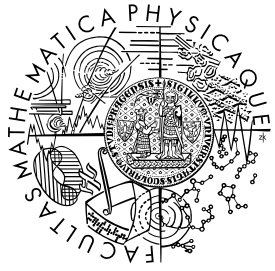


FACULTY
OF MATHEMATICS
AND PHYSICS
Charles University

DOCTORAL THESIS

Emil Svoboda

Modelling Compounds for Multilingual Data Resources

Institute of Formal and Applied Linguistics

Supervisor of the doctoral thesis: Mgr. Magda Ševčíková, Ph.D.
Study programme: Computational Linguistics

Prague 2024

I declare that I carried out this doctoral thesis on my own, and only with the cited sources, literature and other professional sources. I understand that my work relates to the rights and obligations under the Act No. 121/2000 Sb., the Copyright Act, as amended, in particular the fact that the Charles University has the right to conclude a license agreement on the use of this work as a school work pursuant to Section 60 subsection 1 of the Copyright Act.

In ..... date ....

Author's signature

To my girlfriend, fiancée, and wife Annie,
for her endless patience,
unyielding selflessness, proofreading rigor,
and existence,
for on her shoulders it was that the brunt of my ambition lay.

To my arctically excellent Gabriel,
for his canine escapades, need for walks and attention,
grounded presence, emotional intelligence,
and existence.

Title: Modelling Compounds for Multilingual Data Resources

Author: Emil Svoboda

Department: Institute of Formal and Applied Linguistics

Supervisor: Mgr. Magda Ševčíková, Ph.D., Institute of Formal and Applied Linguistics

Abstract: Compounding is a word-formation process wherein several words, roots, or stems are combined to create novel words. It has been observed in many languages, and often stands on the boundary between word formation and syntax. As such, a multilingual perspective on this process can be valuable for several fields of study, namely morphology, syntax, and typology. In this thesis, we focus on Czech, English, German, Dutch, Russian, French, and Spanish.

We first model compounds in terms of the words that they can be traced back to, calling the task compound splitting, and also in terms of identifying them from other words, calling the task compound identification. We begin by demonstrating this on Czech using deep learning and string matching. Then, on the same language, we generalize compound splitting task into parent retrieval, by building a tool called *Word Formation Analyzer for Czech*. It also covers derivation, meaning that we can trace an input word back to only a single word, and unmotivated words (recognizing that the input word has no ancestors) in addition to compounding. Finally, we present a multilingual parent retrieval and word formation classification tool called *PaReNT*, based around a custom-architecture deep model combining character-based and semantic representations, and show how the tool has been used in linguistic research.

We continue by applying this tool in combination with manual annotation to the Czech word-formation DeriNet, releasing version 2.2. We enrich this thus-far almost exclusively derivation-oriented data resource with information on compounding, and discuss the many considerations and decisions that were made along the way.

Finally, we survey the current coverage of compounds in Universal Dependencies in five languages (English, Czech, German, Dutch, Russian, Latin), and propose a way of modeling compounds by endowing them with a dependency structure and embedding them into the syntactic structure found therein.

Keywords: compounding, data sources, deep learning

Contents

Originality Disclaimer

Every single word in this dissertation was deliberately handcrafted into the final thesis, without any assistance from a large language model or similar tool, with the exception of a basic spellchecker. ChatGPT was occasionally consulted regarding technical problems and questions having to do with code, or perhaps helping to come up with appropriate linguistic examples, however no text was copied from its output or edited into this work.

That said, significant sections of this work have been taken edited with various adjustments and additions from already-published papers (whose text was also produced without the help of a large language model). Each of these papers has the writer of this dissertation, Emil Svoboda, as its first author, and his doctoral supervisor, Magda Ševčíková, as its second author. These are:

- *Splitting and Identifying Czech Compounds: A Pilot Study* (Svoboda and Ševčíková, 2021)
- *Word Formation Analyzer for Czech: Automatic Parent Retrieval and Classification of Word Formation Processes* (Svoboda and Ševčíková, 2022)
- *PaReNT: (Parent Retrieval Neural Tool): A Deep Dive into Word Formation Across Languages* (Svoboda and Ševčíková, 2024)
- *Compounds in Universal Dependencies: A Survey in Five European Languages* (Svoboda and Ševčíková, 2024)

Finally, by the very end of the thesis on page 115, one can find the List of Publications by Emil Svoboda chronologically cataloging everything the author of this thesis has published during the course of his PhD journey. The citation counts listed are taken from Google Scholar on the 30th of July, 2024, and exclude self-citations.

1. Introduction

Compounds, such as *waterfall*, are words immediately motivated by at least two words. Compounding is by extension the word-formation process by which compounds are coined. This by definition involves combining two or more words, which we dub parents (or parent words), into a single one, but the way that this is done varies. Sometimes, two words are simply concatenated, e.g. the aforementioned *waterfall* ← *water* + *fall*, but other times an interfix, such as -s-, is added as in *statesman* ← *state* + *man*) or an inflected word enters the compounding process like in *womenfolk* ← *women* (plural) + *folk*. Compounding also borders and interacts with other word-formation processes, such as derivation, conversion, blending, and others. For example, derivation and compounding can apply in a single step in *blue-eyed* ← *blue* + *eye*, and it is debatable if *undershoot* is *shoot* with a prefix or a compound of *under* and *shoot*.

To add to that, some compounds are composed of one or more so-called neoclassical constituents. These are elements typically borrowed from Latin and Greek, which do not occur as separate words, but can be used when combined with one another to form standalone words. An example of this is *biology*, where neither **bio* nor **logy* exist on their own. This means these two elements are not words, but their ability to combine together to form free words disqualifies them from being affixes. We call products of this process *neoclassical compounds*, and they fall into the scope of this thesis.

We use Czech as a starting point of our efforts, because we happen to have the most insight into the language, coupled with access to high-quality data. However, we aim for a multilingual setting, so we branch off into other languages from there. In the thesis, we have been able to cover Czech, English, German, Dutch, Russian, French, Spanish, and (partially) Latin, which represent three genera (Slavic, Romance, Germanic) of the Indo-European language family. As a result, some of the assumptions, approaches, or results may not carry over to languages from other branches or other families.

The way we model compounds is static. This means that we develop methods to take already-existing compounds and determine their compoundhood, find their parents, and additionally propose a framework under which their morphological structure can be analyzed in a way parallel to how syntactic phrases are analyzed. This is in contrast with a dynamic perspective, in which the procedure of compounding as human language ability would be simulated. To reflect this decision, the examples in this dissertation are formatted with the compound on the left, followed by a left arrow, with the parents on the right, alongside their translations and part-of-speech (POS) (cf. ex. 1, 2):

(1) ***Compound*** ← ***Parent***$_1$ + ***Parent***$_2$ (LANG)
translation.POS translation.POS translation.POS

(2) ***clairsemé*** ← ***clair*** + ***semé*** (FR)
thinly scattered.A clear.A spread.A

We call the task of automatically finding the parent words of compounds *compound splitting*, and the more general task of finding the parent or parents of any given word *parent retrieval*. The task of distinguishing compounds from non-

compounds we call *compound identification*, and the task of distinguishing compounds, derivatives and unmotivated words *word formation classification*. Coverage of this word-formation process in computational data resources has varied widely in quality and quantity. This is at least in part because there is a lack of computational tools applicable to the purpose of building and harmonizing multilingual data resources that map compound words. This is the niche we intend to fill.

Further motivation for modeling compounds in the described way is two-fold. From a linguistic perspective, multilingual compound identification can allow researchers to find compound words in lexicons and corpora. Furthermore, compound splitting can be of use in corpus linguistics dealing with inconsistent compound spelling (*flowerpot* vs. *flower pot* vs. *flower-pot*), which may hamper tokenization and by extension word frequencies. Such a model can be used to gauge which formation processes are preferred in a given language compared to another, which can find applications in linguistic typology and perhaps historical linguistics. In data sources such as word-formation networks (data sources mapping which words are created from which), compound identification can be used to help find compounds and propose links to all their parents, allowing morphologists to study which words combine with which across languages, and study how compounding behaves in conjunction with other word-formation processes such as derivation. With parent retrieval, the ability to also find derivative parents can be useful in the creation of new word-formation networks.

From the perspective of computer science, specifically natural language processing, a solution to these tasks can also be useful. Compounds are common out-of-vocabulary (OOV) elements, because especially in German and Dutch (but also in the other languages in scope) they are often spontaneously coined for an immediate purpose. This is especially the case in the domain of medicine, where spontaneous neoclassical coinages are common. Being able to dynamically map such words to their parents may help with tasks and algorithms that may be sensitive to OOV, such as topic modelling or POS tagging.

Even though the language set we are working with does not venture beyond the confines of the Indo-European language family, we nevertheless encounter a wide range of variation. As a result, the approach we take for a multilingual solution for these tasks leans heavily into deep learning. The popularity of this technique has exploded over recent years and decades, especially in computational linguistics and natural language processing, and has opened the door for the usage of this highly adaptable technique for our purposes. In order to help readers who may not be familiar with the concepts and terminology, explanations of deep learning concepts will be interspersed throughout the text whenever relevant.

In spite of this dissertation's focus is on compounding, some of the research conducted as part of it has expanded in scope into other areas. This is an expected development, because as previously mentioned, compounding borders and interfaces with other areas of language. Thus, *PaReNT*, one of the tools developed as part of this thesis, handles derivatives in addition to compounds, broadening the scope into the area of word formation in general.

We observe that compounds are not only composed of more than one word,[1] but that they also carry an implicit syntactic relation within them. As a result, we

[1] Or word, or stem, or lexeme – for a discussion on the exact definition, please refer to Chapter 2 –Compounds in the linguistic and computational context.

pave a way for a data-based analysis of compounds in a syntactic context. As a result, the last content chapter of the thesis is about a proposal to model compounds using existing dependency relations as part of Universal Dependencies, which would pave the way to study compounding not only in the context of word formation, but also in the context of syntax.

The thesis is structured as follows:

In Chapter 2 – Compounds in the linguistic and computational context, we introduce an overview of previous relevant work. This includes listing and comparing the various ways how compounds have been defined in the linguistic literature and offering our own definition tailored for this dissertation, as well as delimiting compounding from syntactic constructions on one side and derivatives on the other in Section 2.1 – Definitions of compounds. We then go over various propositions presented in the literature to classify compounds from a theoretical point of view in Section 2.2 – Classification of compounds. We then list data sources that cover compounding, focusing on the languages in scope (Section 2.3 – Compounds in language data resources), and finally go over some already-existing computational tools, algorithms, and programming languages dealing with compounds in one way or another (Section 2.4 – Compounds in procedural tools).

In Chapter 3 – Developing tools for compound analysis, we present a series of experiments in computational compound modeling. Using Czech as a starting point and spreading out into other languages, we explain what challenges and obstacles compounding has to offer, also touching upon other types of word formation, and present deep learning, the general technique used to tackle the tasks at hand (Section 3.1 – Problems and the solution). We then introduce our first experiment with *Czech Compound Splitter* (*CCS*) in Section 3.2, the to our knowledge first tool for automatically acquiring the motivating words for Czech compounds and identifying compounds from non-compounds. We continue by describing *Word Formation Analyzer for Czech* (*WFA.ces*), the successor to *CCS*, which generalizes compound splitting into parent retrieval – the ability to also find the motivating words for derivatives, and classifies words as *compounds*, *derivatives*, or *unmotivated* words (Section 3.3). The Chapter wraps up with Section 3.4 – *PaReNT* (Parent Retrieval Neural Tool), a freely-available custom-architecture tool providing the same functionality as *WFA.ces*, but for eight different languages.

In Chapter 4 – Annotating compounds in DeriNet, we show how we link compounds in the DeriNet data source to their motivating words using a combination *PaReNT*, existing raw data, and human annotation. We discussing the many decisions that had to be made along the way in Section 4.1 – Annotation scheme, and we showcase the result of our efforts, highlighting practical difficulties and statistical observations in Section 4.2 – DeriNet 2.2.

In the final Chapter 5 – Incorporating compounds into Universal Dependencies, we pave the way for the inclusion of compound annotation in a syntactic context, namely Universal Derivations (UD). We survey how compounds are currently handled in UD in Section 5.1 – Current annotation, highlighting the various intra- and inter-linguistics differences and inconsistencies in the annotations. Then in Section 5.2 – Syntax-based annotation of compounds, we propose how to annotate compound words in an orthographically and cross-lingually consistent way while demonstrating how this information can be useful to the in the field of typology. We summarize the dissertation in Chapter 6 – Conclusion.

2. Compounds in the linguistic and computational context

This chapter is dedicated to delimiting our field of interest and presenting previous work performed by others. In Section 2.1 we map attempts to define compounding, explain how they pertain to this thesis, and go over how compounds are delimited from or related to other linguistic objects such as blends, syntactic phrases, derivatives, and neoclassical compounds. A working definition of compounding usable for the purposes of this dissertation is proposed. In the following section 2.2 attempts to classify compounds, covering both language-specific and multi-lingual taxonomies from a wide range of authors, are mapped. In Section 2.3, we go over a selection of data sources that contain compound words that are in one way or another relevant to this thesis, and in the final Section 2.4 algorithms, tools, and models that can analyze or handle compounds in one way or another are covered.

2.1 Definitions of compounds

A widely accepted definition of compounding has not been put forth – unsurprisingly so, since many cross-lingual definitions of related or antithetical concepts (e.g. word, root, phrase, stem, sentence, multi-word expression...) would be necessary for such a delimitation to be put together (Scalise and Vogel, 2010). Indeed, Haspelmath (2002) argues that multilingually, the fuzzy space ranging between affixes through words right up to syntactic phrases is an unclustered continuum. If we were to accept this view, we would be forced to concede that proposing any definition of compoundhood is simply an act of drawing imaginary lines somewhere into this fuzzy space.

Regardless of whether this particular view is true or not, Scalise and Vogel (2010, 5) non-exhaustively list no less than 8 definitions that have been proposed in the literature to this date, which when sorted chronologically include:

a. *When two or more **words** are combined into a morphological unit, we speak of a compound* (Marchand, 1960);

b. *(...) a compound word contains at least two bases, which are both **words**, or at any rate, **root morphemes*** (Katamba, 1993);

c. *Composition (...) denotes the combining of two **free forms** or **stems*** (Olsen, 2000);

d. *A lexical unit made up of two or more **elements**, each of which can function as a **lexeme** independent of the other(s) in other contexts, and which shows some phonological and/or grammatical isolation from normal syntactic usage* (Bauer, 2001);

e. *A complex lexeme that can be thought of as consisting of two or more **lexemes*** (Haspelmath, 2002);

f. *(...) root compounds consist of two **stems** combined as one, with the compounds as a whole bearing the category and morphosyntactic features of the right-hand stem* (Lieber, 2004);

g. *Its defining property is that it consists of the combination of **lexemes*** (Booij, 2005);

h. *A word-sized unit containing two or more **roots*** (Harley, 2011);

In summary, most definitions posit something along the lines that a compound is a word or word-like linguistic object that contains two or more **words**, **stems**, **lexemes** or **roots**, depending on the author. Notice that none of the listed definitions mentions orthography as a defining or even relevant characteristic – in other words, we can conclude that it is generally accepted that whether a given object is spelled as *flower pot* (henceforth: open compound), *flower-pot* (henceforth: hyphenated compound) or *flowerpot* (henceforth: closed compound) has no bearing on its compoundhood. This observation has important implications in the realm of computational linguistics, because by convention, many compoutational tools depend on orthographic tokenization. Coverage of compounds in certain data resources may therefore be inconsistent, because open and closed compounds are handled differently – specifically, open compounds are tokenized into two or more words, whereas closed compounds are considered to be a single word. This is the case with e.g. Universal Dependencies[1] at present.

Scalise and Vogel remark that in most of these definitions, compounds are a special type of the units that they are composed of [2] – in other words, if a compound is composed of lexemes or words, it is itself a lexeme or words. What follows from this is that compounds can often later undergo compounding, themselves, resulting in so-called *recursive compounds* (cf. ex. (3), (4)). These are distinct from compounds that are flatly traced back to multiple parents, as such non-recursive compounds have no existing in-between step (cf. ex. (5)), the name of the parody epic *Batrachomyomachia*), while no *βατραχομυός or *μυομαχία exists).

(3) **самолётостроение** ← **самолёт** + **строение** (RU)
airplane-building.N airplane.N building.N

(4) **самолёт** ← **сам** + **лететь** (RU)
airplane.N alone.P fly.V

(5) **Βατραχομυομαχία** ← **βάτραχος** + **μῦς** + **μάχη** (GR)
battle-of-frogs-and-mice.N frog.N mouse.N battle.N

It **is not** the goal to decide which one of these listed definitions is the most 'correct' one, or even attempt to introduce its own. However, the goal **is** to computationally model compound words, and therefore *some* idea as to what compounds are is necessary.

Working with Haspelmath's of the aforementioned definitions, we postulate that a compound is, in a way obvious to an average speaker,

a) a lexeme

[1] https://universaldependencies.org/

[2] Unlike e.g. morphemes, which are composed of phonemes while no being a special type of phoneme, or derived words, which are composed of roots/stems and affixes while not being a special type of either one.

b) that can be traced back to at least 2 pre-existing lexemes or neoclassical constituents[3] (henceforth: parent words, parents, or ancestors)

c) with no reconstruction of root material, with the exception of allomorphy.

As pointed out by (Lieber and Štekauer, 2009, 5), definitions relying on the concept of the lexeme run into the problem of making clear what exactly a lexeme is, especially if the definition is to be applicable cross-lingually. Within the context of this dissertation, we focus on working with existing data sources, and therefore rely on the discretion of the authors of the databases we are operating with to handle this problem. Which linguistic object is or is not a lexeme therefore falls out of our scope.

2.1.1 Compounds versus blends

Condition c) presented in the postulated definition is important, because it delimits compounding from blending. When analyzing the English *smog*, which is a blend of *smoke* and *fog*, we notice that it is necessary to reconstruct the /f/ from *fog* in order to trace the word back (as well as the /ouk/ from *smoke*). In this case, both roots are incomplete.

This is in contrast with the Czech *krvotok* 'bloodflow' from *krev* 'blood' and *tok* 'flow'. Even though it is similarly necessary to reconstruct the /e/ that is lost in the compounding process, the same loss occurs in the inflection of *krev* (e.g. the genitive *krve* 'of blood'), and the change is therefore part of the regular allomorphy of the root in question. The root is therefore complete despite missing a phoneme (and the loss is transparent to a native speaker), and as a result satisfies condition c).

This view is in accordance with Adams (1977, 149), who understands blends as words containing so-called *splinters*, which are **incomplete** morpheme fragments – and by condition c), all roots must be complete for something to be a compound.

2.1.2 Compounds versus syntactic phrases

Another aspect of compoundhood that must be addressed is its boundary with syntax.

We understand syntactic phrases as linguistic units which are, similarly to compounds, composed of individual words, but unlike in compounds, the exact relationship among these constituents is explicitly signaled by the usage of function words or inflectional markers such as case endings. Multi-word expressions, such as the English *kick the bucket* or the Spanish *estar en las nubes* (lit. 'to be in the clouds') are a specific type of syntactic phrase whose meaning is non-compositional, meaning that unless one has memorized these objects, there is no way to know they mean 'to die' and 'to daydream' respectively. This makes them *lexicalized*, in the sense that they must be stored in memory in a way that is at least similar to the way words are memorized. Generally, multi-word expressions (MWEs) are

[3] Semantically independent linguistic objects that cannot appear as standalone words but can combine with one another to form standalone words, typically inherited from Latin or Greek; for details please refer to Section 2.1.4.

nevertheless still considered to be syntactic constructs rather than morphological words, because they exhibit internal flexion (e.g. *he kicks the bucket*), and therefore fall out of the scope of this dissertation. Being able to distinguish them from compounds is therefore important.

The problem is that in a multilingual setting, the distinction between morphology and syntax seems to be a continuum. Haspelmath (2002) even claims that "as of now, we do not currently have a good basis for dividing the domain of morphosyntax into morphology and syntax". Nevertheless, (Schlücker, 2019a) addresses this issue in the context of 11 European languages, first in general, and then language by language.

In the general case, Finkbeiner and Schlücker (2019, 1-43) use a notion proposed by Gaeta et al. (2009) that whether or not a given linguistic object is lexicalized and whether or not it is produced by morphological operations are two values that are completely independent of each other. Therefore, when evaluating whether or not a given linguistic object, one that satisfies condition b) in our postulated definition, is a compound, there are four possible combinations of these two variables:

a) $[+\text{morphological}], [+\text{lexical}]$

b) $[+\text{morphological}], [-\text{lexical}]$

c) $[-\text{morphological}], [+\text{lexical}]$

d) $[-\text{morphological}], [-\text{lexical}]$

In this model, $[+M, +L]$ is a typical lexicalized compound such as *Liebesbrief* 'love letter' (as signalled by the interfix -s-) or *milkshake*, $[+M, -L]$ would be a nonce compound like *bike girl* or *bananaphobic*, $[-M, +L]$ is a MWE like *spill the beans*, and $[-M, -L]$ is a simple syntactic construction whose meaning is determined by the composition of its parts like *own a house*. Delineating the boundary between compounds and multi-word expressions therefore boils down to determining the $[+|-M]$ property, the difficulty of which in turn is depends greatly on the language in question.

