

Vasiliki Foufi, Luka Nerima, and Éric Wehrli. Parsing and MWE Detection: Fips at the PARSEME Shared Task. In *Proceedings of the 13th Workshop on Multiword Expressions (MWE 2017)*, pages 54–59, Valencia, Spain, April 2017. Association for Computational Linguistics. doi: 10.18653/v1/W17-1706. URL https://aclanthology.org/W17-1706.

Fabienne Fritzinger, Marion Weller, and Ulrich Heid. A survey of idiomatic preposition-noun-verb triples on token level. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, 2010.

Emma Geniusiené. *The Typology of Reflexives*. De Gruyter Mouton, November 2011. ISBN 978-3-11-085911-9. doi: 10.1515/9783110859119. URL https://www.degruyter.com/document/doi/10.1515/9783110859119/html. Publication Title: The Typology of Reflexives.

Waseem Gharbieh, Virendrakumar Bhavsar, and Paul Cook. A word embedding approach to identifying verb-noun idiomatic combinations. In *Proceedings of the 12th Workshop on Multiword Expressions*, pages 112–118, 2016.

Roxana Girju, Dan Moldovan, Marta Tatu, and Daniel Antohe. On the semantics of noun compounds. *Computer speech & language*, 19(4): 479–496, 2005. Publisher: Elsevier.

Hessel Haagsma. A Bigger Fish to Fry: Scaling up the Automatic Understanding of Idiomatic Expressions. 2020. URL https://research.rug.nl/en/publications/a-bigger-fish-to-fry-scaling-up-the-automatic-understanding-of-id.

Hessel Haagsma, Malvina Nissim, and Johan Bos. 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, 2018.

Hessel Haagsma, Malvina Nissim, and Johan Bos. Casting a wide net: robust extraction of potentially idiomatic expressions. *arXiv preprint arXiv:1911.08829*, 2019.

Hessel Haagsma, Johan Bos, and Malvina Nissim. Magpie: A large corpus of potentially idiomatic expressions. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 279–287, 2020.

Chikara Hashimoto and Daisuke Kawahara. Construction of an idiom corpus and its application to idiom identification based on WSD incorporating idiom-specific features. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 992–1001, 2008.

Chikara Hashimoto, Satoshi Sato, and Takehito Utsuro. Japanese idiom recognition: Drawing a line between literal and idiomatic meanings. In *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*, pages 353–360, 2006.

M. Haspelmath. The Serial Verb Construction: Comparative Concept and Cross-linguistic Generalizations. *Language and Linguistics*, 17 (3):291–319, April 2016. ISSN 1606-822X, 2309-5067. doi: 10.1177/2397002215626895. URL http://lin.sagepub.com/lookup/doi/10.1177/2397002215626895.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. Publisher: MIT Press.

Paul J. Hopper and Elizabeth Closs Traugott. *Grammaticalization*. Cambridge University Press, 2003.

Andrea Horbach, Andrea Hensler, Sabine Krome, Jakob Prange, Werner Scholze-Stubenrecht, Diana Steffen, Stefan Thater, Christian Wellner, and Manfred Pinkal. A Corpus of Literal and Idiomatic Uses of German Infinitive-Verb Compounds. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 836–841, Portorož, Slovenia, May 2016. European Language Resources Association (ELRA). URL https://aclanthology.org/L16-1135.

Ray Jackendoff. English particle constructions, the lexicon, and the autonomy of syntax. *Verb-particle explorations*, 67:94, 2002.

Sarthak Jain and Byron C. Wallace. Attention is not Explanation. *arXiv:1902.10186 [cs]*, May 2019. URL http://arxiv.org/abs/1902.10186. arXiv: 1902.10186.

Graham Katz and Eugenie Giesbrecht. Automatic identification of non-compositional multi-word expressions using latent semantic analysis. In *Proceedings of the Workshop on Multiword Expressions: Identifying and Exploiting Underlying Properties*, pages 12–19, 2006.