In Czech (and by extension Russian), the class of what Bozděchová (1997) calls *proper compounds* (compounds created by spontaneous coinage; more details on her classification in Section 2.2) in most cases contains morphological markers such as the addition of an interfix, most commonly -o- as in the Czech *sil-o-čára* 'line of force' $\leftarrow$ *síla* 'force' $+$ *čára* 'line' or the Russian *земл-е-трясение* 'earthquake' $\leftarrow$ *земля* 'earth' $+$ *трясение* 'shaking'. Another morphological marker of compoundhood is the absence of an inflectional suffix, which allows us to distinguish the compound *vlakvedoucí* 'train conductor' from the equivalent syntactic phrases *vedoucí vlak**u*** or the also grammatically correct but strange-sounding due to its archaic word order *vlak**u** vedoucí*. A similar situation can be found in Russian, where термометр-максимум 'thermometer maximum' is distinguished by the lack of inflectional ending from its associated phrase максимум термометра 'maximum of a thermometer'. The boundary between what Bozděchová calls *improper compounds*, formed by the 'freezing' of a syntactic phrase, and synchronic syntactic phrases, is however much blurrier. In fact, this is the case with the aforementioned *vlak**u** vedoucí* – it could be argued that this in fact **is** an improper compound. The compoundhood status of *vlakvedoucí* is however unambiguous. In practice, the

Czech linguistic tradition does not consider open compounds to be compounds as such.

Some markers of improper compounding do exist; word order is often reversed within compounds compared to their associated syntactic phrases, as can analogously be seen in the Czech *penězchtivý* 'money-wanting' vs. *chtivý peněz* 'wanting money'; the stress pattern of a compound may differ from its originating phrase (*'chválabohu* 'fortunately' behaves as a particle vs. *chvála 'Bohu* 'praise to God'), or the collocability of a compound may differ from its associated phrase (*вечнозелёный* 'evergreen' vs. *вечно зелёный* 'always green'; the former occurs almost exclusively in reference to non-deciduous trees, while the second refers to anything that remains green). Ultimately, the only problematic cases are the so-called improper compounds where the ordering of the constituents is not switched, and specifically in Czech improper compounds may in practice be distinguishable from MWEs or syntactic phrases only by orthography.

The situation is rather similar in German and Dutch, in that distinguishing compounds from MWEs is mostly reliable. Compounds are very productive in Dutch and German, and are generally morphologically obviously distinct from MWEs and syntactic phrases, as evidenced by the common presence of parallel structures such *Frischluft* and *frisches Luft* 'fresh air', or *opoefiets* 'grandma-bike' = 'retro bike' and *opoe's fiets* 'grandma's bike'.

Schlücker (2019b, 69-94) and Booij (2019, 95-126) identify the following properties of German and Dutch (respectively) compounds that do so:

(i) stress;

(ii) absence of inflection;

(iii) inseparability;

(iv) presence of linking elements, though not present in all subtypes;

(v) spelling, as compounds are consistently spelled together or hyphenated in both languages.

Some exceptions do occur, e.g. Dutch $[A + A]$ compounds are not necessarily easy to distinguish from syntactic phrases, because Dutch adjectives can be used as adverbs without being morphologically marked. This makes it unclear if a given object is an adjective syntactically modified by an adverb, or a compound of two adjective formed by concatenation.

In the Romance languages, orthography is much less helpful than in the Slavic languages or German and Dutch. Radimský (2015) on the example of Italian however posits that Romance $[N + N]_N$ (and perhaps by extension all) compounds, can generally be distinguished from outputs of syntactic operations by the fact that the relation between the constituents is implicit (e.g. *caffè latte*; lit. 'milk-coffee') as opposed to explicitly determined by the usage of a preposition, conjunction, or inflectional ending (e.g. *caffè e latte*; lit. 'coffee with milk'), which is compatible with the model proposed by Gaeta et al. (2009). The problem in the Romance languages is therefore distinguishing appositional $[N + N]$ structures equivalent to the French *(Le) président Macron*, like the Italian *(mia) sorella Maria* '(my) sister Maria', from compounds. The distinction can according to Radimský be made based on two

things. The first is the observation that such appositional constructions (with the exception of the numeral type) tend to appear in two roughly equivalent flavors, dubbed *plain structure* and *parenthetical structure* (cf. left vs. right in ex. (6) and (7)). The ability to undergo a transformation into the parenthetical reveals that these constructions are in fact syntactic in nature, and therefore not compounds.

(6) **mia sorella Maria** $\simeq$ **Maria, mia sorella** (IT)
my sister Maria Maria, my sister

(7) **le président Macron** $\simeq$ **Macron, le président** (FR)
president Macron Macron, the president

In English we also run into the problem of non-existent singular nominal endings. This makes it difficult to morphologically distinguish nouns from their respective noun-derived adjectives. Bauer (2019, 45-66) argues that as a result of this lack of morphological distinction, the boundary between compounds and MWEs in English simply is fuzzy. Bauer nevertheless goes over several potential criteria that could hypothetically decisively set this boundary, and shows that they all fail at least some of the time.

i **Stress** can be used to distinguish e.g. *black bird*, a $[-M, -L]$ syntactic phrase with stress on the second syllable from *'blackbird*, a $[+M, +L]$ compound, alongside the inability to modify the first constiuent in the former (**a very blackbird* nor **blackestbird*) and observing that *a brown blackbird* is not a contradiction. Nevertheless, Bauer finds a series of counterexamples such as *'apple cake* vs. *apple 'pie* and points out that real speakers are often inconsistent in stress assignment.

ii **Spelling** reflects stress quite often, and can be used to distinguish *railway* from *iron bar*; but it also fails in many cases, not least because it is like stress also inconsistent (*rainforest*, *rain-forest* and *rain forest* refer to the same concept) and also because the formally and semantically parallel *schoolgirl* and *university student* are spelled separately.

iii The final criterion Bauer pays special attention to is if the object in question **blocks internal inflection**; that is, given a compound verb such as *badge-flash*, we observe that its plural usage leaves the first constituent uninflected, cf. *we badge-flashed our way into the scene* vs. **we badges-flashed our way into the scene*. The sole usage of this criterion would however lead to a situation where *suggestion box* would be a compound, but not *suggestions box*.

In summary, in English the boundary between compounding and syntactic objects seems to be fuzzy, but in many situations there may be and indicator to go by. It is worth pointing out, however, that at least in English, orthography can serve as a sort of lower bound for compoundhood. That is, an object spelled with a space or hyphen may be a compound, a MWE, or a syntactic phrase, but it would be very hard to find an example of a MWE or syntactic phrase spelled together (barring typos).

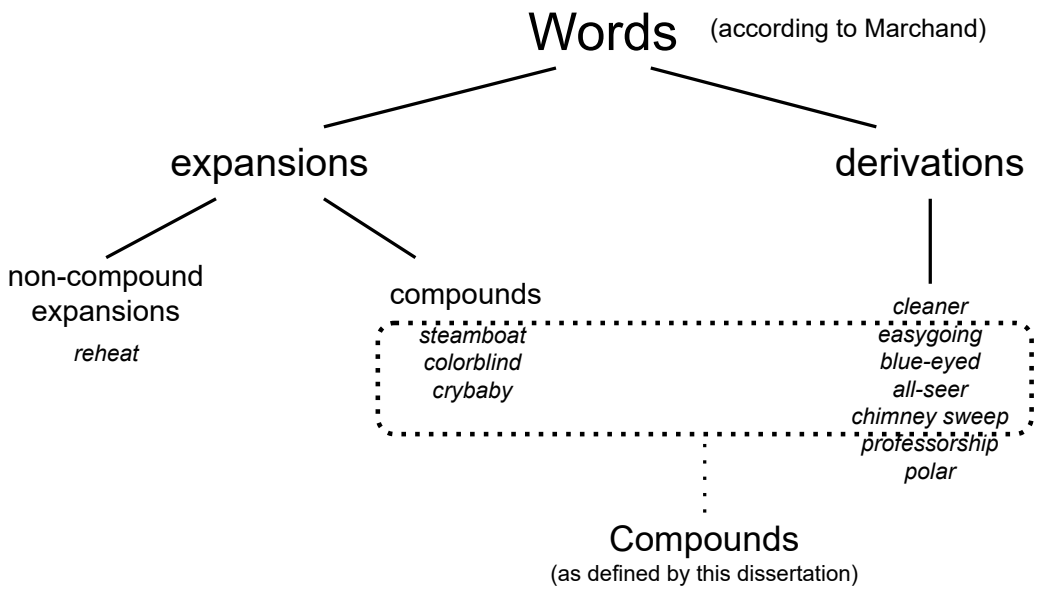


Figure 2.1: Comparison of what is considered a compound by Marchand (1967) and by this dissertation (in the dotted rectangle).

2.1.3 Compounds versus derivatives

In comparison to compounds, derivatives are generally understood to be words that are coined by the addition of affixes to existing words, and can therefore be traced back only to a single word. Unlike roots or stems, affixes must be attached to a root or stem in order to appear, and cannot be combined with one another to produce free forms.

It should be first noted that not all scholars are necessarily of the opinion that the two concepts stand in opposition. Specifically, Marchand (1967) is of the opinion that word formation can be viewed only as comprised of two types processes – *expansion* and *derivation*. Under this consideration, *expansion* is understood as the addition of a modifying element (*determinant*) to a free-standing, pre-existing element, which then dominates the grammatical and lexical properties of the resulting word (*determinatum*). This encompasses both *steamboat* and *reheat*, since *steamboat* behaves grammatically and lexically much like *boat*, and *reheat* behaves much like *heat*.

Marchand specifically states that a word is a compound if and only if it is the product of expansion whose determinant happens to occur on its own. In contrast to expansion, derivation is in Marchand's view a situation where it is a bound morpheme that dominates the lexical and grammatical properties of a given word. Examples include *clean* $\rightarrow$ *cleaner*, where the part-of-speech is determined by the suffix, and *professor* $\rightarrow$ *professorship*, where the suffix changes the lexical class of the input word from PERSONAL SUBSTANTIVE to ABSTRACT CONDITION-DENOTING SUBSTANTIVE. Therefore, words like *easygoing*, *highborn*, *heartbreaking*, *chimney sweep* are considered to be derivations by Marchand, but are compounds under the definition postulated in this dissertation. The difference between these two considerations can be observed in Figure 2.1.

While most scholars operate with the notion of derivation being the addition

of affixes – bound lexical morphemes – instead of Marchand's definition cited above, compounds of the type *blue-eyed* and *chimney sweep* need to be addressed nevertheless. The key observation here is that there is no **eyed*, so *blue-eyed* must be understood as derivation and compounding happening at the same time. Analogously, for the nominal *chimney sweep* there is no corresponding nominal **sweep*, only the verb *to sweep*, forcing us to consider the word formation process as compounding together with conversion. In accordance with Bisetto and Melloni (2008), we call such words *parasynthetic compounds*. This process has been cross-linguistically attested, and it typically but not exclusively involves having something of a certain quality or quantity, or being the agent of a verb with the object attached , cf. ex. (8, 9, 10, 11, 12).

(8) ***quinzeañera*** ← ***quinze*** ***año***, but no **añera* (ES)
fifteen-year-old girl.N fifteen.NUM year.A

(9) ***кровосос*** ← ***кровь*** ***сосать***, but no **сос* (RU)
bloodsucker.N blood.N suck.N

(10) ***dřevorubec*** ← ***dřevo*** ***rubat***, but no **rubec* (CS)
woodcutter.N wood.N cut.V

(11) ***bruinogig*** ← ***bruin*** ***oog***, but no **ogig* (NL)
brown-eyed.N brown.A eye.N

(12) ***langbeinig*** ← ***lang*** ***Bein***, but no **beinig* (DE)
long-legged.N long.A leg.N

While we do consider parasynthetic compounds to be compounds, we find it useful to separate them from *secondary compounds*.

(13) ***sleepwalker*** ← ***sleepwalk*** ← ***sleep walk*** (EN)

(14) ***Fachlehrerin*** ← ***Fachlehrer*** ← ***Fach*** ***Lehrer*** (DE)
subject teacher (fem.).N subject teacher.N subject.N teacher.N

(15) ***garde-robier*** ← ***garde-robe*** ← ***garder*** ***robe*** (FR)
wardrobe-carer.N wardrobe.N retain.V clothes.N

The last point to address is determining whether a given linguistic object is an affix or a freely-occurring word. This question unfortunately in a lot of cases does not have a good answer, and therefore the boundary between derivation and compounding simply remains blurred. A typical example of this blurry boundary are compounds containing elements that can function as prepositions. On the one hand, no one would argue that *under* is a freely-occurring word in English and can combine with other prepositions such (*be*)*neath* to form *underneath*, but on the other hand it is a function word), and very productive in a way reminiscent of an affix. As a rule of thumb, we consider non-lexical objects such as prepositions to be freely-occurring words if they have two syllables or more (e.g. Dutch *onder* 'under', Czech *mimo* 'outside of'), otherwise, we consider them to be affixes if attached to another word (e.g. Czech *pod* 'under').

2.1.4 Neoclassical compounds

However, the inverse also exists, in that there are non freely-occurring linguistic objects are semantically self-contained . These most often come from Greek, Latin, or from Greek throug Latin. Words may be formed out of these in combination with freely-occurring words (*teleprompter* ← *-tel-* 'distance' + *prompt*, *hydroelectric*) *-hydr-* 'water' + *electric* or with one another (*logography* ← *-log-* 'speech' + *-graph-* 'write'; *geomorphology* ← *-ge-* 'Earth' + *-morph-* 'shape' + *-log-* 'word; speech'). The fact that one can create a freely-occurring word just by combining neoclassical constituents forms the primary argument for granting the status of compounds to these words, since it is generally accepted that the combining of bare affixes together does not produce viable words. As a result, we consider neoclassical compounding to be a special case of compounding and therefore generally in scope of this dissertation.

Neoclassical compounding occurs in all of the languages in scope, and another peculiarity of it is the fact that they tend to be shared among languages, in the sense that if e.g. English has a neoclassical compound like *telephone*, it is highly probable that a similar word with the same meaning will be found in the other languages in scope, cf. German *Telefon*, Dutch *Telefoon*, Russian телефон, French *téléphone*, Spanish *teléfono*. In fact, neoclassical compounds are often written about in the context of *internationalisms* (words shared among a wide plethora of languages) rather than in the context of compounding (Wexler, 1969; Pulcini, 2019; Melloni, 2023).

The view that neoclassical compounds are a specific type of compounding is not necessarily shared by other scholars, though. Bauer (1998) argues that at least in English, there is no discrete class of neoclassical compounds, and that instead lexical enrichment should be thought of as a continuous space of three dimensions – *simplex-compound*, *native-foreign*, *abbreviated-nonabbreviated*, and that neoclassical compounding is "a label given to one small section within this three dimensional space, but actual words diverge from the prototype considerably".

Furthermore ten Hacken (2011) explicitly considers neoclassical compounding to be a subsystem of English and other European languages. While ten Hacken denies that neoclassical compounding is productive in the sense of syntax, i.e. being able to unintentionally produce a potentially infinite number of expressions immediately understandable by a native speaker without being stored in the lexicon, the author nevertheless concurs that neoclassical compounding is available to the European speaker for the naming of novel concepts. This means that as time goes on, it can be assumed that more neoclassical compounds will appear in the languages in scope.

Panocová and ten Hacken (2020, 32) follow with the observation that while some neoclassical compounds seem to be borrowed directly from Greek to Latin, novel coinages do occur synchronically. When a new neoclassical compound is coined, it is typically swiftly adopted with minimal morphological changes into other European languages by the speakers of the specialized (such as medical or scientific) communities that typically use these formations. Typically, the source language is English, although in many cases the direction of borrowing is impossible to confirm. In spite of that, the authors give the example *laparoscopy* as a neoclassical coinage originating in German. The authors also compare the neoclassical lexicons of English and Russian, and reach a conclusion that while English does have a

productive discrete mechanism of neoclassical word coinage, Russian mostly does not, and that Russian neoclassical compounds are mostly direct borrowings.

Ološtiak and Vojteková (2021) focus on neoclassical compounding in the West Slavic languages. Four types of word-formation formants are distinguished, namely *bases*, *baseoids*, *affixoids*, and *affixes*. Bases are items that can appear freely and carry lexical meaning (*terapie* 'therapy', like in *ergoterapie* 'occupational therapy'); Baseoids are items that do not appear freely, but carry lexical meaning regardless (*ergo-*, in *ergoterapie* 'occupational therapy'), and Affixoids are items that are diachronically lexical, but have gradually lost their ability to appear independently and have generalized their meaning enough to effectively behave like derivational items; and affixes are elements that behave like bound morphemes. The distinction between these three types of formant seem to be congruent with the fuzzy subspace proposal by Bauer (1998) described earlier, as the distinctions seems to reflect the proposed *simplex-compound* and *abbreviated-nonabbreviated* axes.

Three types of compounds are by Ološtiak and Vojteková delimited according to the type of formants they involve. *Proper compounds*[4] are characterized as being composed of two bases (ex. 16). *Semi-compounds* are composed of one base and one baseoid (ex. 17). Finally, *quasi-compounds* are composed of two baseoids (ex. 18).

(16) ***sér|-o-|pozitivní*** ← ***sérum*** ***pozitivní*** (CS)
seropositive.A serum.N positive.A

(17) ***krypto|politika*** ← ***krypto-*** ***politika*** (CS)
cryptopolitics.N crypto-.BASEOID politics.N

(18) ***eko|logie*** ← ***eko-*** ***-logie*** (CS)
ecology.N eco-.BASEOID -logy.BASEOID

Our conceptualization of neoclassical compounds is mostly congruent with Ološtiak and Vojteková, with a reduction in granularity. Everything the authors consider to be a *baseoid* and some of what the authors consider to be an *affixoid* is considered to be a *neoclassical constituent* (labelled 'neocon' in examples) by us. We also systematically interpret neoclassical constituents as identical whenever their etymology and semantics allow for it, even under circumstances where they undergo formal changes. For instance, the first element of *logografie* 'logography' (*logo-*) and the second element of *sociologie* 'sociology' (*-logie*) are seen to be the same, since they both descend from the same Greek root. In our data, they are represented by the string *-log-*, cf. Section 3.2.1 for more details.

As a final note, it is worth mentioning that Melloni (2023) posit that while compounding is the most typical word formation process for neoclassical constituents, it is not restricted to this one process. It is pointed out that clipping, blending, and even derivation of neoclassical constituents can be attested in English, and therefore operates with the broader concept of neoclassical word formation.

[4]The usage of this term by these authors is distinct from Bozděchová's proposal above.

2.2 Classification of compounds

In the linguistic literature, the debate over compounding and its taxonomy has been centered around a number of topics. In the following subsection, we present a list of parameters which have been used to classify compounds into taxonomies, and then we go over a short history of compound classifications based around these parameters.

2.2.1 Classification parameters

- **Part-of-speech (POS)** category of the compound and its components: if the components obtained by splitting the compound do not correspond to independently existing words, the POS of the component is determined according to the closest word. If this applies to the head, the compound's POS is different from its head's POS (cf. the distinctions below; for examples, see Section 2.2). A common convention for expressing this is

$$[\mathbf{POS_1} + \mathbf{POS_2} + \ldots]_{\mathbf{POS_o}},$$

  where $\mathbf{POS_n}$ refers to the part-of-speech of the nth parent and $\mathbf{POS_o}$ refers to the part-of-speech of the output, i.e. the compound itself.

- **Headedness**: if one of the components plays a prominent role, it is considered the head; left-headed compounds and right-headed compounds are distinguished. Most scholars (e.g. Fabb 1998; Haspelmath and Sims 2013; Scalise and Bisetto 2009; Bozděchová 1997; Štichauer 2013) operate with some notion of a *head*; that is, some compounds have a constituent that in some way governs the properties of the given compound. For example in 19, the resulting noun is masculine, inheriting its gender from its right constituent, and refers to a particular city.

(19) **Волгоград** ← **Волга** + **(град)** (RU)
Volgograd.N Volga.N (city.N)

  Those compounds that do have a head are often analyzed as to whether the head is followed by the non-head element, or vice versa. Compounds in which the head precedes the modifier are termed *left-headed*; compounds in which the head follows are termed *right-headed*; this suggests that ex. 19 is right-headed – otherwise, the word would inherit the feminine gender of *Volga*, and would probably denote some part of the river. Fabb (1998) additionally considers two-headed compounds. Some authors distinguish between *syntactic* heads and *semantic* heads, the former of which governs the compound's formal properties, like gender; and the latter of which governs the compound's meaning.

- **Endocentricity vs. exocentricity**: Centricity evaluates whether a given compound inherits form or meaning from one of its constituents. The head typically determines the POS and meaning in endocentric compounds; an exocentric compound is headless or, as Bauer (2001, p. 70) puts it, it is "a

compound which is not a hyponym of its own head element". constituents by means of subsetting (endocentric; an *apple cake* is a type of cake) or not (exocentric; a *cutthroat* is not a type of throat). This is usually understood as being related to headedness; in fact, Fabb (1998) considers exocentricity to be synonymous with headlessness.

- **Relations** between the compound's constituents: in the literature cited below, the compound's internal structure is indicated by brackets, in analogy to syntactic constituent trees. Most scholars who use the relation concept list one or more relation that is symmetric (often calling such a relation some variant of *coordinate* or *coordinative*) – Fabb (1998), for example, considers *coordinative* compounds to be synonymous with two-headed compounds, as the heads modify each other – and one or more asymmetric relations, such as the subordinative and attributive relations of Scalise and Bisetto (2009).

- **Syntactic type** of the relation between the compound parts: the crucial distinction is whether the components are independent of each other (coordinate, coordinative, additive or copulative are some of the terms used) or whether one depends on the other (subordinate, determinative, etc.).

2.2.2 Compound taxonomies

These features, assigned varying degrees of importance, interrelation, and priority, have been employed to classify compounds by various scholars.

Sanskrit compounds

The first such attempt is proposed by Pāṇini, an ancient logician and grammarian, active sometime between the 7th and 4th centuries B.C. in his grammar of Sanskrit titled Aṣṭādhyāyī (Pāṇini, 1987), who focused on nominal compounds. We rely on the help of the 7th edition of Kale (1931) to help produce this short overview.

Under Pāṇini's consideration, there are six types of compounds in Sanskrit.

1. **Dvandva** are compounds whose constituents are in what we may in the language of contemporary theory call a coordinate or copulative relationship – that is, their constituents are implicitly or explicitly related by a symmetric operator AND or OR, as in *actor-director*.

2. **Tatpuruṣa** (translating to 'his man') are compounds with a nominal head element modified by a noun inflected in one of the oblique cases, and which furthermore denote a special case of the head, much like 'his man' refers to an actual man.

3. **Upapada** refers to a variant of tatpuruṣa compounds where the second element is an otherwise non-existent element, typically a derivative or conversion a verb, such as *pottery-maker* (strictly speaking only if *maker* were non-existent in English on its own).

4. **Karmadháraya** ('what holds together') compounds are similar to tatpuruṣa compounds, except that the nominal head is modified by a uninflected noun, an adjective, or other part-of-speech, such as *watermill*. Like the aforementioned *tatpurusa* compounds, they are endocentric.

5. **Bahuvrihi** ('possessing a lot of rice' = 'rich'), are also compounds in which a nominal head is modified by something, but the whole compound refers to something else than the head element or has a different part of speech (e.g. like *blue-eyed*, *lowlife*).

6. Finally **avyayibháva** refers to indeclinable compounds whose head is a pronoun or adverb.

While this classification seems to be used only seldom in contemporary literature of compounding outside of Sanskrit and the languages of the Indian subcontinent, so much of its terminology and observations have propagated into current theory that it is useful to go over the classification nevertheless.

Multilingual approaches

Bisetto and Scalise (2005) compare the following classifications, sorted chronologically.

- Bloomfield (1933) classifies constructions into either *determinative* and and *copulative* (*bittersweet*, *loudmouth*). *Determinative* compounds are then further classified into *subordinative* (*love story*) and *attributive* (*blackbird*) compounds. In addition, any type can be *endocentric* or *exocentric*.

- Bally (1944) presents an unbranching ternary classification of French compounds into *de coordination* (*sourd-muet* 'deaf-mute'), *d'accord* (*chaleur solaire* 'solar heat'), and *de rection* (*maison de campagne* 'villa').

- Marchand's 1960 classification of endocentric compounds starts with splitting them into *verbal nexus* (=dependent on a governing verb whose argument is the modifier) and *non verbal-nexus*. *Verbal nexus* compounds are then categorized as *non-synthetic* (*crybaby*) and *synthetic* (*babysitter*). Non verbal nexus compounds are in turn comprised of *rectional* and *copula* compounds. The latter are then ternary-branched into subsumptive (*oaktree*), attributive (*girlfriend*), and additive (*fighter-bomber*) compounds.

- Spencer (1991) offers an unbranching quaternary taxonomy of *endocentric* (*head-modifier*; *student film society*), exocentric (*predicate-argument*; *pickpocket*), dvandva (roughly equivalent to copulative; *Austria-Hungary*) and appositional (*learner-driver*) compounds.

- Fabb (1998) classifies two-constituent compounds simply into *headless* (*exocentric*; *bluestocking*), *single-headed* (*endocentric*; blackbird), and *two-headed* (*copulative*; *writer-director*).

- Olsen's 2001 classification breaks off into three branches on the first level – *determinative* (*coffee cup*), *copulative*, and *possessive* (*greybeard*). The only subdivision is of *copulative* compounds, which are divided into *dvandva* (*Simha vyahghra* 'lion and leopard') and *pseudo-dvandva* (*Löwenleopard* 'lion-leopard crossbreed').

- Haspelmath (2002, 137-144) proposes a 5-way unbranching classification into *endocentric* (*lipstick*), *exocentric* (*lavapiatta* 'dishwasher'), *affixed* (*green-eyed*), *coordinative* (*elun-ai* 'adult and child' (Korean)), and *appositional* (*poet-painter*).

- Bauer (2001) splits compounds four ways into *determinative* (*karmadhāraya*), *dvandva* (*Schleswig-Holstein*), *bahuvríhi* (*greybeard*), and *synthetic*, further classifying *determinative* compounds into $[A + N]$ (*blackbird*) and $[N + N]$ (*woman doctor*) subtypes.

- Finally, Booij (2005) proposes an unbranching five-way taxonomy, splitting compounds into *endocentric* (*travel office*), *exocentric* (*lavapiatti* 'dishwasher'), *bahuvríhi* (Kahlkopf lit. 'bare-head'='bald person'), *copulative* (*candra-ditya-u* Sanskrit 'sun-and-moon'), and *appositive* ('prince-bishop').

Based on these classifications, Bisetto and Scalise (2005) propose another classification. The authors speak of grammatical relations:

> "The grammatical relations holding between the two constituents of a compound are basically the relations that hold in syntactic constructions: subordination, coordination and attribution."

The relation between the components – *Coordinate*, *Attributive*, and *Subordinate* – is used as the first-level criterion in Bisetto and Scalise's classification:

1. **Coordinate** compounds exhibit a relation that is symmetric; that is, neither constituent syntactically or semantically dominates over the other. In contrast, the relation is asymmetric in both subordinate and attributive compounds.

2. In **subordinate** compounds, the modifier functions as a syntacto-semantic complement to the head, typically (but not exclusively) an *of* relation in the case of nominal compounds (*taxi driver* = *driver of a taxi*).

3. In **attributive** compounds, the modifier of the compound is either a property expressing adjective or a noun. In the case of nouns, attributive compounds differ from subordinate compounds is that a singular specific property of the modifier is used to describe the head, and otherwise for the specific property the modifier has nothing to do with the head. As an example, *mushroom soup* contains the literal spore-bearing fruiting body of a fungus and is therefore subordinate, but a *mushroom cloud* has nothing to do with mushrooms apart from one single property of a mushroom, which is its shape.

The second level is the distinction between *endocentric* and *exocentric* compounds. A *windmill* is a type of mill, but a *pickpocket* is not a type of pocket (nor a type of *to pick*, as *pickpocket* is not a verb); similarly, the English endocentric

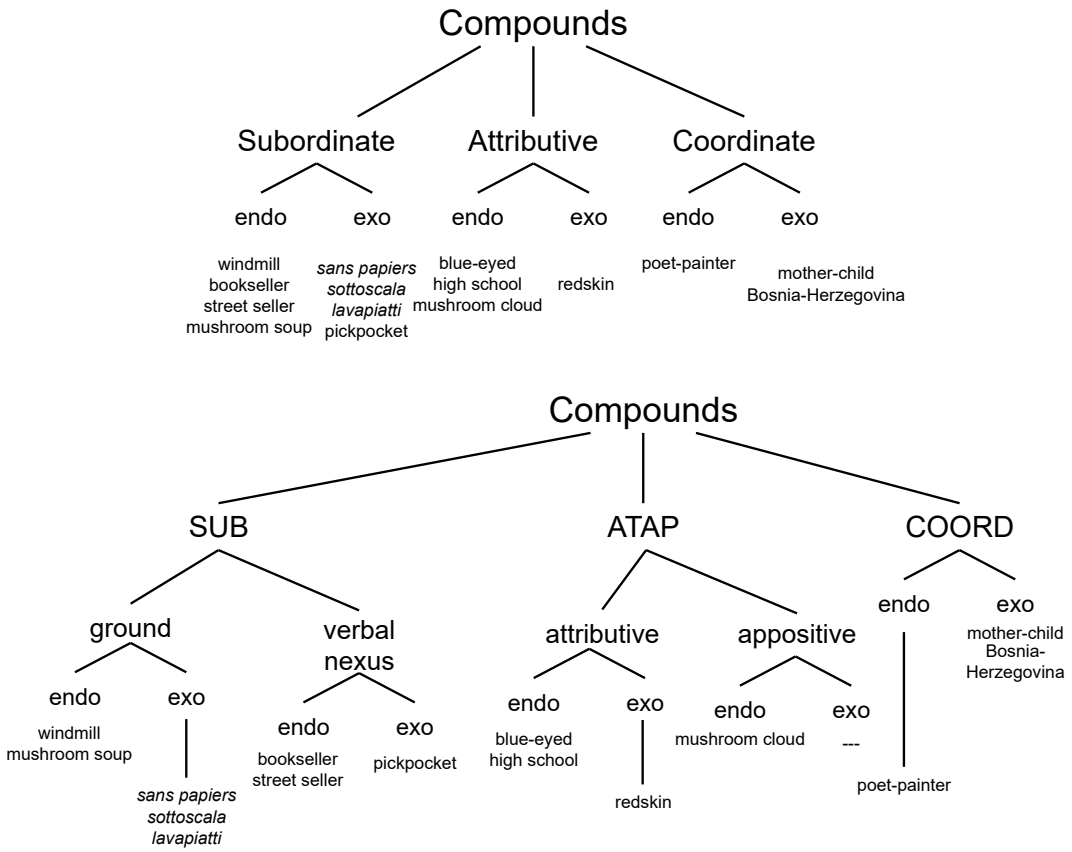


Figure 2.2: Comparison of the multilingual taxonomies of Bisetto and Scalise (2005) and Scalise and Bisetto (2009).

blue-eyed refers to a person who actually has blue eyes, whereas the German equivalent *blauäugig* can either also be endocentric if used literally as 'blue-eyed', or in other cases, can be used to mean 'naive', in which case it is exocentric. This classification was implemented in the annotation scheme of the MORBO/COMP database, which is one of the resources reported on later on in in Section 2.3 – Compounds in language data resources.

Hearkening back to Bloomfield (1933) and Marchand (1960), Scalise and Bisetto (2009) presented an extended classification scheme. Here,

1. **subordinate** (SUB) compounds are now further split into **ground** and **verbal nexus** compounds,

2. **attributive** compounds are now a sister group to appositive compounds, together forming the **attributive/appositive** (ATAP) group,

3. and the **coordinate** compound (COORD) group remains unchanged.

A visual comparison of the Bisetto and Scalise (2005) and Scalise and Bisetto (2009) taxonomies over the same set of examples can be inspected in Figure 2.2.

Ground compounds are compounds whose interpretation

> "depends on the possibility of varying the association between the features of the respective qualia structures [=bodies of semantico-

> encyclopedic information associated with the consituents]: these features are elicited from the context in which the compound formations are to be found".

In other words, the exact relationship between the two constituents can only be reliably determined by understanding the context(s) in which the compound is used.

In contrast, *verbal nexus compounds* are defined by the presence of a verbal head, whose non-head is then either its argument (*bookseller = sells books*) or its adjunct (*street seller = sells X on the street*).

The *attributive/appositive* distinction can be understood as follows: *attributive* compounds express a property of the head by modifying it using an adjective (*high school*) or a verb (*playground*), whereas in *appositive* compounds the property is expressed by a noun, whose one selected property acts as an attribute (the aforementioned *mushroom cloud*).

Scalise and Vogel (2010) point out that compounds exhibit an internal syntax – that is, their interpretation rests upon the ability to of a speaker to impute a syntactic relation between the components, citing the examples (20), (21), (22).

(20) ***taxi driver*** ← ***driver of a taxi*** (EN)

(21) ***hard ball*** ← ***ball which is hard*** (EN)

(22) ***poet painter*** ← ***poet and painter*** (EN)

Furthermore, they observe that there may be more than one acceptable interpretations like that, and more than one unacceptable, and that the disambiguation of these may require extra-linguistic knowledge. As an example, they cite the compound *water mill*, which may refer to a mill powered by water, located by water, or producing water, but not one that grinds, drinks, or is made out of water. Experimentally, this is reflected in Ó Séaghdha's 2008 dissertation, who reports that on a sample of 2000 English noun-noun compounds, two annotators achieved 66.2%[5] agreement in categorizing compounds into eight relational types.

Ackema and Neeleman (2010) directly discuss the involvement of syntax in the formation of compounds. While they argue that syntax is not directly involved in the creation of compounds, they argue that the competition between morphology and syntax dictates what sort of compound may or may not be formed in a given language – specifically, that the existence of a phrase with a one-to-one correspondence with regards to internal syntax blocks the existence its associated compound.

Czech compounds

As explained earlier, the starting point for this thesis is Czech, due to the unique insight we have into the language and the availability of relevant high-quality data. As a result, we present classifications of specifically Czech compounds.

[5] 0.62 Cohen's kappa

Štichauer (2013) presents an attempt to classify Czech compounds using three levels of categorisation, akin to the way Romance compounds are handled by Scalise and Bisetto (2009). The first level is the distinction between *coordinative*, *subordinative* and *attributive* compounds. The second level distinguishes between *exocentricity* and *endocentricity*, or *headedness* – in other words, whether or not the compound has a semantic head. The third level distinguishes between every possible combination of part-of-speech category of the input words and the part-of-speech category of the output compound in the format $[X + Y]_Z$, where X and Y stand for the part of speech of the parents and Z stands for the part of speech of the product compound. This style of description is used later on in this dissertation to statistically show the distributions of various compound types across various data sources.

As already touched upon earlier in Section 2.1.2, Bozděchová (1997) distinguishes two types of compounding in Czech, depending on whether the words entering the composition are formally modified or not. *Compounding proper*, which requires morphological adjustment of the input words, and *compounding improper*, which is the result of simple concatenation of a syntactic phrase with no morphological adjustments. In addition, Bozděchová puts forth a multi-level classification, starting from the part-of-speech category of the output compound and then proceeding to semantic criteria (considering the meanings of the input items, of the output compounds and the relationship between the output and the inputs).

Bozděchová proposes a hierarchical classification of compounds within the onomasiological theory of word formation. The classification is applied to a dataset of 3000 Czech compounds. The highest level is classified by the POS[6] of the compound. The middle level is classified by the type of referent that the compound names – e.g. *person*, *property bearer*, *place name* for nouns. The lowest level is the formal division of compounding into three categories – simple compounding proper, complex compounding proper, and compounding improper (juxtaposition). Compounding proper refers to the spontaneous coining of a two-rooted word by a speaker, spurred on by the need to name a particular object in a particular speech situation, which makes it a genuine word-formation phenomenon. This is contrasted with compounding improper, which is the phenomenon of syntactic expression gradually solidifying over time, which places it in the domain of syntax. In Czech, the phrase that is encoded by an improper compound can be reconstructed solely by finding an appropriate split-point and splitting the compound there with no morphological adjustments. For example, splitting the improper compound *vždy|zelený* "evergreen" yields the valid, correctly formed phrase *vždy zelený* "always green", whereas the proper compound *bělobřichý* "white-bellied" does not yield a correctly formed phrase when split without adjustment. Complex compounding proper corresponds to what we call *parasynthetic compounding*, and is described in detail in 3.1.1.

Moreover, it is taken into account whether the compound is a result of composition only or whether also other word-formation processes (derivation, conversion) were at play. For instance, the compound adjective in (23) was coined through composition proper, when the ending *-ý* in the first input adjective (*tmavý* 'dark') was dropped and an *-o-* interfix was used to concatenate it with the second adjec-

[6] In the introduction section for each POS, Bozděchová explains how it corresponds to a particular onomasiological category, e.g. nouns correspond to the onomasiological category of *substance*.

tive (*modrý* 'blue'). In (24), the input adjective (*tvrdý* 'hard') undergoes a similar formal modification, but the second item (the noun *hlava* 'head') is converted into an adjective through replacing the nominal ending by an adjectival one (*hlava* 'head' → **-hlavý* 'headed', which cannot be used separately in Czech). Analogically to this example of compounding and conversion in one step, in (25) the compound is formed through compounding and derivation (i.e., the addition of the agent suffix *-ec* to the input verb). A straightforward example of composition improper is the concatenation of two nouns to a compound adverb in (26). A reversal of the ordering of the input words is permissible, resulting in the compound verb in (27).

(23) ***tmav|-o-|modrý*** ← ***tmavý modrý*** (CS)
dark-blue.A dark.A blue.A

(24) ***tvrd|-o-|hlavý*** ← ***tvrdý hlava*** (CS)
stubborn.A hard.A head.N

(25) ***čern|-o-|oděnec*** ← ***černý odít*** (CS)
black dressed man.N black.A dress.V

(26) ***chvála|bohu*** ← ***chvála Bohu*** (CS)
thankfully.ADV praise.N God.N-DAT.SG

(27) ***blaho|přát*** ← ***přát blaho*** (CS)
congratulate.V wish.V wellness.N-ACC.SG

2.3 Compounds in language data resources

In this section, we alphabetically list data sources that are relevant to the modeling of compounding across languages. One of the strongest motivations for the research described in this thesis is the fact that compounds are often underrepresented in word-formation resources, and even when they are not, their handling is inconsistent across languages or even across individual datasets.

2.3.1 CELEX2

CELEX2 (Baayen et al., 2014) is a general lexical database covering English ($50,964$ items), Dutch (118.029 items) and German ($51,278$ items), which apart from word formation also covers inflection and syntactical properties of the included lexical items. The database covers compound structure as well – it includes the morphological segmentation of each word using nested parentheses, with an associated part-of-speech tag for each segment. Out of all the linguistic annotations provided in this resource, delimitation of the components (and the linking element, interfix, if present), POS of the components, and annotation of the internal structure using nested brackets (cf. (28) to (30)) were the most important. In the bracketed structures in German, some morphs are replaced with a representative form (cf. *gang* substituted by *geh*, which occurs in the infinitive *gehen* 'to go' in (28); but in the English example (30) *woman* is not used instead of *women*).

(28)
```
Umgangssprache ... Umgang+s+Sprache NxN ...
((((um)[V|.V],(geh)[V])[V])[N],
(s)[N|N.N],((sprech)[V])[N])[N] ...
```

(29) `Grossmachtpolitik ... Grossmacht+Politik NN ... (((gross)[A],(Macht)[N])[N], ((polit)[R],(ik)[N|R.])[N])[N] ...`

(30) `womenfolk ... women+folk NN ((women)[N],(folk)[N])[N] ...`

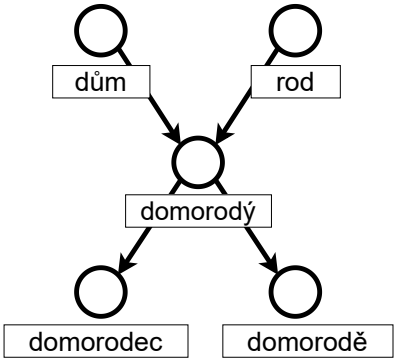
2.3.2 DeriNet

DeriNet is a lexical database of Czech where words that share a common root are arranged into tree-like graphs according to their morphological structure – from the morphologically simplest words (unmotivated words) to the most complex. Unmotivated words are represented as roots, and derivatives are linked to them, forming derivational trees. The database contains over a million entries, of which less than a half (ca. four hundred thousand) are corpus-attested. Originally, the resource handled exclusively derivation, but from version 2.0 (Vidra et al., 2019) onward, support for compounding has been added in the form of a binary yes/no compound flag for each lexeme, as well as by allowing the API to support a single lexeme having multiple parents. While derivatives are linked to a single ancestor, compounds can be connected to two or more ancestors. As a result, word-formation families containing compounds are no longer trees, but rather directed acyclic graphs (DAGs), or multi-rooted trees. The lemmaset is based on MorfFlex[7] (Hajič et al., 2020), a morphological dictionary of Czech aiming for complete average. The latest public version, DeriNet 2.1 (Vidra et al., 2021a) (apart from version 2.2 released as part of this dissertation – cf. Chapter 4), only contained nouns, verbs, adjectives, and adverbs.

Some compounds were identified (=their parents have not yet been found) based on heuristics and lexical lists of compound parts. When the compounds both with and without the links to their ancestors are counted (all of them having the explicit Boolean compoundhood flag set to `true`) together with the derivatives of all these compounds, the number totals to 45 thousand corpus-attested compounds available in DeriNet 2.1 (Vidra et al., 2021a). This version contains over two thousand compounds with assigned parents. The left graph in (31) shows the unmotivated nouns *dům* 'house' and *rod* 'kin' as ancestors of the adjectival compound *domorodý* 'native', from which the noun *domorodec* 'native man' and the adverb *domorodě* 'in a native way' are derived.

[7] http://hdl.handle.net/11234/1-3186

(31)



2.3.3 GermaNet

GermaNet (Henrich and Hinrichs, 2010; Hamp and Feldweg, 1997a) is a database that relates German verbs, nouns, and adjectives. It currently contains $215,000$ lexical units, of which $121,655$ are split compounds. This source lists for each compound the lemmas of two immediate ancestors from which it was composed ((32) to (34)). The ancestors provided are existing words, not just strings occurring in the compound (cf. (33) where the verb *abbiegen* 'to turn' is given, because **Abbiege* is not a separate word in German). Compounds with more than two roots are split in succession; see (34) where the second ancestor is a compound which is analyzed in a separate entry in the resource. For the first component, two possibilities are given, if both are equally relevant (cf. the action noun *Umfrage* 'survey' and the verb *umfragen* 'to survey' in (34)).