Paul Kay, Ivan A. Sag, and Daniel P. Flickinger. A lexical theory of phrasal idioms. *Ms. CSLI Stanford*, 2015.

Suzanne Kemmer. The Middle Voice. *The Middle Voice*, pages 1–311, 1993. URL https://www.torrossa.com/en/resources/an/5015312. Publisher: John Benjamins Publishing Company.

Chérifa Ben Khelil, Archna Bhatia, Claire Bonial, Marie Candito, Fabienne Cap, Silvio Cordeiro, Vasiliki Foufi, Polona Gantar, Voula Giouli, Najet Hadj Mohamed, Carlos Herrero, Uxoa Iñurrieta, Mihaela Ionescu, Iskandar Keskes, Alfredo Maldonado, Verginica Mititelu, Johanna Monti, Joakim Nivre, Mihaela Onofrei, Viola Ow, Carla Parra Escartin, Manfred Sailer, Carlos Ramisch, Renata Ramisch, Monica-Mihaela Rizea, Agata Savary, Nathan Schneider, Ivelina Stoyanova, Sara Stymne, Ashwini Vaidya, Veronika Vincze, Abigail Walsh, and Hongzhi Xu. PARSEME Shared Task 1.2 - Annotation guidelines, 2022. URL https://parsemefr.lis-lab.fr/parseme-st-guidelines/1.2/?page=home.

Ryan Kiros, Yukun Zhu, Russ R. Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Skip-thought vectors. In *Advances in neural information processing systems*, pages 3294–3302, 2015.

Natalia Klyueva, Antoine Doucet, and Milan Straka. Neural networks for multi-word expression detection. In *Proceedings of the 13th workshop on multiword expressions (MWE 2017)*, pages 60–65. Association for Computational Linguistics, 2017.

Kevin Knight, Bianca Badarau, Laura Baranescu, Claire Bonial, Madalina Bardocz, Kira Griffitt, Ulf Hermjakob, Daniel Marcu, Martha Palmer, and Tim O'Gorman. Abstract meaning representation (amr) annotation release 3.0. 2021. Publisher: Abacus Data Network.

Ioannis Korkontzelos, Torsten Zesch, Fabio Massimo Zanzotto, and Chris Biemann. 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, 2013.

Sabine Krome. *Die deutsche Gegenwartssprache im Fokus korpusbasierter Lexikographie. Korpora als Grundlage moderner allgemeinsprachlicher Wörterbücher am Beispiel des WAHRIG Textkorpus digital.* Winter, 2017.

T. S. Kuhn. *The structure of scientific revolutions*. The structure of scientific revolutions. Chicago, University of Chicago Press, 1962.

Varun Kumar, Ashutosh Choudhary, and Eunah Cho. Data Augmentation using Pre-trained Transformer Models, January 2021. URL `http://arxiv.org/abs/2003.02245`. arXiv:2003.02245 [cs].

Murathan Kurfali. TRAVIS at PARSEME Shared Task 2020: How good is (m) BERT at seeing the unseen? In *International Conference on Computational Linguistics (COLING), Barcelona, Spain (Online), December 13, 2020*, pages 136–141, 2020.

Maximilian Köper and Sabine Schulte im Walde. Distinguishing literal and non-literal usage of German particle verbs. In *Proceedings of the 2016 conference of the north American chapter of the association for computational linguistics: Human language technologies*, pages 353–362, 2016.

Linlin Li and Caroline Sporleder. A cohesion graph based approach for unsupervised recognition of literal and non-literal use of multiword expressions. In *Proceedings of the 2009 Workshop on Graph-based Methods for Natural Language Processing (TextGraphs-4)*, pages 75–83, 2009.

Linlin Li, Benjamin Roth, and Caroline Sporleder. Topic models for word sense disambiguation and token-based idiom detection. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1138–1147, 2010.

Jiangming Liu, Shay B. Cohen, and Mirella Lapata. Discourse representation structure parsing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 429–439, 2018.

Joseph Lovestrand. Serial Verb Constructions. *Annual Review of Linguistics*, 7(1):109–130, 2021. doi: 10.1146/annurev-linguistics-031920-115317. URL `https://doi.org/10.1146/annurev-linguistics-031920-115317`. _eprint: https://doi.org/10.1146/annurev-linguistics-031920-115317.

Minh-Thang Luong, Hieu Pham, and Christopher D. Manning. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*, 2015.

Alfredo Maldonado, Lifeng Han, Erwan Moreau, Ashjan Alsulaimani, Koel Chowdhury, Carl Vogel, and Qun Liu. Detection of verbal multi-word expressions via conditional random fields with syntactic dependency features and semantic re-ranking. In *Proceedings of the 13th Workshop on multiword expressions (MWE 2017)*, pages 114–120. Association for Computational Linguistics, 2017.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013a.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013b.

István Nagy T and Veronika Vincze. Vpctagger: Detecting verb-particle constructions with syntax-based methods. Association for Computational Linguistics, 2014.

Geoffrey Nunberg, Ivan A. Sag, and Thomas Wasow. Idioms. *Language*, 70(3):491–538, 1994. Publisher: Linguistic Society of America.

Franziska Pannach and Tillmann Dönicke. Cracking a walnut with a sledgehammer: XLM-RoBERTa for German verbal idiom disambiguation tasks. *Proceedings of the Shared Task on the Disambiguation of German Verbal Idioms at KONVENS*, 2021.

Caroline Pasquer, Agata Savary, Jean-Yves Antoine, and Carlos Ramisch. Towards a Variability Measure for Multiword Expressions. In Marilyn Walker, Heng Ji, and Amanda Stent, editors, *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 426–432, New Orleans, Louisiana, June 2018a. Association for Computational Linguistics. doi: 10.18653/v1/N18-2068. URL https://aclanthology.org/N18-2068.

Caroline Pasquer, Agata Savary, Carlos Ramisch, and Jean-Yves Antoine. 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, August 2018b. Association for Computational Linguistics. URL https://aclanthology.org/C18-1219.

Jing Peng, Anna Feldman, and Ekaterina Vylomova. Classifying Idiomatic and Literal Expressions Using Topic Models and Intensity of Emotions. In Alessandro Moschitti, Bo Pang, and Walter Daelemans, editors, *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2019–2027, Doha, Qatar, October 2014. Association for Computational Linguistics. doi: 10.3115/v1/D14-1216. URL https://aclanthology.org/D14-1216.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014. URL https://aclanthology.org/D14-1162.pdf.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. *arXiv:1802.05365 [cs]*, March 2018. URL http://arxiv.org/abs/1802.05365. arXiv: 1802.05365.

Thomas Pickard. Comparing word2vec and GloVe for Automatic Measurement of MWE Compositionality. In *Proceedings of the Joint Workshop on Multiword Expressions and Electronic Lexicons*, pages 95–100, online, December 2020. Association for Computational Linguistics. URL https://aclanthology.org/2020.mwe-1.12.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. 2018. URL https://www.mikecaptain.com/resources/pdf/GPT-1.pdf. Publisher: OpenAI.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019. URL https://insightcivic.s3.us-east-1.amazonaws.com/language-models.pdf.

Carlos Ramisch, Silvio Cordeiro, Agata Savary, Veronika Vincze, Verginica Mititelu, Archna Bhatia, Maja Buljan, Marie Candito, Polona Gantar, and Voula Giouli. 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. Association for Computational Linguistics, 2018.

Carlos Ramisch, Agata Savary, Bruno Guillaume, Jakub Waszczuk, Marie Candito, Ashwini Vaidya, Verginica Barbu Mititelu, Archna Bhatia, Uxoa Iñurrieta, and Voula Giouli. Edition 1.2 of the PARSEME shared task on semi-supervised identification of verbal multiword expressions. In *Proceedings of the Joint Workshop on Multiword Expressions and Electronic Lexicons*, pages 107–118, 2020.