(32) `Umgangssprache Umgang Sprache`

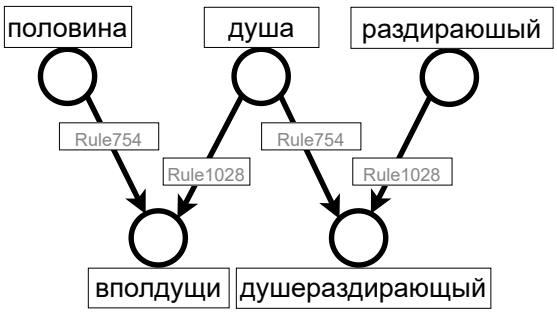
(33) `Abbiegeassistent abbiegen Assistent`

(34) `Umfrageteilnehmer Umfrage|umfragen Teilnehmer`

2.3.4 Golden Compound Analyses

Golden Compound Analyses (Vodolazsky and Petrov, 2021) is a collection of around 2000 Russian two-parent compounds hand-annotated for the purposes of training a Russian compound splitter. Each entry also contains a morphological rule by which the compound's parent is incorporated ininto the word. While it also contains information on POS of the parents and of the resulting compound, there is no information about derivational parents or children. The structure is shown in Figure 35.

(35)



2.3.5 MORBO/COMP

MORBO/COMP (Guevara et al., 2006) is a database of compounds covering 23 languages, including the ones in scope except for Czech. It provides information consistent with the classification of Scalise and Bisetto (2009). The database describes the part-of-speech of each compound as well as its constituents, centricity, syntactic headedness, semantic head (if present), linking element, and gloss in English. In Table 2.1, the annotation provided is exemplified by three nominal Italian compounds composed of words from different POS categories (cf. 2nd and 3rd column).

The compound's lemma (1st column) is followed by its POS category (2nd column), the POS categories of the components (column Struct[ure]), syntactic relation between the components (Class: subordinate/attributive/coordinate), endocentricity (End[ocentric]: True/False), placement of the semantic head (Head-C), placement of the syntactic head (H-S), the form of the first component (1st-C) and of the second one (2nd-C), and the gloss.

While the first compound (*madrelingua* 'mother tongue') is endocentric with the right component playing the role of head, the latter two are exocentric (and headless). The components are listed as they occur in the compound (8th and 9th column), they may not be existing words (cf. the third compound in the table). While potentially highly useful for the purposes of this thesis, as of 2024 the project seems to have been discontinued and the data are not publicly available.

Compound	POS	Struc	Class	End	Head-C	Head-S	1st-C	2nd-C	Gloss
madrelingua	N	[N+N]	SUB	Tru	right	right	madre	lingua	mother+tongue
mano lesta	N	[N+A]	ATT	Fal	none	none	mano	lesta	quick+hand = thief
dormiveglia	N	[V+V]	CRD	Fal	none	none	dormi	veglia	sleep+be awake = dozing

Table 2.1: Annotation of Italian compounds in the MORBO/COMP database.

2.3.6 UniMorph

UniMorph (Batsuren et al., 2022b) is a massive-scale coordinated effort by a team of researchers from all over the world to build a collection of morphological resources covering 169 different languages. As a result, its coverage varies wildly. For the

purposes of this thesis, only the Spanish ($42,825$ derivatives; 130 compounds) and French ($72,789$ derivatives; 161 compounds) branches of *Unimorph* are relevant. In ex. (2.3.6), the first two examples are French, the second two are Spanish.

(36)
```
feuille     portefeuille        N:N porte-
pigeon      pigeonite           N:N -ite
Estados     Unidos              estadounidense  N:N -ense
utilizar    utilización         V:N -ción
```

2.3.7 Wiktionary

Wiktionary presents a massive amount of compounds for many languages. Unfortunately, the data resource is very inconsistently structured, and in practice, it is difficult to extract compounds and/or descriptions thereof safely, because the labeling conventions are in practice very inconsistent. As a result, we only accepted entries that which

a) were explicitly tagged as *compound*, and

b) had their parents formatted in Wiktionary's standard format, and

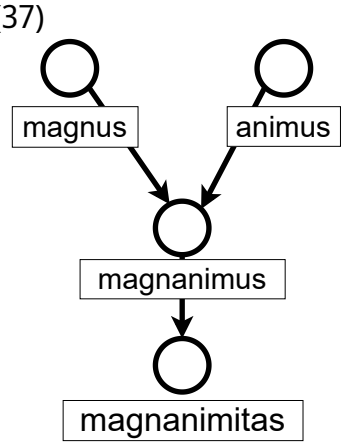
c) did not contain hyphens on either side (so as to rule out affixes).

Trying to accommodate other matches, such as non-standardly formatted parents, yielded a large number of false positives, because Wiktionary annotation turned out to be very inconsistent. As a result, we were able to scrape 257 French compounds and 384 Spanish compounds, alongside their parents.

2.3.8 Word Formation Latin

The Word Formation Latin database contains more than 3 thousand Latin compounds, with their derivatives (Litta et al., 2016). The database is organized in a way similar to DeriNet; cf. the right graph in ex. (31) modeling the Latin adjective *magnanimus* 'high-spirited' as being formed by combining the adjective *magnus* 'high' and the noun *animus* 'spirit', and giving rise to the noun *magnanimitas* 'high-spiritedness'. The structure is shown in Figure 37. meaning that we were able to extract both compounds and their derivatives, totaling $3,198$ entries.

(37)



Authors	Lang(s)	Approach	POS scope	Parent no.
Khaitan et al. (2009)	en	Split-point	Any	Any
Fritzinger and Fraser (2010)	de	Split-point	Any	2
Henrich and Hinrichs (2011)	de	Valid-output	Nominal	2
Clouet and Daille (2014)	en, ru	Valid-output	Any	2
Martin Riedl (2016)	de, nl, en	Split-point	Any	Any
Krotova et al. (2020)	de	Split-point	Nominal	Any
Svoboda and Ševčíková (2021)	cs	Valid-output	Any	Any
Vodolazsky and Petrov (2021)	ru	Valid-output	Any	2

Table 2.2: Comparison of various compound splitters sorted by year of publication.

2.4 Compounds in procedural tools

By procedural tools, we mean programs, algorithms, and models that, unlike the data sources described in the previous section (2.3), do something with the input word or words to produce the desired output, as opposed to simply looking for the input in a fixed database. As a result, procedural tools can handle any word, not just a limited set that happens to be covered by the given database, but their output may be incorrect much more often.

In this section, we will focus primarily on so-called compound splitters, but will also briefly go over other tools that deal with compounding in one way or another.

2.4.1 Compound splitters

A compound splitter (a.k.a decompounder) is any tool that takes a compound word as input and decomposes it into two or more linguistic sub-elements in some way. In contrast with static data resources, compound splitters are procedural, and as a result should not output an out-of-vocabulary error when presented with a novel coinage. We present a short non-exhaustive overview and classification of compound splitters that have been presented in the literature for the languages in scope. It is summarized in Table 2.2 .

Split-point compound splitters simply return the index of the place wherein the given word should be split, which is what Khaitan et al. (2009) applied to English using normalized frequency and character n-grams. While the split-point approach allows the task to be handled as a regression or classification problem (as opposed to a sequence-to-sequence approach), the drawback is that in many languages a point-split systematically fails to yield valid lemmas. This is of little concern in English, where examples of this situation are rare, but such an approach started being a problem once the attention in NLP shifted to more morphologically complex languages.

For example, in the Dutch example (38), we see an *-e-* interfix. Inserting a split-point at *bruide.gom* would result in **bruide*, which is not a Dutch word; conversely, a split-point at *bruid.egom* results in the similarly nonsensical **egom*. This problem can be solved by building split-point splitters so that they drop interfixes, for instance by using a list of them like Henrich and Hinrichs (2011) did. However, in some languages, even interfix dropping falls short, and the split-point approach starts being impractical. For example, splitting Czech *zeměpis* as *země.pis* results

in **pis*, which is not a valid lemma (cf. ex. 59). Instead, the appropriate output would be *psát* 'to write'. Similarly, Russian *вод.о.провод* (cf. example51) cannot be point-split correctly, as it would yield the non-existent **вод* (40). Sporadically, the problem appears even in English, where *women.folk* (ex. 41) would yield *women*, which is a plural, and therefore not a lemma.

(38) ***bruidegom*** → ***bruid*** ***gom*** (NL)
bridegroom.N bride.N groom.N

(39) ***zeměpis*** → ***země*** ***psát*** (CZ)
geography.N earth.N write.V

(40) ***водопровод*** → ***вода*** ***провод*** (RU)
water_piping.N water.N conduit.N

(41) ***womenfolk*** → ***woman*** ***folk*** (EN)
N N N

Valid-output compound splitters attempt to deal with the previously described problems. This may be achieved by equipping a split-point splitter a table of rules and/or a vocabulary or corpus, like Clouet and Daille (2014) or Vodolazsky and Petrov (2021) did, or by treating the task as a sequence-to-sequence procedure outright, which is an approach that Svoboda and Ševčíková (2021) took.

Some compound splitters restrict themselves to nominal compounds, such as the splitters of Henrich and Hinrichs (2011) or Krotova et al. (2020) for German; general compound splitters handle any compound regardless of the POS of either the parents or the compound. Finally, splitters differ in how many parents they return. Some splitters either return exactly two parents, like the splitter of Fritzinger and Fraser (2010); others return any number of parents, like Svoboda and Ševčíková's 2021 splitter.

Out of the languages in scope, compound splitters have been built for five of the seven languages. To the best of our knowledge, no splitters have been built for French or Spanish at the time of submission of this thesis, with the exception of *DériF* (Namer, 2003), which only handles neoclassical compounds like those in example 42. Before the introduction of *Czech Compounds Splitter* (more details in Chapter 3.2), Czech also had no procedural tool for identifying or splitting compounds. Derivational Analyzer of Czech (Derivancze; Pala and Šmerk, 2015), as its name suggests, is limited to derivational relations in the lexicon of Czech.

(42) ***psychologie*** → ***-psych-*** ***-log-*** (FR)
psychology.N soul.NEOC word.NEOC

Splitting of Czech compounds has been addressed by *Czech Compound Splitter* (Svoboda and Ševčíková, 2022), which is the predecessor of *WFA.ces* (Svoboda and Ševčíková 2022; Chapter 3.3) and *PaReNT* (Svoboda and Ševčíková 2024; Chapter 3.4). Its primary capability, compound splitting, can be understood as a special case of parent retrieval limited to confirmed compounds. Analogously, it also performed compound identification, which is word formation classification limited to a binary set of classes – *compounds* and *non-compounds*.

Henrich and Hinrichs (2011) linked German nominal compounds to their respective parents in GermaNet (Hamp and Feldweg, 1997b) using an ensemble of pattern-matching models with an accuracy of 92%. Sugisaki and Tuggener (2018)

achieved an F1-score of 92% for finding split-points in German compounds using an unsupervised approach, although they also restricted their efforts to noun-headed compounds only. Ma et al. (2016) achieved an accuracy of 95% using a neural approach trained on the aforementioned GermaNet. Their model performed both splitting and identification of compounds, with the accuracy being an aggregated score of both. Krotova et al. (2020) achieved an accuracy of 96% with a deep-neural model trained on GermaNet data, again restricting themselves to nominal compounds. Clouet and Daille (2014) achieved F1-scores of 80% and 63% respectively for finding split-points in English and Russian compounds using a corpus-based statistical approach on manually split compounds.

It is worth mentioning that apart from the languages in scope, a significant amount of research has been dedicated to the study of Sanskrit compounds. This ranges from early, relatively simple rule-and-lexicon based attempts by Huet (2005), who lists no accuracy in his study, to Hellwich and Nehrdich's (2018) deep-learning solution trained on a corpus of $560,000$ Sanskrit sentences with its compound split-points annotated, achieving an accuracy of 96%.

2.4.2 Other procedural tools

Out of all the other already-established tasks in computational linguistics and natural language processing, the closest to compound splitting is probably morphological segmentation, in that morphological segmenters are typically expected to shed light into the internal structure of compounds. These tools typically return a list of morphs or morphemes (e.g. *happiness* -> [*happ*, *i*, *ness*] or [*happ*, *y*, *ness*], or seasickness -> [*sea*, *sick*, *ness*]). Such morphs or morphemes may or may not be valid words.

A well-known segmenter is Morfessor, introduced as unsupervised in 2002 by Creutz and Lagus, extended into semi-supervision in 2010 by Kohonen et al., and generalized beyond morphological segmentation in 2013 by Virpioja et al. It has been shown that its application on two languages can improve machine translation (Grönroos et al., 2018).

Other morphological segmenters, however, are tailored to a particular language. For example, Cotterell et al. (2016) built one for English using weighted context-free grammars, and the SIGMORPHON 2022 Shared Task on Morpheme Segmentation (Batsuren et al., 2022a) challenged researchers to segment words in 8 different languages using training data from the aforementioned Unimorph database (Batsuren et al., 2022b). While this resource does cover segmentation, extracting the information may be difficult. To address this problem, a multilingual annotation scheme for morphological segmentation has been proposed by Žabokrtský et al. (2022), potentially streamlining the development of multilingual segmenters in the future.

Another task related to parent retrieval is *stemming*. The now classic Porter algorithm was developed in 1979 and published in 1980. There is also a programming language built by Porter, specifically tailored for writing stemmers, called Snowball (Porter, 2001), in which a Czech stemmer called Czech Snowball Stemmer (Chmelař et al., 2011) was implemented.

It has been demonstrated in several languages that NLP tasks such as information retrieval and text classification are significantly improved if the input data is

first stemmed. This has been shown for Swedish (Carlberger et al., 2001), Albanian (Biba and Gjati, 2014) and even Czech (Dolamic and Savoy, 2009), which suggests that the task of parent retrieval, addressed in the present thesis, might also potentially be of practical interest for the purposes of applications like information retrieval.

Parent retrieval, under our interpretation, differs from stemming in that

- it requires the input to have already been lemmatized;
- it *has to* return a lexical item that appears in the given language's usage as an independent item; and
- it only returns the immediate ancestor of the input word.

For instance, given the English word *unhappiness*, the string **happi* in (43) might be considered to be a correct stemming, despite the fact this string does not occur by itself in written English. When stemming, emphasis is placed on lumping words like *unhappiness*, *happiness* and *happiest* under a single label (**happi* in this case), be it linguistically correct or not. In contrast, (44) or alternatively (45) is what we would expect a parent retriever to do.

(43) ***unhappiness*** ← ****happi***

(44) ***unhappiness*** ← ***unhappy***

(45) ***unhappiness*** ← ***happiness***

To conclude, the coverage of compounds in data sets and tools highly depends on the language in question, with German and Dutch receiving the most attention and French and Spanish the least. In this dissertation, we will make use of all the data sets listed in this chapter, except for MORBO/COMP, which unfortunately does not seem to be available.

3. Developing tools for compound analysis

This chapter presents a series of three tools, each an improvement over the next – *Czech Compound Splitter*, *Word Formation Analyzer for Czech*, and *PaReNT*. Originally, the plan had been to start with Czech and then later spread out into the other language. While that ultimately did happen, what also happened is that we spread out into other types of words formation. *Czech Compound Splitter* is therefore the only tool in the series that is specialized in compounding only – the following two can also handle derivation. Before we present the three tools, however, we will introduce some of the considerable number of formal challenges presented by the languages in scope in Section 3.1.1, and the reasoning and principles behind the solution employed to tackle them in Section 3.1.2.

3.1 Problems and the solution

3.1.1 Challenges

Sometimes, the process of compounding is a simple concatenation of otherwise freely occurring words, as in ex. (46)

(46) ***chevalvapeur*** ← ***cheval*** + ***vapeur*** (FR)
horsepower.N horse.N steam.N

Often, compounding is however far more complex than that, and this section demonstrates the non-triviality of handling these words.

When dealing with compounds, an immediate problem arises – where lies the boundary between compounding and other word-formation processes, and between compounding and syntax? This is a heated topic in morphology because the question begs the answer to other unsolved questions, such as the precise definitions of wordhood and morpheme boundness, as already addressed in Chapter 2. As a result, numerous edge cases exist – and for computational purposes, these need to be resolved one way or another.

(47) ***ondersteunen*** ← ***onder*** + ***steunen*** (NL)
support.V under.P support.V

In ex. (47), *ondersteunen* can either be considered a compound, or it can be understood as a derivative of *steunen* with the prefix *onder-*. The case for the compounding interpretation can be made by observing that *onder* syntactically behaves like a free word in German. However, the productivity pattern of compounds with *onder* is much more reminiscent of derivation. Furthermore, Lieber and Štekauer (2011) propose that roots should have more *semantic substance* than affixes, but it is difficult to argue that *onder* has more semantic substance than for example the undisputed affix *pre-*. As a rule of thumb, we tend to consider edge cases like this to be compounds if the leading element is two syllables or longer, and derivatives otherwise.

Another type of word straddling the boundary is the already-mentioned neoclassical compound. It is a special case of compounding wherein elements borrowed from Ancient Greek and Latin are combined either with each other or with free words. We term such elements *neoclassical constituents*.[1] For example, in the English *monolog*, neither the first constituent **mono-* nor the second constituent **-log* can be attested on their own.

Neoclassical compounds, under our interpretation, constitute what Ološtiak and Vojteková (2021) consider *semi-composition* and *quasi-composition*. The German noun *Soziologie* 'sociology' in (48) is an example of *quasi-composition* in their framework. In a broader sense, chemical compounds also satisfy the definition of semi-composition, as in (49).

(48) ***-soci-*** + ***-log-*** → ***Sozi|-o-|logie***, but no ****Sozi*** nor ****logie*** (DE)
-soci-.NEOCON -log-.NEOCON sociology.N

(49) ***-tetra-*** + ***chlor*** + ***ethylen*** → ***tetra|chlor|ethylen***, but no ****tetra***
-tetra-.NEOCON chlorine.N ethylene.N tetrachlorethylene.N
(CS)

On the other end of the spectrum, there is the fuzzy boundary between compounding and syntax, or what Lieber and Štekauer (2011) term the *macro question*. It may not always be clear at which point a given syntactic phrase has 'solidified' enough to be considered a word on its own. The Czech tradition would, for instance, consider the adjective *vždyzelený* mentioned in Section 2.2 to be a single word, but this largely relies on orthographic convention, which may not be reliable in English (cf. *flowerpot*, *flower-pot*, *flower pot* are all valid spellings) or other languages (e.g. French *portemonnaie* vs. *porte-monnaie* 'wallet', Italian *mezzaluna* vs. *mezza luna* 'half-moon', Czech *machinelearningový* vs. *machine learningový* vs. *machine-learningový* 'related to machine learning').

Morphological variation

Some compounds are formed by the mere juxtaposition of existing words. However, this is often not the case. In ex. (50), we observe the addition of an *-e-* interfix between the constituents. In some languages, variation goes beyond interfix addition.

In *вод.о.провод* (cf. ex. (51)), the interfix replaces the ending of the first constituent **вод*. Internal flexion also appears, like in the English *womenfolk* (ex. (52)), where the first constituent is inflected for plurality. Additionally, stem allomorphy often appears. It may takes the form of vowel alternation, for example /o/ → ∅, like in (53).

(50) ***bruidegom*** ← ***bruid*** + ***gom*** (NL)
bridegroom.N bride.N groom.N

(51) **водопровод** ← **вода** + **провод** (RU)
water piping.N water.N conduit.N

(52) ***womenfolk*** ← ***woman*** + ***folk*** (EN)
N N N

[1] Also known as baseoid under (Ološtiak and Vojteková, 2021)

(53) **любв|-е-|обильный** ← **любо̲вь** + **обильный** (RU)
affectionate.N love.N abundant.A

In (54), the compound is traced back to the noun phrase *porta lettere* 'carries letter' where both words are inflected – the first for the 3rd person (infinitive is *portare* 'to carry') and the second for number (singular is 'lettero'). Additionally, there are compounds that cannot be meaningfully split into two parents; cf. the compound in (55) which is composed of a multi-word numeral expression (*dvě a půl* 'two and a half') and the final part which was converted from a noun (*léto* 'year.N' → *-letý* '-year.A').

(54) ***portalettere*** ← ***porta*** ***lettere*** (IT)
postman.A carries.V letters.N.PL

(55) ***dvaapůlletý*** ← ***dvě*** + ***a*** + ***půl*** + ***léto***, but no ***letý***
two-and-a-half-year-old.A two.NUM and.C half.NUM year.A
(CS)

Parasynthetic compounding

One of the ways compounding interacts with other aspects of language is when it occurs simultaneously with some other word-formation process, which as previously mentioned we call *parasynthetic compounding*.

Compounding and derivation in one step (56) as well as compounding and conversion in one step (57) are possible, often accompanied by vowel and consonant changes; for instance, in (57) two cases of stem vowel alternation (∅ ← /e/ in *ps* ← *pe̲s* and /o/ ← /e:/ in *vo̲d* ← *vé̲st*), a stem consonant alternation (/d/ ← /s/ in *vod̲* ← *vés̲t*), and an interfix insertion all occur at the same time. Note that in parallel to (57), the compound in can also be analysed as an output of compounding and conversion in one step (59). In contrast, for *psovod* such an alternative is not available because **vod* is not attestable as a separate noun in Czech.

(56) ***modr|-o-|oký*** ← ***modrý oko***, but no **oký* (CS)
blue-eyed.A blue.A eye.N

(57) ***ps|-o-|vo̲d*** ← ***pe̲s*** ***vé̲st***, but no **vod* (CS)
dog-handler.N dog.N lead.V

Whether or not a given compound is parasynthetic may be a matter of analysis. This leads to difficulty in annotation. Similarly to the examples just discussed, the second parent of Czech *přímotop* (ex. (58)) can only be *topit*, not **top*, which is a bare stem and not a word, so the motivating process behind this word must be compounding together with conversion. However, the similar *krvotok* (ex. 58) can be either analogously understood as compounding together with conversion, assigning the verb *téci* 'to flow' as the second parent, or we can assign the noun *tok* 'flow' as the parent and understand the motivating process as simple compounding proper (cf. Bozděchová (1997)).

(58) ***přímotop*** ← ***přímo*** + ***topit*** (CS)
heater.N directly.ADV heat.V

(59) ***krvotok*** ← ***krev*** + ***téci/tok*** (CS)
bloodflow.N blood.N to flow/flow.V/N

The reason words like *haired as in red-haired or *legged as in bow-legged are unattested is probably because the base assumption is all humans have these body parts, and therefore such words would carry minimal information value.

(60) ***белокурый*** ← ***белый*** + ***кура*** , but no **курый* (RU)
white-haired.A white.A hair.N

(61) ***blauwogig*** ← ***blauw*** + ***oog***, but no **ogig* (NL)
blue-eyed.A blue.A eye.N

(62) ***blue-eyed*** ← ***blue*** + ***eye***, but no **eyed* (EN)
A A N

(63) ***albicapillus*** ← ***albus*** + ***capilla***, but no **capillus* (LA)
white-haired.A white.A hair.N

Looking at the difficulties outlined above, it soon became clear that in a multi-lingual setting, there was very little we could assume that would help us computationally analyze compounds. We could not assume we could find the parents of compounds by calculating a split-point, because all of the languages in scope except English use interfixes and exhibit frequent stem allomorphy. We could not rely on string overlap, because e.g. *psovod* contains neither *pes* nor *vést*. Also, many compounds contain multiple parents, which makes dictionary-search approaches inefficient.

It also became clear that the amount of linguistic knowledge required to cover a new language was considerable despite the amount of data available, and it would be necessarily to employ a language expert for each new language in order to support a multi-lingual rule-based system for compound analysis, which would not scale very well in the future. Another observation regarding compound analysis is that for a human, it is intuitively very easy. No Czech person has trouble figuring out that *psovod* comes from *pes* and *vést* (or alternatively vodit), although the amount of morphological reconstruciton necessary to arrive at these parents is considerable.

Fortunately, the one thing we did have was data, and we decided to use a general technique that tends to be very good for things that are difficult to program explicitly but easy for human intuition – deep learning.

3.1.2 A general solution

Deep learning is generally understood as the branch of machine learning that deals with deep neural network models using representation learning. A neural network is a computational model historically inspired by the functioning of biological neural structures, but has since diverged from its original goal of modeling biological structures and instead focused on the development of highly general models applicable to a wide plethora of input data structures and tasks.

As a result, a large portion of this dissertation relies on deep learning in one or another, we include a short introduction explaining the basic concepts pertaining to this topic for readers who may not be familiar with it. It must be stressed that this area of research is both wide and deep, and that the concepts presented here

are carefully selected so that they are immediately relevant to the topics of the dissertation.

Multi-layer perceptron

The most basic type of neural network is called a *multi-layer perceptron*, or alternatively *feedforward layer* (FF) if used as part of a larger model (Rumelhart et al., 1986).

In an MLP, the input represented as a vector **v** of size n is multiplied by a weight matrix **W** of shape $[n \times m]$, a bias vector **b** is added to it, and the resulting vector **u** of size m has the neural network's so-called *activation function* applied to it. Finally, the vector **u** is multiplied by a reshaping matrix **X** of shape $[m \times o]$, where o is the size of the desired output vector, and passed through an output function, whose choice depends on the task at hand. For regression tasks, the identity function is used, but for classification tasks, the *softmax*[2] output function is typically utilized, which is defined as

$$softmax(\mathbf{o})_i = \frac{e^{o_i}}{\sum_{j=1}^{K} e^{o_j}},$$

where o represents the number of classes being predicted, and the values of the *softmax* output representing the probabilities assigned to each class by the model.

The step represented by the multiplication of the input by **W** and the addition of b is known as the *hidden layer* of the given MLP. It can be shown using what's known as the Universal Approximation Theorem (Cybenko, 1989; Hornik, 1991) that any continuous function from $\mathbb{R}^n$ to $\mathbb{R}^m$ can be approximated by a MLP which is either sufficiently wide (= m is sufficiently large) or deep (= there is a sufficient amount of hidden layers). Since strings of characters can be represented as vectors (though achieving that is not as simple as just using a MLP – see Section 3.2.2 for details), it stands to reason that neural networks were just the general solution that we were looking for.

Training of neural models

In the setup of supervised learning, neural networks are generally trained on a dataset of (*input*, *expected output*) pairs using *stochastic gradient descent* (SGD) (Rumelhart et al., 1986; Robbins and Monro, 1951) or its variants (SGD with momentum (Polyak, 1964), SGD with Nesterov momentum (Nesterov, 1983), *ADAM* (Kingma and Ba, 2014)...). The usage of SGD necessitates the selection of a so-called *loss function*, which is, roughly speaking, a function of two variables which returns a single variable that in some sense describes the 'closeness' or 'similarity' of the two input variables. The higher the loss function, the less similar the two inputs are In the case of classification, this is usually categorical crossentropy (Mannor et al., 2005), defined as

$$H(e, \hat{e}) = -\sum_{i=1}^{C} e_i \log \hat{e}_i,$$

[2] *softmax* is a differentiable function that takes a vector and re-scales it so that its elements sum to 1. This makes it so that the output vector can be re-interpreted as a distribution.

where C is the number of categories being predicted, e is the expected output (which is a one-hot vector corresponding to the correct category), and $\tilde{e}$ is the output distribution, and e_i is the i-th entry in e.

The purpose of the training procedure is then to set the network parameters so that the loss function's outputs are minimized. In SGD, the entire neural network is interpreted as a single differentiable function parameterized by the network's weight matrices and bias vectors. First, the neural network's weights are randomly initialized. Then, in each training step, a random example pair (i, o_gold) is selected from the dataset, and the output o_pred from the neural network is saved. Then, the gradient of the loss with respect to the neural network's parameters is calculated. For this purpose, an algorithm called *backpropagation* is used. Backpropagation uses the fact that each layer of the model is a function, and that the forward pass through the neural network is therefore a composition of those functions. It can then, starting from the output function, use the chain rule of derivatives to find the gradient of each layer and therefore of the whole model.

Then, the gradient is subtracted from the neural model's weights. The intuition behind this step is that subtracting the gradient sets the model's weight to what would have resulted in zero loss. However, the model would never generalize this way (since it would only learn to recognize that one example), so the gradient is first element-wise multiplied by a small number called the learning rate. The higher the learning rate, the faster the model trains, but the danger of the model not converging properly or at all also gets higher. It therefore often leads to the best results if the learning rate is dynamically adjusted throughout the training process using a *learning rate schedule*.

In Python-like pseudocode:

```
model.initialize_weights()
learning_rate = 0.01
epochs = 10

for epoch in epochs:
    for example in dataset:
        i, o_gold = example
        o_pred = model.classify(i)
        loss = Categorical_crossentropy(o_pred, o_gold)
        gradient = take_gradient(model.weights, loss)
        update = -1*(learning_rate * gradient)
        model.weights = model.weights + update
```

Having trained a model, it is now critical to gauge whether or not the model actually performs the given task well. Since the model is based around large matrices of numbers which are difficult to interpret, and the fact that it learns without direct human involvement, a rigorous methodology must be implemented to ensure that the model performs as it should.

Evaluation of neural models

Our evaluation methodology was guided by the important observation that the data on which a model is evaluated must be different from the data it was trained on.

This is because of a phenomenon known as overfitting, where, roughly speaking, the model learns the training examples 'by heart' instead of generalizing, which makes it perform excellently on learned data, but terribly on unseen data. Before model development, it is therefore necessary to split the data into three parts which do not overlap:

1. The training (*train*) set, which is used to train an array of models based on various hyperparameter (such as layer size or learning rate) settings (60 % of the data);

2. The development (*devel*) set, which is used to select the best model from the array based on one or more *evaluation metrics* (20 % of the data);

3. The evaluation (*eval*) set, which is used to calculate evaluation metrics for the purposes of model publication or deployment (20 % of the data).

The split between the devel and eval sets exists because the act of selection from an array of models introduces bias into the evaluation. The choice of evaluation metric largely depends on the task at hand, but for the classification and compound splitting/parent retrieval models presented in this thesis, we used these:

Accuracy refers to the number of times the model predicted correctly divided by the total amount of predictions. Accuracy is very simple, but does not take into account type of error[3] or potential imbalance of classes in the devel and eval data.

Precision refers to all correctly predicted instances of class (True Positives) i divided by all instances of class i. Unlike accuracy, Precision must be calculated for each class separately.*Sensitivity* (*True Positive Rate*; *TPR*; *Recall*) refers to all True Positives divided by all True Positives + all False Negatives. *Specificity* (True Negative Rate; *TNR*) refers to all True Negatives of class i divided by all False Positives + all True Negatives.

Balanced accuracy refers to the sum of Sensitivity and Specifity divided by two and averaged over all of the classes. In classification, it is important to take into account that there exists a *dummy accuracy* or *no information rate*, which is defined as $\frac{1}{num\_of\_classes}$. Roughly speaking, this is the accuracy of a model that predicts by guessing at random, e.g. predicting a binary (e.g. predict *Red* vs. *Blue*, with no other values possible) by tossing a coin Models performing with balanced accuracy around this value are not valuable or useful. The purpose of balanced accuracy is to take into account potential imbalance in the devel or eval data. For example, in a dataset of 100 examples, where 99 cases are *Red* and 1 is *Blue*, a dummy model that always guesses *Red* would get 99% accuracy, which looks impressive at first glance. However, it would get 50% balanced accuracy, revealing that it is not actually a useful model.

Finally, the F_1 *score* for binary classification is defined as the harmonic mean of Precision and Recall as

$$F_1 = 2 \times \frac{Precision \times Recall}{Precision \times Recall}.$$

The F_1 score is, like balanced accuracy, useful for imbalanced datasets.

[3]This can be a huge problem in applications such as medical machine learning, where misclassifying a tumor as a healthy bundle of cells is a much more costly mistake than the opposite.

For about 38% of the hand-annotated compounds in our dataset, there was ambiguity as to which parents they should be linked to. For instance, *monoprogramový* 'having a single programme' may be considered to be either composed of the neoclassical constituent *-mon-* and the adjective *programový*, or it alternatively may be composed of *-mon-* and the noun *program*, which would be derivation and compounding in one step. For the purposes of evaluation, both were considered to be correct splittings.

Additionally, a more relaxed metric was proposed by us which considers a predicted parent-candidate to be correct if it belongs to the same morphological family as the annotated parent. This metric is referred to as *root accuracy*, because all items of a morphological family are represented as a tree structure with the unmotivated word as the root node in DeriNet. DeriNet data are used to determine whether or not the predicted parent-candidate shares the same morphological family as the annotated parent. The solutions described in the following section exhibit different weaknesses and strengths.

3.2 *Czech Compound Splitter*

For some languages such as Sanskrit or German, compounding had been mapped and modeled extensively in both static data resources and procedural tools, but this was until 2021 not the case in Czech.

This chapter focuses on modelling compounding in Czech, which is a language where compounds are nearly always represented in writing as a single string of graphical symbols unbroken by whitespace. The problem we tackle is twofold: a) upon being given a graphical word, to decide whether or not it is a compound; and b) upon being given a confirmed compound, to return the citation forms of its base words (from here: *parent words* or *parents*). Task a) will be referred to as *compound identification* and is approached as an instance of binary classification; and task b) will be referred to as *compound splitting*. The tasks can be seen as part of the more general problem of morphological segmentation, which refers to the splitting of a word into morphemes (affixes, roots, endings).

After a brief report on the compilation of the data set including examples of some challenges of Czech compounding (Section 3.2.1), the experiments are described and their performance is compared in Section 3.2.2. The solutions we implemented include a baseline solution which performs compound splitting only. A more advanced approach based on phonemic string similarity we call *Interlexical Matrices of Likeness*, or $IML()$, is also limited to compound splitting. Finally, a deep learning based tool dubbed *Czech Compound Splitter* was trained, which simultaneously carries out both compound identification and compound splitting.

3.2.1 Data

The training data of *Czech Compound Splitter* had two primary components – a small dataset of manually-annotated genuine Czech compound words taken from DeriNet and a large set of mostly nonsensical but roughly correctly formed compounds synthetically generated by combining non-compounds together.

Manual annotation of DeriNet data

The compilation procedure began by extracting $1,500$ words from the DeriNet word-formation resource that had previously been labelled as having compound status. As their parent words had not been yet identified, this had to be done by hand. 53 were dropped, because they had been labelled as compounds mistakenly (*levopimar*, a medicine brand name), or are derivatives of compounds (e.g. the adverb *velechytře* derived from the adjective *velechytrý* 'very clever'). After this cleanup process was done, $1,447$ compounds remained in the data set. 20% of the data set compounds was held out for the purposes of validation. The training set therefore consisted of $1,158$ hand-annotated compounds, while the *holdout data set* set consisted of 289 hand-annotated compounds. The holdout set was further split in half. The first half, the *test set*, was used to determine when to stop training *Czech Compound Splitter*. The performance of all the approaches presented here was evaluated on the other half, the *validation set*.

Neoclassical constituents, as they do not have an agreed-upon citation form, are labelled with hyphens on both sides, maintaining the original Greek stem as bare as possible. We also systematically interpret these elements as identical whenever their etymology and semantics allow for it, even under circumstances where they undergo formal changes. For instance, the first element of *logografie* 'logography' (*logo-*) and the second element of *sociologie* 'sociology' (*-logie*) are seen to be the same, since they both ultimately descend from the same Greek root. In our data, they are represented by the string *-log-*.

Greek orthography is respected as much as possible, so we respect the distinction between τ and θ, so the first element of *teologie* 'theology' is labeled as *-theo-* (not *-the-*, as that would be ambiguous with the root of *teorie* 'theory'). Zero ablaut forms are preferred as labels of neoclassical compounds, unless this would result in an asyllabic label. Thus, both the first element of *gastronomie* 'gastronomy' and the second element of *melanogaster* (the epithet of the fruitfly *Drosophila melanogaster*) is labeled as *-gastr-*, but the first element of *gonokok* 'gonococcus' and the second element of *mutagen* 'mutagen' are labelled as *-gen-*.

Generation of synthetic data

Because the hand-annotated data set of compounds obtained from DeriNet is too small to reliably train a deep learning model, we simulated various compound formation procedures that take place in Czech. For example, in (64) we see the process of taking a random adjective stripped of its ending and concatenating it with an *-o-* interfix and with another random adjective. The output is usually nonsensical, like in the example, but formally correctly formed.

(64) **Adjective 1** $+$ ***-o-*** $+$ **Adjective 2** $\rightarrow$ **Compound Adjective**
důležitý $+$ ***-o-*** $+$ ***neomylný*** $\rightarrow$ ***důležitoneomylný***
important.ADJ infallible.ADJ important-infallible.ADJ

For the purpose of training *Czech Compound Splitter*, we simulated a number of such compound formation procedures in Python using randomly selected lexemes from DeriNet, creating a data set of about $280,000$ synthetic compounds. The

Method	1st parent accuracy	2nd parent accuracy	Overall accuracy	Overall root accuracy
Baseline	22%	42%	11%	16%
$IML()$	42%	66%	24%	39%
Czech Compound Splitter	61%	66%	54%	61%

Table 3.1: Overall performances the three solutions exhibited.

compound part of the training data set therefore consisted of this synthetic data set combined with all of the hand-annotated compounds apart from the holdout described above.

3.2.2 Experiments

Baseline solution

We first present a naive algorithm only intended as a baseline to help provide context for the performances of the other solutions. This solution assumes the given compound has two parents. It attempts to find an 'o' grapheme in the middle third of the input word. If it finds one, it splits the word on this 'o', creating two subwords. If no 'o' is found, it does the same with 'i'. If no 'i' is found, it simply splits the input in the middle, if the number of graphemes in the graphical word is even, the left subword ends up being the longer one. Between each subword and every word in the lexicon, Levenshtein (1966) distance is calculated, and the word with the smallest distance from the subword is selected. Please refer to Table 3.1 to see its performance.

The IML()-based heuristic algorithm

The second attempt to split compounds is based on a phonological similarity measurement function developed specifically for this purpose. We developed a function that takes two words as input and returns a rational number representing the total degree of phonological similarity between the two words. We then attempted to find pairs of words which, when concatenated, exhibited a low degree of $IML()$ similarity with the compound in question. $IML()$ cannot perform compound identification, because the method already assumes the input word has exactly two parents.

We began by manually defining a phonemic correspondence weight by hand for each possible pair of phonemes in Czech, basing these weights on linguistic intuition. The minimum weight is 0, which is the correspondence weight strictly between a phoneme and itself, and the maximum weight is 1, which is the correspondence weight between a phoneme and a phoneme it never alternates with, like between /a/ and /t/. Note that this relationship is asymmetric by design, because we estimated that, for example, /h/ $\rightarrow$ /z/ is much more common than /z/ $\rightarrow$ /h/. From this, it directly follows that the ordering of the words that are input into the $IML()$ function matters. There are 32 phonemes in the Czech language, so it follows that the total amount of phonemic correspondences equals $32^2 = 1024$.

	/t/	/n/	/r/	/s/	/z/	/ts/	...
/t/	0	0.8	1	0.8	0.9	0.8	...
/n/	0.7	0	0.9	1	1	1	...
/r/	0.7	0.9	0	0.9	1	1	...
/s/	0.6	1	0.7	0	0.2	0.6	...
/z/	0.8	0.3	0.9	0.2	0	1	...
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋱

Table 3.2: *S*ample of the referential matrix of correspondence weights between pairs of Czech phonemes.

This can be described by a square matrix, where each column and row corresponds to one of the Czech phonemes and each element describes the correspondence weight between the Czech phonemes. This is what we call a *correspondence matrix*. Part of the matrix used in this study is shown in Table 3.2. Note that the diagonal is composed entirely of zeroes, and that the matrix is not symmetric with respect to said diagonal, which reflects the asymmetric nature of Czech phoneme alternation described in the previous paragraph.

The $IML()$ similarity measurement function takes two words, transcribes both of them phonologically, and uses the values found in the *correspondence matrix* to build a separate matrix of correspondence weights between every single pair of phonemes from the two input graphical words. The cheapest path through it is found, beginning in the top left corner of the matrix, and ending in the bottom right corner. We used the $A*$ algorithm, an extension of Dijsktra's algorithm, to find the shortest path (Hart et al., 1968).

The lower the output value, the higher the similarity, with $IML(word_1, word_2)$ being equal to 0 if and only if $word_1 = word_2$, because the correspondence weight between a pair of phonemes is zero if and only if the two phonemes in the pair are identical – which is why the diagonal of the *referential matrix* is composed of zeros, and in that the only zeros in the *referential matrix* are located on the diagonal.

Based on this similarity function, we were able to find the pair of words from the lexicon mentioned previously which, when concatenated, exhibited the highest similarity with the compound word in question. The algorithm therefore requires the compound in question and a lexicon to find its parents in. A visual demonstration of the idea behind the algorithm with the word *černomodrý* 'black and blue' and a toy lexicon can be viewed in Table 3.3. The table shows the outputs of the $IML(\textit{compound}, \textit{word}_1 + \textit{word}_2)$ calculations for each word pair from the $\{$*černý*, *červený*, *modrý*$\}$ {'black', 'red', 'blue'} lexicon. The algorithm generates all pairs of lexemes from a given lexicon, concatenates them and calculates $IML()$ for each pair. It then (correctly in this case) selects the pair with the smallest value. The problem is that the size of our lexicon ultimately exceeded $800,000$ lexemes, meaning that every time a compound is split, over $800,000^2 = 6.4 \times 10^{11}$ interlexical matrices need to be built and run through the $A*$ pathfinding algorithm.

A heuristic filter was therefore added. For this purpose, a variant of the $IML()$ function, the $IML_{\text{sub}}()$ function, was defined. The two functions are similar with two key differences. First, in the case of the $IML_{\text{sub}}()$, the cheapest path does not have to reach the bottom right corner of the matrix. Instead, the path's to-

IML(*černomodrý*,*černý* + *černý*) $= 5.8$
IML(*černomodrý*,*černý* + *červený*) $= 5.0$
IML(***černomodrý***,***černý + modrý***)$=$ **0.6**
IML(*černomodrý*,*červený* + *černý*) $= 5.7$
IML(*černomodrý*,*červený* + *zelený*) $= 6.9$
IML(*černomodrý*,*červený* + *modrý*) $= 2.6$
IML(*černomodrý*,*modrý* + *černý*) $= 9.6$
IML(*černomodrý*,*modrý* + *zelený*) $= 9.8$
IML(*černomodrý*,*modrý* + *modrý*) $= 10.6$

Table 3.3: Sample of the algorithm's functioning, without the heuristic filter.

tal cost is calculated whenever it reaches either the right or bottom edge of the *interlexical matrix*. $IML_{\text{sub}}(word_1, word_2)$ returns the *degree* to which $word_2$ is a fuzzy substring of $word_1$, with respect to their phonological similarity. Second, the pathfinding algorithm used in $IML_{\text{sub}}()$ is not $A*$, but a best-first solution. This makes $IML_{\text{sub}}()$ significantly faster than $IML()$, because the whole *interlexical matrix* need not be constructed beforehand. Only word pairs $(lexeme_1, lexeme_2)$ which satisfied the following conditions were selected:

1. $First2Chars(lexeme_1) = First2Chars(compound)$,

2. $CountSyl(lexeme_1 + lexeme_2) \geq CountSyl(compound)$,

3. $IML_{\text{sub}}(lexeme_1) \leq 2.2 \quad AND \quad IML_{\text{sub}}(lexeme_2) \leq 2.2$,

where $First2Chars()$ is a function which returns the first two graphical characters of a given graphical word, $CountSyl()$ counts the syllables of the given graphical word (assuming it is a Czech word) and $compound$ is the input compound being split.