Lance A. Ramshaw and Mitchell P. Marcus. Text Chunking using Transformation-Based Learning, May 1995. URL `http://arxiv.org/abs/cmp-lg/9505040`. arXiv:cmp-lg/9505040.

Siva Reddy, Diana McCarthy, and Suresh Manandhar. An empirical study on compositionality in compound nouns. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 210–218, 2011.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. A primer in bertology: What we know about how bert works. *Transactions of the Association for Computational Linguistics*, 8:842–866, 2020. Publisher: MIT Press.

Ivan A. Sag, Timothy Baldwin, Francis Bond, Ann Copestake, and Dan Flickinger. Multiword expressions: A pain in the neck for NLP. In *International conference on intelligent text processing and computational linguistics*, pages 1–15. Springer, 2002.

Hazem Al Saied, Marie Candito, and Matthieu Constant. A transition-based verbal multiword expression analyzer. In *Multiword expressions at length and in depth*, pages 209–226. Language Science Press, Berlin, October 2018. ISBN 978-3-96110-123-8. doi: 10.5281/zenodo.1469561. URL `https://zenodo.org/record/1469561`.

Giancarlo Salton, Robert Ross, and John Kelleher. 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, 2016.

Satoshi Sato. Compilation of a comparative list of basic Japanese idioms from five sources. *Information Processing Society of Japan SIG Notes, NL-178*, 1:1–6, 2007.

Agata Savary, Manfred Sailer, Yannick Parmentier, Michael Rosner, Victoria Rosén, Adam Przepiórkowski, Cvetana Krstev, Veronika Vincze, Beata Wójtowicz, and Gyri Smørdal Losnegaard. PARSEME–PARSing and Multiword Expressions within a European multilingual network. In

7th Language & Technology Conference: Human Language Technologies as a Challenge for Computer Science and Linguistics (LTC 2015), 2015.

Agata Savary, Carlos Ramisch, Silvio Ricardo Cordeiro, Federico Sangati, Veronika Vincze, Behrang Qasemi Zadeh, Marie Candito, Fabienne Cap, Voula Giouli, and Ivelina Stoyanova. The PARSEME shared task on automatic identification of verbal multiword expressions. In *The 13th Workshop on Multiword Expression at EACL*, pages 31–47, 2017.

Agata Savary, Silvio Cordeiro, Timm Lichte, Carlos Ramisch, Uxoa Iñurrieta, and Voula Giouli. Literal occurrences of multiword expressions: rare birds that cause a stir. *The Prague Bulletin of Mathematical Linguistics*, 2019.

Agata Savary, Chérifa Ben Khelil, Carlos Ramisch, Voula Giouli, Verginica Barbu Mititelu, Najet Hadj Mohamed, Cvetana Krstev, Chaya Liebeskind, Hongzhi Xu, and Sara Stymne. PARSEME corpus release 1.3. In *Proceedings of the 19th Workshop on Multiword Expressions (MWE 2023)*, pages 24–35, 2023. URL https://aclanthology.org/2023.mwe-1.6/.

Helmut Schmid. Improvements in part-of-speech tagging with an application to German. In *Natural language processing using very large corpora*, pages 13–25. Springer, 1999.

Nathan Schneider and Noah A. Smith. A corpus and model integrating multiword expressions and supersenses. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1537–1547, 2015.

Nathan Schneider, Emily Danchik, Chris Dyer, and Noah A. Smith. Discriminative Lexical Semantic Segmentation with Gaps: Running the MWE Gamut. *Transactions of the Association for Computational Linguistics*, 2:193–206, December 2014a. ISSN 2307-387X. doi: 10.1162/tacl_a_00176. URL https://direct.mit.edu/tacl/article/43301.