This pair of words then constituted the predicted parents. The performance of this method in compound splitting can be found in Table 3.1. The application of the algorithm seems to be much less practical than that of *Czech Compound Splitter*, because it takes about ten to fifteen minutes to split a single compound on a single processor given a lexicon of our size, despite the fact that the algorithm's asymptotic time complexity (even without the heuristic) is $O(n) = n^2$, where n refers to the size of the lexicon. The matrix building step takes $|word1| \times |word2|$ correspondence matrix lookup operations, but because the step occurs exactly once for each parent-candidate pair, it constitutes a constant, and is therefore by convention omitted when assessing asymptotic time complexity. It is additionally of interest that the root accuracy of this method was higher by 11 percentage points than its raw accuracy. Error analysis revealed that this increase primarily caused a common error where a substring of a compound is homonymous to a noun derived from an adjective, while that adjective is the parent. For example, *bíločerný* 'black and white' is split into the noun *bílo* 'whiteness' and the adjective *černý* 'black', while two adjectives (*bílý* 'white' and *černý* 'black') are the correct parents.

Czech Compound Splitter

Because the performance and practicality of the $IML()$-based heuristic algorithm was deemed unsatisfactory, a neural compound splitting tool we named *Czech Compound Splitter* was created. It decides if an graphical word is a compound and if so, it returns its predicted parent words, all in one step. If the graphical word is identified as a compound, it returns its parents separated by spaces. The estimated number of parents is thus the number of spaces in the output $+1$, and the status of a compound is determined if this number is greater than 1.

The tool was created by using the Marian machine translation framework developed by Microsoft (Junczys-Dowmunt et al., 2018) to build a model and train it. This was done by feeding the model a parallel corpus of input and output data, where the model is trained to take an element of the input data, which was a Czech word, and the output was either the single derivational parent of that word if it was a non-compound, or all of the parents of that word separated by spaces if it was a compound. For example, *Czech Compound Splitter* was trained to return *kov* 'metal' upon being given the graphical word *kovový* 'made of metal', and to return *uhlík vodík* 'carbon hydrogen' upon being given the graphical word *uhlovodík* 'carbohydrate'. The non-compounds and their parents were taken from DeriNet.

The total training data set for *Czech Compound Splitter* consisted of:

- $1,158$ genuine compounds, with their splittings
- $280,000$ synthetic compounds, with their splittings
- the near entirety of DeriNet's non-compounds, with their derivational parents

The rest of DeriNet's non-compounds, totalling 144 lexemes, was held-out as a small collection of counter-examples in order to test the performance of *Czech Compound Splitter* in compound identification.

An inherent weakness of MLP's is the fact that their input must be represented as a fixed-length vector. However, compound words are not fixed-length, be they represented as sequences of graphemes, phonemes, syllables, morphemes, or otherwise, but there fortunately exists a neural architecture which can handle sequential data.

The simplest way to handle such data would be to initialize a feedforward layer for input of size $2 \times n$[4] and output of size n, called the RNN *cell*. Then, the first grapheme in the given input compound would be concatenated with a zero vector, resulting in a vector of size $2 \times n$, and fed back into the model. In the next step, the output of the MLP is concatenated with the second vector, fed back into the model, and then the third vector in the sequence is concatenated with the output and fed back in, until the sequence has run out. The last output vector then represents

[4]Where n is the *embedding dimension* – each grapheme is embedded into a vector space either by a so-called one-hot vector, which is a vector of zeros except the entry corresponding to the character's index in a dictionary, which is set to 1. n is therefore the size of the collection of all graphemes in the training data, plus three special tokens – <START>, <END>, and <UNKNOWN>, which is used when an unknown grapheme is encountered.

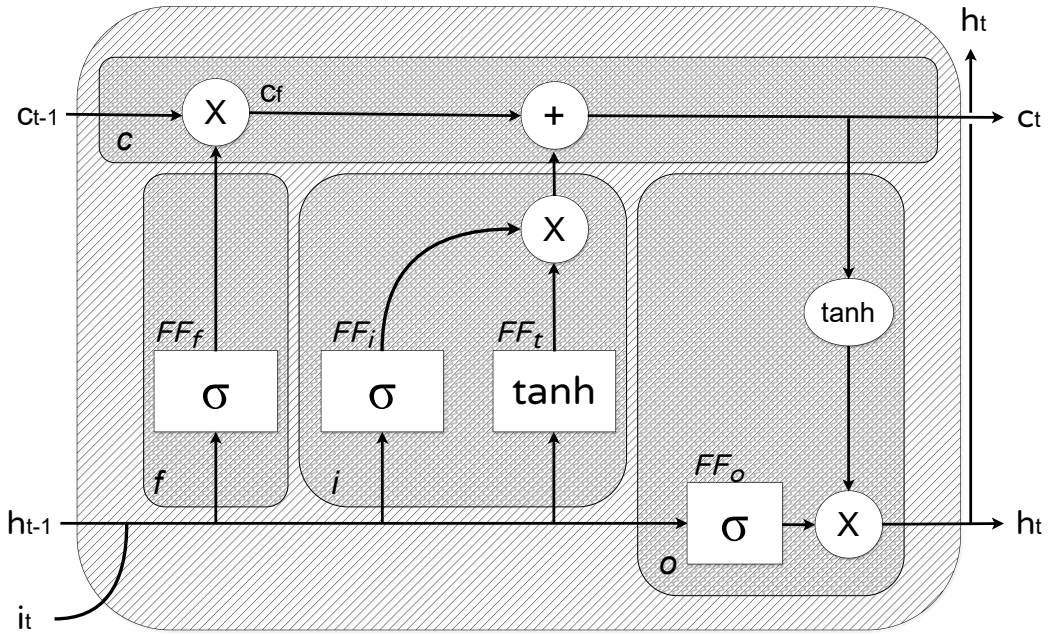


Figure 3.1: LSTM Cell.

the whole sequence, and can be fed into e.g. a feedforward layer with the *softmax* output function to produce a recurrent classifier model.

The problem with the previous setup is that the longer the sequence, the higher the probability that some important information from the previous step does not get passed along to the next. Obviously, the problem compounds itself as the sequence iterates along, making it highly probable that a given model has forgotten e.g. information about the prefix of a given word once it gets to its last grapheme.

To solve this, in 1997 Hochreiter and Schmidhuber introduced the Long Short-Term Memory (LSTM) model architecture, which was used in Czech Compound Splitter. The high-level principle of the LSTM cell is that it combines three FF layers with the *sigmoid* activation function (FF_f, FF_i, FF_o), one FF layer with the *hyperbolic tangent* (tanh) activation function (FF_t), and the operations of Hadamard (elementwise) vector product, elementwise vector sum, and the $tanh$ function.

The sigmoid function:

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

The hyperbolic tangens function:

$$\tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}$$

The structure of the cell is illustrated in Figure 3.1, where white rounded rectangles represent singular operations and white angled rectangles represent FF layers. The cell is composed of four blocks:

1. The *forget gate* (*f*), which takes the output from the previous time-step (h_{t-1}), concatenates it with the current time-step's input (i_t), passes it through its

own sigmoid-activated FF layer (FF_f), and multiplies it with the previous step's carousel vector (c_{t-1}), resulting in c_f;

2. The *input gate* (*i*), which then takes the concatenation of h_{t-1} and i_t, passes it through its own sigmoid-activated FF layer FF_i and the hyperbolic tangent-activate FF layer FF_t in parallel, multiplying the results, and adds it element-wise to the carousel vector $\tilde{c_t}$, resulting in c_t;

3. The *output gate* (*o*), which passes the concatenation of h_{t-1} and i_t through FF_o, and multiplies the result with $\tanh(c_t)$, resulting in h_t;

4. The *constant error carousel* (c). In addition to feeding back the cell's previous output at each time-step, the LSTM cell emits (c_t) and feeds back (c_{t-1}) a vector of equal size as h.

c_t and h_t are then used in the next iteration as c_{t-1} and h_{t-1}. In the first iteration, both of these are set to zero.

The purpose of the constant error carousel is to help prevent the exploding/vanishing of errors in the gradient during training. The problem with recurrent or very deep neural networks is that the repeated application of a squashing function such as sigmoid or hyperbolic tangent leads to the constant diminishing of the input. This is a problem during backpropagation, because during training as the gradient flows backward through time, it gets squashed almost to zero. The usage of a non-squashing activation function, on the other hand, may result in the gradient exploding, which is also a problem. The LSTM cell architecture uses squashing functions, but the constant error carousel maintains a consistent flow of information through time, forwards and backwards, unbroken by non-linearities, which mitigates this issue.

In compound splitting of Czech, the required output sequence often needs to be longer than the input sequence, as is the case with the compound *psovod* (6 characters), whose correct splitting is *pes vést* (8 characters). The sequence-to-sequence (*seq2seq* or *s2s*) architecture is designed specifically to transform a sequence of (textual) data into another sequence of (textual) data without any assumption regarding the relative lengths of either one (= it does not assume the input sequence must be longer or shorter than the input), which makes this approach suitable not only for compound splitting, but also for tasks such as machine translation (Sutskever et al., 2014; Cho et al., 2014) or parent retrieval (Chapter 3.4). The input sequence is fed into an LSTM layer, which is an LSTM cell set up so that it iterates over any sequence fed into it. The last output h_t (where t is the length of the input sequence) can be understood as representing the whole sequence. This layer is called the *encoder*, since it encodes the input sequence into a fixed-length vector known as the context vector.

The decoder (using *greedy autoregressive decoding* in this case) is another LSTM layer, which in the first time step takes as its input the context vector concatenated with a vector representing a special <START> token. The output vector of this step is then run through a classifier FF layer, in which each of the classes corresponds to one item in the vocabulary (which in the case of e.g. English parent retrieval is simply the English alphabet, plus the <START> and <START> special tokens). The item thus selected from the vocabulary is then embedded into the same vector

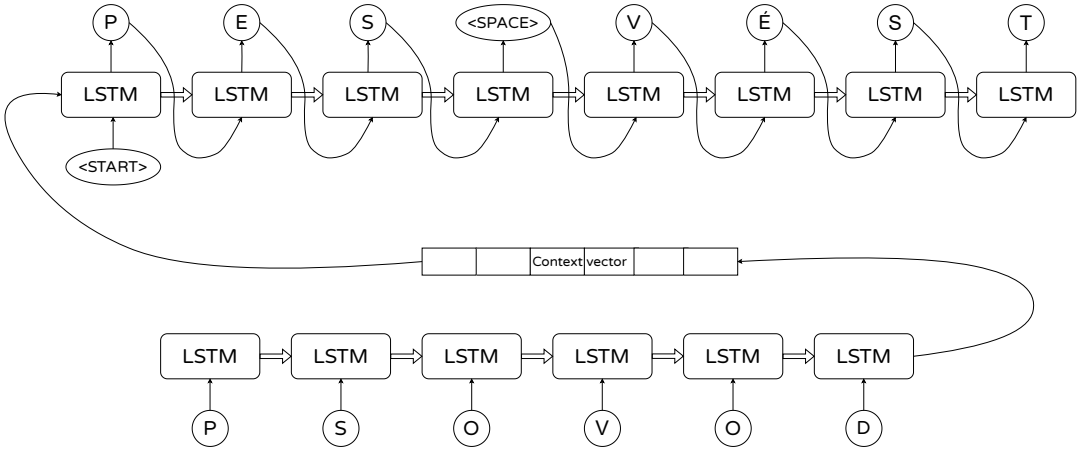


Figure 3.2: LSTM-based seq2seq example compound splitter.

space as in the encoder and fed back into the LSTM cell as the input for the next iteration. The LSTM cell iterates until the special <START> token is emitted.

The procedure is illustrated in Figure 3.2 on the Czech compound *psovod*. The Marian framework also offers a s2s model enriched with *attention*, which is a way of supplying the decoder of the model with information about the whole input sequence at every step. Specifically, attention enriches each element of a data sequence A with information about data sequence B by taking one element from sequence A, scoring the relevance of those two elements in some way, and then adds the thus-weighted sum to that element. The attention mechanism does this with every element of A, and outputs the result. In cases where $A = B$, we refer to a specific type of attention called *self-attention*.

Bahdanau attention (Bahdanau et al., 2014), also known as additive attention, is the type of attention used in *Czech Compound Splitter*. It utilizes a FF layer of input shape $[2 \times n]$ (where n is the dimensionality of the items in sequence A) that returns a single score variable to score the pairs of items from A and B. It simply takes a pair from A and B, concatenates it together, and thus obtains a sequence of scores of length B. It runs the score sequence through the softmax function, forcing the sequence to sum to 1, and then calculates the weighted sum of sequence B, using the softmaxed score sequence as the weights, and adds it to the relevant item from sequence A.

To sum up, the architecture of *CCS* is a s2s LSTM model enriched with Bahdanau attention.

3.2.3 Tool performance

In compound identification, *Czech Compound Splitter* achieved an accuracy of 92% and an F1-score of 91%. Its performance in compound splitting can be found in Table 3.1. We see that root accuracy is just barely higher than accuracy. Error analysis reveals that this was due to the fact that a large proportion of the mistakes *Czech Compound Splitter* made because it often did not recognize the input as a compound. Similarly, it frequently returned a nonsensical string that is not a Czech word; see a sample of errors in Table 3.4.

It is worth noting that *Czech Compound Splitter* made only a single false positive

Compound	English translation	*CCS* splitting	Correct splitting
dlouhohořící	'long-burning'	*dlouhohořící*	*dlouho* + *hořící*
CCS returns the original string, performing no splitting.			
osmiramenný	'eight-armed'	*osm* + *ramenný*	*osm* + *rameno*
CCS returns a non-existing derivative of an existing word.			
petrogeneze	'petrogenesis'	*-petro-* + *geneze*	*-petr-* + *geneze*
CCS includes the interfix in one of the parents.			

Table 3.4: A sample of the errors *Czech Compound Splitter* (CCS) typically makes.

error, meaning that it almost never labelled a non-compound as a compound. This suggests that it primarily recognizes compound status by detecting lexical-seeming substructures, as opposed to focusing on surface-level criteria like character length or the presence of an *-o-* interfix. *Czech Compound Splitter* run on a single GPU takes about 0.2 seconds to perform a single identification and splitting. The entire compiled model is about 300 MB in size and can be compiled to run on CPUs.

3.3 *Word Formation Analyzer for Czech*

Having developed a compound splitter for Czech, we decided to generalize the task of compound splitting to include derivation. This was in large part motivated by the discovery that the underlying model of *Czech Compound Splitter* benefitted from being trained on derivatives from DeriNet as well, albeit only as examples of non-compounds. The natural next step was therefore to expand the binary task of compound identification to the ternary task of word formation classification, and to build a tool that can not only return the parents of compounds, but also of derivatives. After all, a native speaker of Czech, when given a word, generally finds it easy to determine which Czech word or words it comes from, or if any such ancestor word exists. In contrast, there is no trivial automatic procedure that can do the same.

Research on this topic had so far been mostly limited to creating static data resources, similar in principle to dictionaries, capturing Czech words with links to their respective ancestors. The problem is that speakers and writers constantly coin new words to suit their communicative requirements, which means that no static data resource can capture the entirety of Czech word formation at any given point in time. This creates the need for a procedural tool capable of handling any word, regardless if it is a long-established word or a new coinage, irrespective of whether or not it is a compound.

In this chapter, we therefore present *Word Formation Analyzer for Czech* (*WFA.ces*), a tool based around an ensemble of three sequence-to-sequence deep-learning models. The tool takes as its input a string of characters assumed to be a Czech lexeme in its dictionary form (lemma), and returns a predicted sequence of one or more words the input string was originally motivated by. Since the tool receives nothing but an isolated string as its input, the procedure is entirely based on the written form of the input. *WFA.ces* can perform two tasks:

1. *Parent retrieval*

   WFA.ces predicts which word or words the input lemma is motivated by. It does this by generating a list of candidate sequences of parent words, and returning the best sequence based on a particular reranking procedure of the user's choice. This task is similar to that of *stemming*, but with a stronger focus on linguistic adequacy. Parent retrieval does *not* handle inflection, so inputting *happiest* into *WFA.ces* may in practice result in unexpected behavior.

2. *Word formation classification*

   WFA.ces classifies the input lexeme into one of the classes *compound*, *derivative*, or *unmotivated*. It returns the class *compound* if there are two or more words in the output (*hlavonožec* ('cephalopod') ← *hlava* ('head') + *noha* ('leg')); the class *derivative* if there is one word AND it differs from the input (*hlavička* ('little_head') ← *hlava* ('head')); and finally, if there is one word AND it is identical to the input, it returns the class *unmotivated* (*hlava* ('head') ← *hlava* ('head')).

3.3.1 Data

The golden data was acquired from DeriNet 2.0 (Vidra et al., 2019). From there, all lexemes that fulfill all of the following requirements at the same time were taken and designated as *derivative*:

- have a single parent,
- are attested in the SYN2015 corpus of Czech (Křen et al., 2016),
- and are not labeled as either *unmotivated* or *compound*,

Then they were paired with their respective DeriNet parent, alongside the class label for *derivative*.

Similarly, all lexemes that fulfilled the following properties were taken and designated as *unmotivated*:

- have no parents,
- are attested in the SYN2015 corpus of Czech,
- and are labeled as *unmotivated*,

The compounds used were compounds from DeriNet with both parents linked. In addition, 285 compounds were hand-annotated specifically as part of creating *WFA.ces*. This data was then compiled into a dataframe of three columns – the first was the lemmas of the lexical items, the second was the parent(s) of these items, and the third contained the respective word class labels.

The data was split into a train set (60%), a test set (20%) and a validation set (20%) according to the *compound* class, as it was the class with the least items. The *unmotivated* and *derivative* classes were split such that there was the same number

Model type	Dropout	Direction	Training iterations
default	0.2	left to right	100,000
transformer	0.5	left to right	900,000
s2s	0	right to left	30,000

Table 3.5: Description of the configurations in the model ensemble used in *Word Formation Analyzer for Czech*

of items from each of the classes in both the test and validation sets. The rest of the *derivative* items and *unmotivated* items were added into the train set.

Some errors in class labelling were manually found in the test and validation sets, and were appropriately corrected, which resulted in a class imbalance, albeit very slight. The exact composition of the resulting train, test, and validation sets can be viewed in Table 3.6.

For the purposes of evaluating parent retrieval, we use accuracy, which we define as the proportion of cases wherein *all* parents were correctly predicted by *WFA.ces*.[5] In the case of neoclassical compounds, we strictly require the predicted constituents to be correctly hyphenated, as in (65), otherwise the prediction counts as incorrect, cf. (66) and (67).

(65) ***krypt|-o-|fašista*** ← ***-krypt-*** ***fašista*** ✓
cryptofascist.NOUN -crypt-.NEOCON fascist.NOUN

(66) ***krypt|-o-|fašista*** ← ***krypt-*** ***fašista*** ✗
cryptofascist.NOUN crypt-.NEOCON fascist.NOUN

(67) ***krypt|-o-|fašista*** ← ***krypt*** ***fašista*** ✗
cryptofascist.NOUN crypt.NEOCON fascist.NOUN

For the purposes of evaluating *word formation classification*, we rely on convention, using balanced accuracy (balanced so as to compensate for the slightly imbalanced train and validation sets) to assess the model's performance across all three classes; and precision, recall, and F1-score metrics, to evaluate the tool for each word class separately.

For about 38% of the hand-annotated compounds in our dataset, there was ambiguity as to which parents they should be linked to. For instance, *rybolov* 'fishery' may be considered to be either composed of the noun *ryba* 'fish' and the noun *lov* 'hunt', or it alternatively may be analysed as an output of compounding and conversion with the noun *ryba* 'fish' and the verb *lovit* 'to hunt' as inputs (cf. (68a), (68b)). For the purposes of evaluation, both were considered to be correct retrievals. This decision is technical rather than linguistic, and is not supposed to reflect any theoretical preference or view on directionality of conversion and other related issues.

(68) a. ***ryb|-o-|olov*** ← ***ryba*** ***lov*** ✓
fishery.N fish.N hunt.N

(69) b. ***ryb|-o-|lov*** ← ***ryba*** ***lovit*** ✓
fishery.N fish.N hunt.V

[5] Parent retrieval accuracy of unmotivated words is equal to the precision of word formation classification, if we consider *unmotivated* to be the positive class.

Class	Training	Testing	Validation
Compounds	1,164	284	280
Synth. compounds	280,000	0	0
Derivatives	148,921	285	287
Unmotivated	4,911	284	288
Total	435,280	853	855

Table 3.6: The number of lexemes in each formation class, alongside their respective parents, that composed the datasets used to train, develop, and test *Word Formation Analyzer for Czech*

3.3.2 Experiments

The core of *WFA.ces* was, like its predecessor, built using the *Marian* framework developed by Junczys-Dowmunt et al. (2018), utilizing an ensemble of three models described in Table 3.5. All of the models in the ensemble were then trained on the dataset described in Table 3.6 with layer regularization. The model ensemble was trained to take a lexeme from the train set as its input (left-hand side of the arrow in the examples in the previous section) and return its corresponding parent(s) as output (right-hand side of the arrow), separated by spaces if there is more than one parent.

Model ensemble training and tuning

The model ensemble contained Marian's default model (GRU encoder-decoder; Cho et al. 2014), an s2s model similar to the one used in Czech Compound Splitter (but with every input fed into it in reverse), and a Transformer model akin to the one described in (Vaswani et al., 2017).

The Transformer architecture was introduced in 2017 by Vaswani et al. and revolutionized the field. The Transformer, like the RNN and LSTM, is designed to operate on sequences of data of any length. However, unlike in the case of recurrent models, the Transformer operates on the sequential data *in parallel*. Information about the ordering of the sequence is not maintained by the model's architecture, but is instead encoded into the sequence itself using so-called positional encoding.

In positional encoding, for each vector in the sequence, a positional vector is generated and then added, giving the model information about the position of each sequence element. This has two main advantages over the recurrent setup.

1. The model is more parallelizable, since it is unlike in the case of Vanilla RNNs and LSTMs not necessary to wait until item 1 from sequence A is processed to start processing item number 2.

2. The positional encoding makes it so that it is not difficult for the model to find long-term dependencies. In the RNN/LSTM/GRU setup, the model has to learn to hold onto important information for six timesteps between e.g. item 1 and item 7. In contrast, with the positional encoding setup, it is no more difficult to learn a dependency between item 1 and item 7 than between item 1 and item 2.

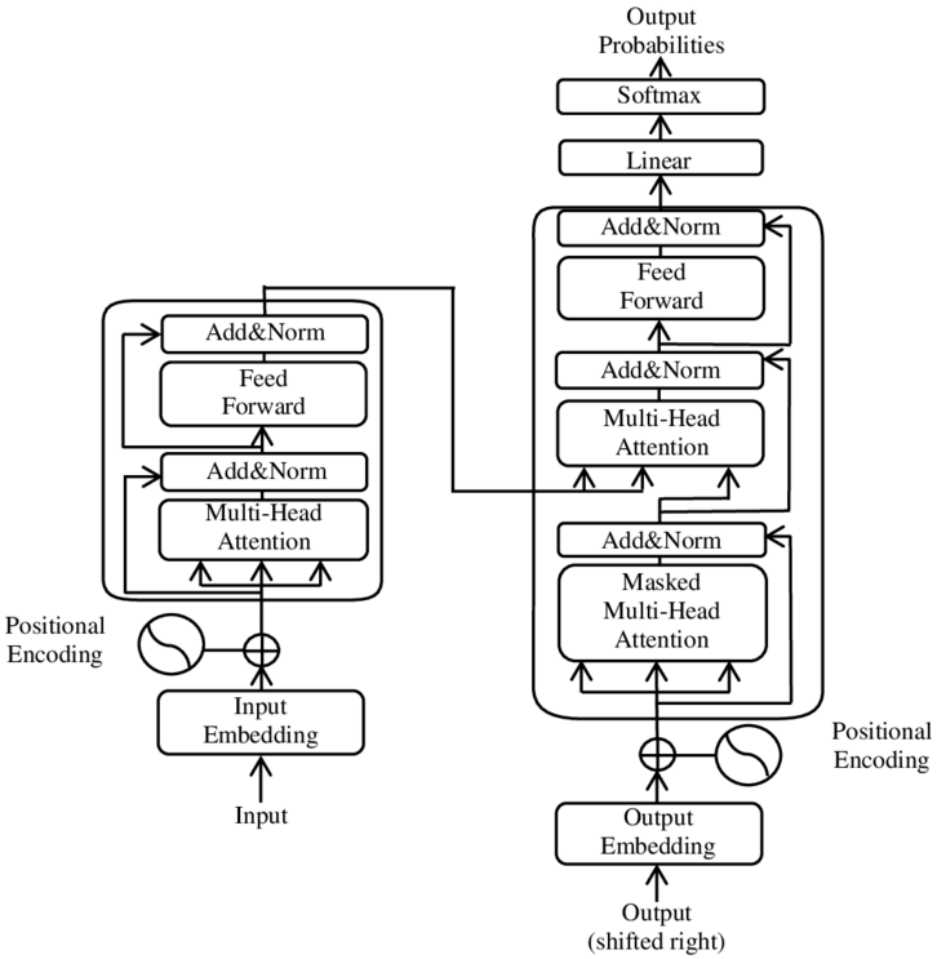


Figure 3.3: The Transformer. By Yuening Jia, CC BY-SA 3.0, via Wikipedia.

In the Vaswani et al. setup, the positional encoding for each vector in the input sequence is calculated by the equation

$$PE(pos, 2i) = sin\left(\frac{pos}{10000^{2i/d_{model}}}\right)$$

for even indices in the vector, and the equation

$$PE(pos, 2i+1) = cos\left(\frac{pos}{10000^{2i/d_{model}}}\right)$$

for odd indices. Since these have the same size as the input vectors, they can be summed.

Next, the positional embedding-enriched sequence is fed through a stack of Transformer blocks. The input of a Transformer block is first run through so-called multi-head scaled dot-product attention.

In scaled dot-product attention (Vaswani et al., 2017), three trainable weight matrices are used – W_{Query}, W_{Key}, and W_{Value}. Then, three linear transformations take place.

1. W_{Query} is used to transform A into a matrix Q of shape $[len(A) \times d_Q]$,
2. W_{Key} is used to transform B into a matrix K of shape $[len(B) \times d_K]$,

3. and W_{Value} is used to transform B into a matrix V of shape $[len(B) \times d_V]$,

where $len(X)$ refers to the length of sequence X and d_Y refers to the dimensionality of item Y. The attention is then performed by the equation

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_K}}\right)V,$$

where the softmax function is performed per row. The matrix QK^T is divided by the square root of the key dimensionality, because the magnitude of some of the matrix values increases with dimensionality. This would lead to very small values after the application of the softmax function, leading to small gradients and by extension slow training. In multi-head attention, multiple attention blocks with separate trainable weight matrices are run in parallel, summed, and then multiplied by a separate trainable weight matrix W_o before being output.

The output from the multi-head attention is then added back to the input of the block (a skip connection), and run through a Layer normalization block (Ba et al., 2016). Then, each vector in the sequence is separately run alongside a skip connection through a feedforward layer and normalized again before being returned as output. Since these blocks return a sequence of the same shape that they accept as input, they can be stacked any number of times.

While a positional encoder followed by simple Transformer block stack ending in a sum-across-sequence layer and an output layer can be useful for tasks like classification, the Transformer is typically utilized in an encoder-decoder fashion akin to the s2s architecture described in the previous Section. This arrangement, which can be viewed in Figure 3.3, is useful for sequence-generation tasks, such as natural language generation, machine translation, and compound splitting or parent retrieval.

The hyperparameters of the model ensemble, such as the dropout rate, number of training iterations, amd directionality of the specific model were fine-tuned manually on the development set.

Tool functioning

WFA.ces works by feeding the *Marian* model ensemble an input lexeme L in its lemma form and generating a list of possible parent sequences of size n, where n is a natural number chosen by the user. The parent sequences in the list are ordered by their probabilities as predicted by the model ensemble. It then uses simple procedures to find the best candidate in this list to produce the desired outcome for each of the two tasks.

- *Parent retrieval. WFA.ces* takes the list of possible parent sequences and uses one of the following reranking procedures, as chosen by the user, to select the best one:
  - *First best*: *WFA.ces* simply returns the first parent sequence in the list.
  - *Lexicon*: *WFA.ces* uses a provided lexicon to select the first parent sequence in the parent sequence list whose elements are all attestable in that lexicon. If none such sequence can be found in the list, it uses *First best*.

Lexeme class	Reranking method			
	Oracle	First best	Lexicon	Frequency
Compound	70%	56%	55%	57%
Derivative	87%	69%	75%	59%
Unmotivated	91%	71%	84%	67%
Total	83%	65%	71%	61%

Table 3.7: The accuracy scores of *Word Formation Analyzer for Czech* in the task of parent retrieval, broken up for each word formation class, as measured on the validation set for $n = 4$.

  - *Frequency*: *WFA.ces* uses a list of relative corpus frequencies[6] and assigns each element of each sequence in the list of possible parent sequences. It then selects the parent sequence with the smallest sum of squared frequencies.
  - *Oracle*: This method is only available if the ground truth is already known, and as such, it is only useful for the purpose of evaluation of the other reranking methods. It returns the correct result, if present in the sequence list.

- *Word formation classification*. *WFA.ces* takes the list of possible parent sequences, and:
  1. Checks if any of them contains a space character.
  2. If yes, it classifies L as a *compound*.
  3. If not, it checks whether or not any of the parent sequences are equal to L.
     (a) If yes, it classifies L as an *unmotivated lexeme*.
     (b) If not, it classifies L as a *derivative*.

From this, it follows that when using *WFA.ces* as a *word formation classification* tool, one can consider n to be a user-defined classification threshold: the lower it is, the more *WFA.ces* tends to classify lexemes as *compounds*; the higher it is, the more *WFA.ces* tends to classify words as either *unmotivated* or *derivative*.

3.3.3 Performance evaluation and error analysis

The performance of *WFA.ces* in parent retrieval can be viewed in Table 3.7. The best reranking method in total is *Lexicon*, though of interest is also *Frequency*, due to its performance in the retrieval of the parents of compounds. This is important, because a user of the tool might decide that the retrieval of compositional parents is more important than the retrieval of derivational parents for the user's purposes, and may select the reranking procedure appropriately. Similarly, a user might decide to use the *First best* method for applications where a reliable lexicon of

[6] Acquired from DeriNet 2.0 for the purposes of this Section.

Positive lexeme class	Classification metric		
	Precision	Recall	F1
Compound	96%	92%	94%
Derivative	74%	97%	84%
Unmotivated	96%	70%	81%

Table 3.8: The Precision, Recall and F1 scores achieved by *Word Formation Analyzer for Czech* for each word formation class.

potential parent words might not be available, such as for the analysis of technical or medical vocabulary, despite the fact that the method exhibits the lowest performance in general performance on our validation set.

In word formation classification, the tool additionally achieved a balanced accuracy of 87% across all three word formation classes. Its performance in this task with regards to each class can be viewed in Table 3.8, wherein each line corresponds to the given class being considered positive and all the others being considered negative for the purposes of the metrics listed in each column. The performance in the classification of compounds is especially promising, suggesting that Czech compounds carry a very distinctive formal fingerprint.

Error analysis confirms that each reranking method presents its own set of strengths and weaknesses. The weakness of the *First best* method is that it often returns strings which are not Czech lemmas (cf. the first line in Table 3.9). The *Lexicon* method partially solves the problem of nonsensical string outputs, but introduces other problems. For example, it often assumes that neoclassical compounds are unmotivated, because even when a correct splitting comes up in the predicted sequence list, one or more of its constituents might not be present in the lexicon. *WFA.ces* therefore searches for other candidates in the list, wherein the entire neoclassical compound often appears, and is thus returned as the only candidate attestable in the given lexicon (cf. the second line in Table 3.9). The shortcoming of the *Frequency* reranking, on the other hand, is that it returns highly frequent words even when they are a formally dissimilar candidate from the input (ex. third line in Table 3.9 – *malý* 'small'). Additionally, the tool has no way of leveraging semantics to its advantage, leading it to analyze *siný* 'light blue' as a derivative of *sít* 'to sow' (the penultimate line of Table 3.9). Some errors were not specific to any particular reranking method. For example, many adverbs in Czech are derived from adjectives. The single most common error in derivational retrieval was in the analysis of such adverbs – instead of retrieving the motivating adjective, *WFA.ces* retrieved the adjective's parent, essentially skipping one derivational step (cf. the last line of Table 3.9).

In parent retrieval, *WFA.ces* outperforms *Czech Compound Splitter*. Parent retrieval, restricted to compounds, is equivalent to compound splitting; *WFA.ces* exhibits an accuracy of 57% in this task, whereas *Czech Compound Splitter* scores three percentage points less.

The result of *WFA.ces* in word formation classification is somewhat comparable to *Czech Compound Splitters*'s performance of 92% in *compound identification*, but the difference between the two is that the former discriminates between three classes (and thus has a random hit baseline of ca. 33.3%), while the latter discriminates

Reranking	Input word	Predicted	Correct
First best	*plnovous* 'full_beard'	**plnový*	*plný vous* 'full beard'
Lexicon	*ombrograf* 'ombrograph'	*ombrograf*	*-ombr- -graf-*
Frequency	*malamut* 'Malamute'	*malý* 'small'	*malamut* 'Malamute'
All	*siný* 'light_blue'	*sít* 'sow (verb)'	*siný* 'light_blue'
All	*ženský* 'womanly (adv)'	žena 'woman'	ženský 'womanly'

Table 3.9: A sample of the errors of *WFA.ces* under various reranking methods.

between two classes (having a random hit baseline of 50%). Since the difference between the accuracy scores is five percentage points, but the difference between the baselines is ca. 17 percentage points, we can conclude that *WFA.ces* represents an improvement over *Czech Compound Splitter*. Another feature which sets *WFA.ces* apart in this regard is its classification threshold, which *Czech Compound Splitter* notably lacks, and strongly prefers to identify words as non-compounds.

3.4 *PaReNT* (Parent Retrieval Neural Tool)

The successor of *WFA.ces* is *PaReNT* (Parent Retrieval Neural Tool), a deep-learning-based multilingual tool performing *parent retrieval* and *word formation classification* in English, German, Dutch, Spanish, French, Russian, and Czech. *Parent retrieval* refers to determining the lexeme or lexemes the input lexeme was based on (e.g. *darkness* is traced back to *dark*; *waterfall* decomposes into *water* and *fall*). Additionally, *PaReNT* performs *word formation classification*, which determines the input lexeme as a *compound* (e.g. *proofread*), a derivative (e.g. *deescalate*) or as an unmotivated word (e.g. *dog*). From a computational perspective, distinguishing between these may not be trivial, as exemplified in the compound ex. (70), the derivative ex. (71) and the unmotivated ex. (72). Data is aggregated from a range of word-formation resources, as well as Wiktionary, to train and test the tool. The tool is based on a custom-architecture hybrid transformer block-enriched sequence-to-sequence neural network utilizing both a character-based and semantic representation of the input lexemes, with two output modules – one decoder-based dedicated to parent retrieval, and one classifier-based for word formation classification. *PaReNT* achieves a mean accuracy of 0.62 in parent retrieval and a mean balanced accuracy of 0.74 in word formation classification.

(70) English

backache $\rightarrow$ ***back ache***

(71) English

backness $\rightarrow$ ***back***

(72) English

baklava $\rightarrow \lambda$

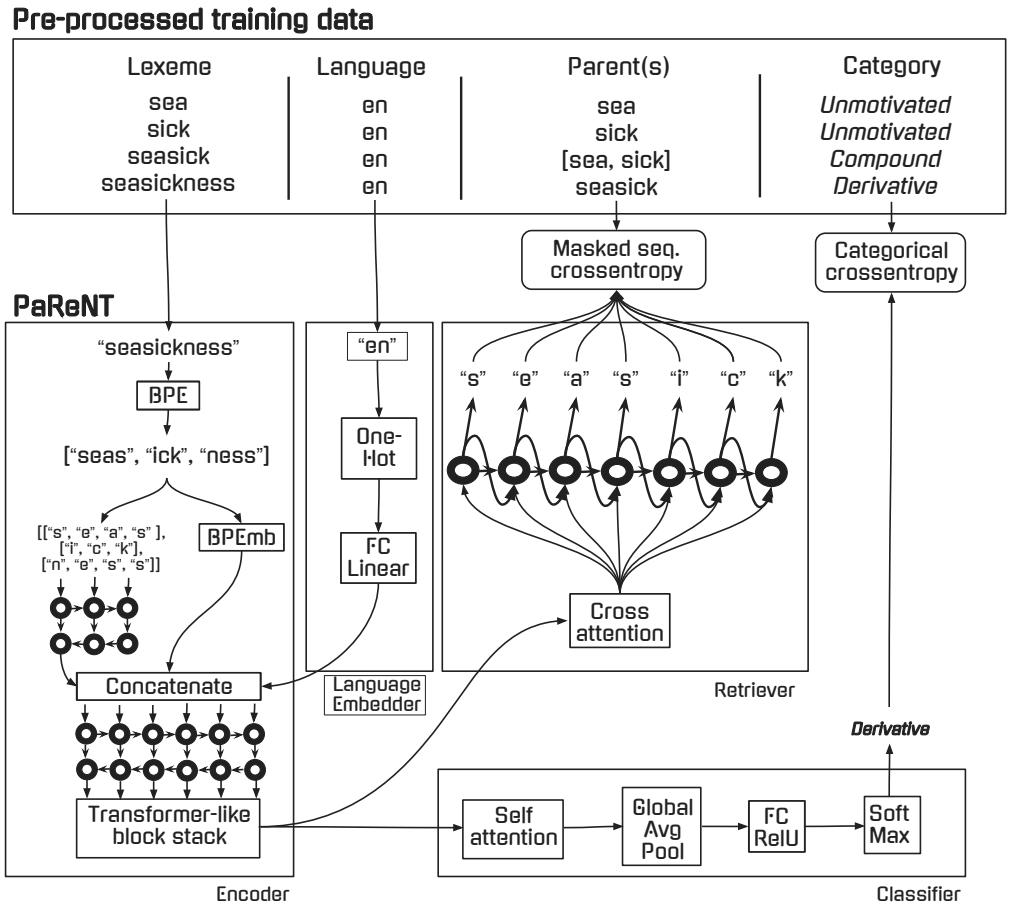


Figure 3.4: Visual schema of the process of training PaReNT.

3.4.1 Data

The high-level overview of *PaReNT*'s functioning is simple – all available data sources are utilized to compile a flat list of lexemes in their lemma form and a corresponding list of parent sequences:

```
sea             [sea]
seasick         [sea sick]
seasickness     [seasick]
```

The list pair is then split into training, development, and validation sets. A deep neural model is then trained on the training dataset to take a triple consisting of a special language token, the input lexeme as a sequence of character sequences, and the input lexeme as a semantic embedding, and to retrieve the sequence of its parents separated by spaces and classify the lexeme into one of: *compound*, *derivative*, *unmotivated*. Its hyperparameters are tuned using the development set, and its performance is calculated on the validation set. This process can be viewed in the third part of Figure 3.4.

For the purposes of this Section, we adjust two established linguistic concepts to suit our needs. First, we understand *compounds* as lexical units that regularly appear as graphical words, as opposed to compounds written with a space, which we

consider syntactic constructs and therefore out-of-scope. This criterion (as Haspelmath 2017 argues) is an artifact of a combination of the Western linguistic tradition and often arbitrary orthographical conventions, but since the parent retrieval of such compounds is identical to tokenization, it was deemed uninteresting.

The data sources used in this study were selected so that in summary, they contained at the very least several hundred examples of each of the given categories for each of the languages in scope. A concise description of the sizes of the data sources used in this study can be viewed in Table 3.10, alongside their respective citations. Additionally, we crawled Wiktionary to enrich our dataset with French and Spanish compounds, because MorphoLex (2020) offered only several hundred compounds for each language. The data is split into three subsets – training, development, and validation, at a $60/20/20$ ratio. The split was done according to what we call *lexicographical blocks*. This means that lexemes belonging to the same DAG had to end up in the same subset. As a specific example, each of the German words *Arbeit*, *arbeitslos* and *Fabrikarbeit* were placed in the development subset. In total, the training set contained $543,066$ words, the development set $180,293$ words and the validation set $179,725$ words. Splitting along lexicographical blocks ensures that the model's ability to retrieve and classify lexemes bearing unseen roots is evaluated. If information about DAGs is missing from a particular dataset, as is the case with Wiktionary or GermaNet, we consider lexemes sharing the rightmost parent to belong to the same lexicographical block. Lastly, on input, every Russian word is losslessly transcribed into the Latin script, and transcribed back again on output. Evaluation is calculated in the original Cyrilic.

3.4.2 Experiments and evaluation

We use TensorFlow (Abadi et al., 2015) to build a custom architecture encoder-decoder recurrent neural model. It is equipped with a sequential decoder module (that we call the Retriever module) with Luong cross-attention for parent retrieval, and a classification head with self-attention for word formation classification. In Luong attention (Luong et al., 2015), the attention score is calculated by simply calculating the dot product between the relevant items from A and B. A trainable weight matrix may however optionally be used to multiply the item from A first, with no activation function.

We use a multi-lingual subword embedding model provided by *BPEmb* (Heinzerling and Strube, 2017) to feed semantic information about the input lexemes into the model. The model architecture can be viewed in Figure 3.4.

First, the input lexeme is byte-pair-encoded into a sequence of subwords, and its respective language token is embedded into a two-dimensional space. Then, the subword sequence is fed into the Encoder module, where each subword is in parallel:

1. embedded into a 300-dimensional semantic space provided by *BPEmb*;
2. split into individual characters and fed into a time-distributed bi-directional LSTM layer with a dimensionality of 300.