Nathan Schneider, Spencer Onuffer, Nora Kazour, Emily Danchik, Michael T. Mordowanec, Henrietta Conrad, and Noah A. Smith. Comprehensive annotation of multiword expressions in a social web corpus. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation*, pages

455–461. European Language Resources Association (ELRA), 2014b. URL https://www.research.ed.ac.uk/en/publications/comprehensive-annotation-of-multiword-expressions-in-a-social-web.

Florian Schäfer. The passive of reflexive verbs and its implications for theories of binding and case. *The Journal of Comparative Germanic Linguistics*, 15(3):213–268, 2012. Publisher: Springer.

Roland Schäfer and Felix Bildhauer. Building large corpora from the web using a new efficient tool chain. In *Proceedings of the eighth international conference on language resources and evaluation (LREC'12)*, pages 486–493, 2012.

Djamé Seddah, Reut Tsarfaty, Sandra Kübler, Marie Candito, Jinho D. Choi, Richárd Farkas, Jennifer Foster, Iakes Goenaga, Koldo Gojenola Galletebeitia, and Yoav Goldberg. Overview of the SPMRL 2013 shared task: A cross-framework evaluation of parsing morphologically rich languages. In *Proceedings of the Fourth Workshop on Statistical Parsing of Morphologically-Rich Languages*, pages 146–182. Association for Computational Linguistics, 2013.

Melanie Seiss. On the Difference between Auxiliaries, Serial Verbs and Light Verbs. *Proceedings of the LFG09 Conference*, 2009. Stanford, CA: csli Publications.

Katalin Ilona Simkó, Viktória Kovács, and Veronika Vincze. USzeged: Identifying Verbal Multiword Expressions with POS Tagging and Parsing Techniques. In *Proceedings of the 13th Workshop on Multiword Expressions (MWE 2017)*, pages 48–53, Valencia, Spain, April 2017. Association for Computational Linguistics. doi: 10.18653/v1/W17-1705. URL https://aclanthology.org/W17-1705.

Caroline Sporleder and Linlin Li. 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, 2009.

Caroline Sporleder, Linlin Li, Philip Gorinski, and Xaver Koch. Idioms in context: The idix corpus. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, 2010.

Suzanne Stevenson, Afsaneh Fazly, and Ryan North. Statistical measures of the semi-productivity of light verb constructions. In *Proceedings of the Workshop on Multiword Expressions: Integrating Processing*, pages 1–8, 2004.

Regina Stodden, Behrang QasemiZadeh, and Laura Kallmeyer. TRAPACC and TRAPACCS at PARSEME Shared Task 2018: Neural Transition Tagging of Verbal Multiword Expressions. In *Proceedings of the Joint Workshop on Linguistic Annotation, Multiword Expressions and Constructions (LAW-MWE-CxG-2018)*, pages 268–274, Santa Fe, New Mexico, USA, August 2018. Association for Computational Linguistics. URL https://aclanthology.org/W18-4930.

Milan Straka. UDPipe 2.0 prototype at CoNLL 2018 UD shared task. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 197–207, 2018. URL https://aclanthology.org/K18-2020/.

Shiva Taslimipoor and Omid Rohanian. SHOMA at Parseme Shared Task on Automatic Identification of VMWEs: Neural Multiword Expression Tagging with High Generalisation, September 2018. URL http://arxiv.org/abs/1809.03056. arXiv:1809.03056 [cs].

Shiva Taslimipoor, Sara Bahaadini, and Ekaterina Kochmar. MTLB-STRUCT @Parseme 2020: Capturing Unseen Multiword Expressions Using Multi-task Learning and Pre-trained Masked Language Models. In *Proceedings of the Joint Workshop on Multiword Expressions and Electronic Lexicons*, pages 142–148, online, December 2020. Association for Computational Linguistics. URL https://aclanthology.org/2020.mwe-1.19.

Stefan Thim. *Phrasal Verbs: The English Verb-Particle Construction and its History*. De Gruyter Mouton, October 2012. ISBN 978-3-11-025703-8. doi: 10.1515/9783110257038. URL https://www.degruyter.com/document/doi/10.1515/9783110257038/html. Publication Title: Phrasal Verbs.

Rob van der Goot, Ahmet Üstün, Alan Ramponi, Ibrahim Sharaf, and Barbara Plank. Massive Choice, Ample Tasks (MaChAmp): A Toolkit for Multi-task Learning in NLP, March 2021. URL http://arxiv.org/abs/2005.14672. arXiv:2005.14672 [cs].

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, \Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Veronika Vincze, János Zsibrita, and István Nagy. Dependency parsing for identifying Hungarian light verb constructions. In *Proceedings of the*

Sixth International Joint Conference on Natural Language Processing, pages 207–215, 2013.

Jakub Waszczuk. TRAVERSAL at PARSEME Shared Task 2018: Identification of Verbal Multiword Expressions using a discriminative tree-structured model. In *Joint Workshop on Linguistic Annotation, Multiword Expressions and Constructions (LAW-MWE-CxG-2018)*, 2018.

Jakub Waszczuk, Rafael Ehren, Regina Stodden, and Laura Kallmeyer. 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, August 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-5113. URL https://aclanthology.org/W19-5113.

Jason Wei and Kai Zou. EDA: Easy Data Augmentation Techniques for Boosting Performance on Text Classification Tasks, August 2019. URL http://arxiv.org/abs/1901.11196. arXiv:1901.11196 [cs].

Sarah Wiegreffe and Yuval Pinter. Attention is not not Explanation. *arXiv:1908.04626 [cs]*, September 2019. URL http://arxiv.org/abs/1908.04626. arXiv: 1908.04626.

Heike Wiese. „Ich mach dich Messer“: Grammatische Produktivität in Kiez-Sprache („Kanak Sprak“). *Linguistische Berichte*, (207):245 – 273, 2006.

Nicolas Zampieri, Manon Scholivet, Carlos Ramisch, and Benoit Favre. Veyn at PARSEME Shared Task 2018: Recurrent Neural Networks for VMWE Identification. In *Proceedings of the Joint Workshop on Linguistic Annotation, Multiword Expressions and Constructions (LAW-MWE-CxG-2018)*, pages 290–296, Santa Fe, New Mexico, USA, August 2018. Association for Computational Linguistics. URL https://aclanthology.org/W18-4933.

Aston Zhang, Zachary C. Lipton, Mu Li, and Alexander J. Smola. Dive into Deep Learning, August 2023. URL http://arxiv.org/abs/2106.11342. arXiv:2106.11342 [cs].

Rafael Ehren - Curriculum Vitae

BILDUNG

Bachelor of Arts
2007 – 2012
Kernfach: Modernes Japan
Nebenfach: Informationswissenschaft
Heinrich-Heine-Universität, Düsseldorf

Master of Arts
2012 – 2016
Hauptfach: Informationswissenschaft und Sprachtechnologie
Heinrich-Heine-Universität, Düsseldorf

Promotionsstudium
seit 2017
Allgemeine Sprachwissenschaften
Heinrich-Heine-Universität, Düsseldorf

BERUFSERFAHRUNG

Werkstudent
2011 – 2012
Kolumbus Sprachreisen GmbH
Köln

Werkstudent
2013 – 2015
trivago GmbH
Düsseldorf

Wissenschaftliche Hilfskraft
2015 – 2016
Institut für Romanische Sprachwissenschaft
Heinrich-Heine-Universität, Düsseldorf

Wissenschaftlicher Mitarbeiter
seit 2016
Institut für Allgemeiner Sprachwissenschaft (Computerlinguistik)
Heinrich-Heine-Universität, Düsseldorf

Eidesstattliche Versicherung

Ich versichere an Eides Statt, dass die Dissertation von mir selbständig und ohne unzulässige fremde Hilfe unter Beachtung der ‚Ordnung über die Grundsätze zur Sicherung guter wissenschaftlicher Praxis an der Heinrich-Heine-Universität' erstellt worden ist.

Düsseldorf, June 5th 2024

_____

Rafael Ehren