Thus, a 300-dimensional dual representation, one semantic and one character-based, is obtained for each word. These representations are then concatenated

Data	Lang	Comps	Derivs	Unmot	Authors
DeriNet 2.1	cze	2,240	264,748	13,748	Vidra et al. (2021a)
CELEX	dut	66,428	19,703	7,569	Baayen et al. (2014)
CELEX	eng	6,267	15,435	14,661	Baayen et al. (2014)
Wiktionary	eng	20,253	0	0	—
Unimorph	fre	161	72,789	2	Batsuren et al. (2022b)
MorphoLex	fre	313	0	6,655	Mailhot et al. (2020)
Wiktionary	fre	173	0	0	—
CELEX	ger	19,304	18,372	9,140	Baayen et al. (2014)
GermaNet	ger	99,080	0	0	Henrich and Hinrichs (2010)
Golden Comps	rus	1,699	0	0	Vodolazsky and Petrov (2021)
DerivBase.ru	rus	0	133,645	20,612	Zeller et al. (2014)
Unimorph	spa	130	30,646	1	Batsuren et al. (2022b)
DeriNet.ES	spa	0	42,825	16,141	Kyjánek et al. (2021)
Wiktionary	spa	329	15	0	—
All sources	All	216,377	598,178	88,529	—

Table 3.10: The data sources used in the training of *PaReNT*, grouped by language.

and fed into a bi-directional LSTM layer, the result of which is the output of the Encoder module. A stack of so-called Transformer-like blocks follows. This construct is similar to the familiar Transformer Multi-Head Attention Block, except it runs in a time-distributed manner, as opposed to its parallel-running Transformer counterpart. The number of stacked Transformer-like blocks was one of the hyperparameters that was tuned during the training process. The model then branches off into the Classifier module and the Retriever module.

In the Classifier module, self-attention is calculated on the Encoder output, and the sequence dimension is globally averaged over. Then, the result is passed through a fully-connected layer and passed through a three-unit Softmax layer.

In the Retriever module, the Encoder output is used to recursively generate the parent of the given lexeme grapheme by grapheme in the way that is described in (Vaswani et al., 2017). First, the input is fed into a self-attention block, which calculates attention between each pair of items from the input sequence. The attention is then added to the original input sequence. Next, alongside a skip connection, it is passed through a time-distributed fully-connected layer and added back. Finally, it is passed through a layer normalization.

For the evaluation of *PaReNT*'s performance in parent retrieval, we use Accuracy, which we define as the number of times *PaReNT* returned parents exactly string-equivalent (including capitalization) to the label parents in the correct ordering divided, by the number of items in the test set. To evaluate word-formation classification, we used Accuracy, defined as the percentage of class hits divided by the number of items in the test set. Because the datasets we used are generally imbalanced in terms of word-formation class, we additionally used Balanced Accuracy from *scikit-learn*, which is defined as $(Specificity + Sensitivity)/2$. As an auxiliary metric, we use Family accuracy, which describes the proportion of cases where PaReNT's retrieval output shares its word formation family with the label. We used this metric only on Czech, since DeriNet 2.1 is the only resource at our disposal with the required structure and completeness.

PaReNT

Lang	Retrieval accuracy	Classification accuracy	Class bal acc
Czech	0.64 (0.75)	0.96	0.66
German	0.60	0.95	0.86
English	0.69	0.86	0.84
Spanish	0.75	0.98	0.74
French	0.50	0.94	0.54
Dutch	0.55	0.89	0.80
Russian	0.64	0.97	0.72
Mean	0.62	0.94	0.74

Most-Frequent

Lang	Retrieval accuracy	Classification accuracy	Class bal acc
Czech	0.05 (N/A)	0.94	0.33
German	0.06	0.81	0.33
English	0.25	0.49	0.33
Spanish	0.18	0.82	0.33
French	0.08	0.91	0.33
Dutch	0.09	0.70	0.33
Russian	0.13	0.86	0.33
Mean	0.12	0.79	0.33

ChatGPT

Lang	Retrieval accuracy	Classification accuracy	Class bal acc
Czech	0.39 (0.66)	0.87	0.36
German	0.28	0.57	0.71
English	0.33	0.39	0.38
Spanish	0.3	0.64	0.64
French	0.4	0.53	0.31
Dutch	0.11	0.60	0.69
Russian	0.25	0.74	0.63
Mean	0.29	0.62	0.53

Table 3.11: The performance of PaReNT and baselines for each language.

The best model uses 2 bidirectional layers of $2,048$ units in the encoder, and a single Transformer-like block with two attention heads of dimensionality 512. In the retriever, Luong attention is used for Cross-Attention and 1 unidirectional layer of $2,048$ units to decode the output. In the classification head, Luong self-attention is used, with 512 units and a dropout of 0.3 in the final fully-connected layer. It was trained for 13 epochs, with a recurrent dropout of 0.2 in all recurrent layers and a regular dropout 0.5 of in all fully connected layers. The optimizer we used was ADAM, and we used a cyclical learning rate schedule (Smith, 2017) with a starting value of 10^{-4} and a final value of 10^{-5}. Its performance, broken down by language, can be found in Table 3.11.

PaReNT is directly compared against two baselines. The Most-Frequent baseline performs parent retrieval by returning the input unchanged, and always guesses *Unmotivated* as the category. The other baseline is ChatGPT (OpenAI, 2021), which is given the following prompt:

> Perform parent retrieval (predict which word or words the input lemma is motivated by) and word formation classification (predict whether the input lemma is a compound, a derivative, or unmotivated) on the given words. For each word, you will also be given its language of origin as a language token {cs : Czech, ru : Russian, de : German, es : Spanish, fr : French, nl : Dutch, en : English}. Format the output as tsv.
>
> The words:
>
> <list of words>

ChatGPT formats the output differently on each query, or sometimes even misunderstands the task or outright refuses to perform it, so its output has to be manually checked, regenerated if needed, and then reformatted. As a result, evaluation of ChatGPT is performed on a small sample of $n = 300$ words. These were fed into ChatGPT in increments of 100 words, prepended by the prompt each time.

The dummy balanced accuracy in classification is 0.33 for each language, because there are 3 word formation categories. Most-Frequent accuracy for retrieval is the same as the proportion of unmotivated words in the given language's test set. The figure in (*parentheses*) on the second line indicates Family accuracy, which describes how many times the system correctly identified the Czech word formation family of the correct parent(s). It is not listed for the Most-Frequent model, because it always returns the word unchanged, and a word is trivially part of its own word family in 100% of cases.

Comparing tables 3.10 and 3.11, it seems that the performance of the model for a given language not only depends on the amount of data available, but also on the morphological complexity of the language. For instance, despite the sparsity of the data available for Spanish, the model achieves high Accuracy classifying its lexemes. In Czech, the situation is the opposite – the amount of data available is large, but the performance is lower.

Not only does *PaReNT* outperform ChatGPT by a considerable margin in both tasks for every language (cf. Table 3.11), but also performs the tasks much more consistently. Since ChatGPT tends to format the output slightly differently upon each query, and also sometimes refuses to perform the task in the first place, *PaReNT* is much more suitable for pipelining in downstream applications.

We also compare *PaReNT* to *Word Formation Analyzer for Czech*. *PaReNT*'s performance in parent retrieval of Czech words at 0.64 is slightly lower than that of *WFA.ces* at 0.67. We attribute this to our splitting the data set by lexicographical block, which was not used in the case of *WFA.ces*.

We additionally attempt to compare *PaReNT*'s performance in splitting German compounds with Krotova et al. (2020)'s deep splitter, which was trained and evaluated solely on GermaNet. At face value, it achieves an Accuracy of 0.95. We subset our validation dataset so that only compounds from GermaNet are left, and we measure Accuracy on the subset and arrive at a markedly worse 0.69. However, we note the a) Krotova et al.'s tool is a split-point splitter, and is evaluated as such[7]; and also, PaReNT was trained not only on GermaNet, but also on CELEX, which has somewhat different annotation conventions. We use two different adjustments to try and take these differences into account. First, we adjust the performance of *PaReNT* to closer match the evaluation conventions of Krotova et al., as outlined in the Error Analysis section of their paper. We hand-annotate a 10% sample of the errors in the validation set in accordance with the classification set forth in the Error analysis section, and only consider errors of type **3**, **4**, **6**, **7**, and **8**, which amount to 51% of all the errors. After this adjustment, *PaReNT*'s Accuracy climbs to 0.84 compared to the 0.95 of Krotova et al. Second, we adjust the Accuracy of Krotova et al. to closer match our evaluation conventions, one of which is that all predicted parents must be valid words in their lemma form. However, only 60% of compounds in GermaNet are formed by a simple concatenation of their parent lemmas. As a result, the rest cannot be handled by a split-point splitter according to the aforementionded criterion[8]. We therefore consider the 60% to be the oracle score (maximum attainable value) for the split-point splitter, and arrive at 0.57 Accuracy for (Krotova et al., 2020)'s splitter compared to *PaReNT*'s 0.69.

3.4.3 Manual error analysis

Presented here is a linguist-performed analysis and interpretation of the errors made by *PaReNT*, in the hopes of not only shedding light on the tool's functioning, but also on the word formation systems of the languages in question. The errors have been analyzed by a human expert on a 1% random sample of the $179,720$ item validation dataset.

Due to the character-by-character decoding of PaReNT, it possible to retrieve any number of proposed parent sequences as generated by the model. For the purposes of evaluation and analysis in this thesis, the most probable candidate according to the model's beam search module ($beam\_size = 6$) was selected, but often, it was also interesting to see what alternatives the model emitted. For example, the model's behavior to some degree reflects the arbitrary nature of the distinction between compounding and derivation can be often rather arbitrary. For example, in the training data, the *-less* substring from *doubtless* is traditionally considered to be a suffix despite the attested appearance of the isolated word *less*, and therefore the model is trained and expected to return *doubt*, which the model correctly retrieves. However, as the second candidate retrieval, the model

[7] Interfixes are left attached to the left-hand parent.

[8] Unless it has explicit interfix handling, which the authors do not mention and their Error analysis section seems to indicate otherwise. The reason is explained and exemplified in Section 2.2.

proposes *doubt less*, interpreting the word as a compound (behaving similarly in cases like *headless* → *head less* and even cross-linguistically with *Arbeitslos* → *Arbeit los*). In other words, the model emergently learns to partially operate in a different linguistic framework than it was trained on.

The error classification that follows was however, as previously mentioned, made on the best available candidate for each input.

Type 1: Data conflict

The output of the model is correct, but conflicts with the label in the data.

In the data, each lexeme is assigned a single set of parents. Language reality is however often ambiguous, and the parents of a lexeme can be assigned in different ways. As a result, the model sometimes returns a lexeme that is correct, but disagrees with the label in the data, like in the Spanish ex. (73), where the expected output is *alcoholización* 'alcoholization'. Additionally, the datasets we used sometimes contain typos or other errors. This was in fact the case in the English ex. (74), where the listed output is in fact correct – and the item is wrongly listed as *Unmotivated*.

(73) ***desalcoholización*** → ***desalcoholizar*** (ES)
dealcoholization.N to dealcoholize.V

(74) ***north-northeast*** → ***north north east*** (EN)

Type 2: Inflectional confusion

The model mishandles the behavior of inflectional morphemes in word formation.

The role of inflection in word formation, typically compounding, has been touched upon in Section 2.2. The example *womenfolk* was used to illustrate that sometimes inflected words enter a word-formation process. When *womenfolk* specifically is fed into the model, it fails to return *woman* and instead returns *women*. It also occasionally generates an inflectional ending in a context where it looks like it could have been dropped (Spanish ex. (75), where *PaReNT* attaches a verbal ending to the expected *poem* 'poem").

(75) ***poema*** → ****poemar*** (ES)
poem.N NONSENSE

Type 3: Morphophonemic ambiguity

The model fails to compensate for a difficult-to-account-for morphophonemic process.

The bulk of the model's predictions are based on the reverse application of word-formation rules. For instance, the model notices that there exists a pattern in English <*root*>+*ment*, so upon seeing *development*, it returns *develop*. The problem is that it may be unclear how the rule should be retroactively applied. As an example, in Czech, the addition of a a suffix can induce stem allomorphy, resulting in /s/ → /š/. The application of the same suffix on another word, however, can yield /š/

→ /š/, so when analyzing a word of the pattern *s + <*suffix*>, the model has to guess whether to generate /s/ or /š/. In the Czech ex. (76), the expected result is *rychlokvasit* 'to ferment quickly'.

(76) ***rychlokvašený*** → ***rychlokvašit*** (CS)
quickly fermented.A NONSENSE

Type 4: Neural hallucination

The model baselessly hallucinates non-existent structures.

Sometimes, the model for unclear reasons simply switches, skips, replaces, overgenerates a character or split, or does something else that is difficult to interpret. Occasionally, it even hallucinates an entire morpheme, like in the Czech ex. (77), where an adjectival suffix and ending is added to the first nonsensical parent, or in the Dutch ex. (78), where an infinitive ending is added to both parents.

(77) ***přezůvka*** → ***přazový*** ***lažba*** (CS)
slipper.N NONSENSE NONSENSE

(78) ***hoegrootheid*** → ***hoegen*** ***roten*** (NL)
amount.N NONSENSE to rot.V

Type 5: Overretrieval

The model does not return the parent of the input, but the parent of the parent.

In ex. (79), we would expect PaReNT to output множитель 'factor' – the parent of which is the actually received output *множить* 'to multiply'. Similarly, we would expect the verb *mouler* 'to mold' as an intermediate step in ex. (80). This error often overlaps with **Type 1: Data conflict**, typically in cases where the model interprets a compound as parasynthetic when such an interpretation is unnecessary. This is the case in ex. (81), where *Wechsel* 'change' would be a simpler interpretation. The consistency of this error's appearance is reflected by the fact that Family accuracy is considerably higher than raw Accuracy, as shown in Section 3.11.

(79) ***множительный*** → ***множить*** (RU)
multiplicable.A multiply.V

(80) ***moulerie*** → ***moule*** (FR)
molding.N mold.N

(81) ***Tempowechsel*** → ***Tempo*** ***wechseln*** (DE)
change of tempo.N tempo.N to change.V

Type 6: False morphemehood

The model misjudges the presence of a morpheme, typically an interfix. In ex. (82), *PaReNT* mistakes part of the stem of the verb *wandern* 'to hike; to wander' for an interface, drops it, andends up with the completely unrelated word *Wand* 'wall'.

The most interesting aspect of this error is that its consistent presence across all languages seems to imply that the concept of the morpheme is in some way a meaningful way to break down language into constituent parts (as Haspelmath

2017 proposes), as opposed to a mere artifact of a particular linguistic tradition is the case with the distinction between inflectional and derivational morphology). The model for example (cf. ex. (82)) noticed that the *um/eum* originally Latin ending gets frequently dropped across many of the languages in the set, retrieving *Jagdmuseum* 'hunting museum' as *Jagd* 'hunt', **Muse* 'muse', but failed to take into account that it does not behave like a morpheme in German, so the expected result would be *Jagd* 'hunt' *Museum* 'museum'. In other words, the fact that the morpheme emerges out of a linear-algebra based approximation of word-trees completely unrelated to either the human brain implies that morphemes are more than a result of arbitrary choices made in the context of a particular linguistic tradition.

(82) ***Wandermöglichkeit*** → ***Wand Möglichkeit*** (DE)
hiking opportunity.N wall.N opportunity

Type 7: Semantic irrelevance

The model retrieves a word in a formally correct manner, but in a way that no human would find intuitive or meaningful.

The problem in ex. (83) is that while the compound is split on a morphological boundary, the given compound is not a compound of three words, but rather a recursive compound. A human knows that, because the first real parent *Wochenende* is a common word meaning 'weekend', but the model has no way of inferring this from its training data. The most obvious way to mitigate this type of error is to rely on the fact that such meaningless retrievals are unlikely to find support in corpora, because the probability of an appearance of a semantically redundant word like this is low.

(83) ***Oktoberwochenende*** → ***Oktoberwochen ende*** (DE)
October weekend.N October weeks.N end.N

Type 8: Missplitting

A non-compound is split (ex. (84)), a compound is left unsplit (ex. (85)), or a compound is split outside of a morphological boundary (ex. (86)).

(84) ***compartmental*** → ****compart mental*** (EN)

(85) ***bullring*** → ***bullring*** (EN)

(86) ***raceabout*** → ****racea bout*** (EN)

As a final note regarding the error analysis, it needs to be stressed that certain word-formation phenomena are easier for tools like PaReNT to model than others. For instance, the highly productive and common English derivational schema *develop* → *development*, *enjoy* → *enjoyment* is generally easily reconstructed by simply dropping the '-ment' suffix, while the unproductive and much rarer *broad* → *breadth*, *deep* → *depth* pattern is much harder, since there exist fewer examples and the sound changes are much less predictable. In other words, In the

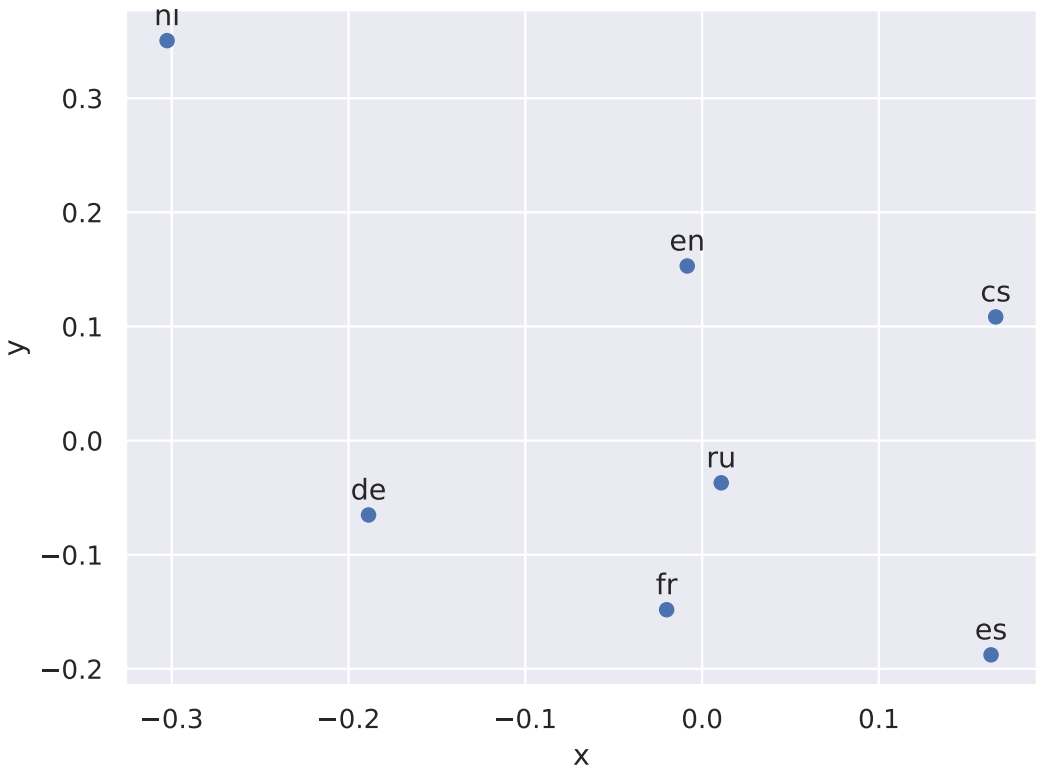


Figure 3.5: Vector space of PaReNT's Language Embedder module.

future, we would therefore like to develop an evaluation methodology that takes the complexity and frequency of certain word-formation processes into account.

3.4.4 Language Embedder analysis

As described earlier, one of the important parts of the model is the Language Embedder, which encodes each language token as a two-dimensional vector. A visualization of the underlying linear space of the language embedding module is shown in Figure 3.5. It can be used to show similarities between languages in terms of word formation. The closer the points are together, the more similar their word formation systems. The axes simply correspond to the two dimensions of the linear space, and have no obvious direct interpretation. The Embedder is trained just like everything else in the network, and the language embedding is concatenated alongside the semantic and character representations in the sequence right before the first Transformer-Like block. What this means is that the model, as it is trained, was during training forced to cluster together languages that are in some way or another similar with regards to their word formation systems, analogously to how word embedders cluster words that are similar in terms of meaning. The two dimensions were specifically chosen so that the embedding space can be viewed without the help of dimensionality reduction, whose choice of parameters may carry selection bias into the interpretation. Our initial hypothesis was that a clustering based on the languages' genetic or typological proximity would emerge, but this seems to not have happened, and we therefore consider this to be a negative result.

3.4.5 Final remarks

The deep-learning nature of *PaReNT* makes its usage somewhat computationally intensive. It does not however require a GPU to be practical, since the model processes about 20 lexemes per second on a CPU with batch inference, and takes up about 2 GB of disk space. While this does make it more cumbersome than e.g. a Snowball-based stemmer or a similar rule-based tool, it can still be comfortably used on a consumer-grade computer.

As a culmination of the efforts described in this chapter, we release a public version of *PaReNT*, alongside the part of the training data scraped from Wiktionary, which can now be found on GitHub alongside a short manual describing its usage.[9] *PaReNT* can be used in three ways:

- in **interactive mode**, which is run in a Linux terminal and is used for toying around and showcasing the tool;
- in **CLI mode**, which is also run in a Linux terminal, but takes in a `.tsv` file containing a `lemma` column and optionally a `language` column, and it outputs the same file with the following columns added:
    1. `PaReNT_retrieval_best`: Best parent(s), selected from PaReNT_retrieval_candidates based on columns 4), 5) and 6).
    2. `PaReNT_retrieval_greedy`: Parent(s) retrieved using greedy decoding.
    3. `PaReNT_retrieval_candidates`: All candidates retrieved using beam search decoding, sorted by score.
    4. `PaReNT_Compound_probability`: Estimated probability the word is a Compound.
    5. `PaReNT_Derivative_probability`: Estimated probability the word is a Derivative.
    6. `PaReNT_Unmotivated_probability`: Estimated probability the word is Unmotivated
- as an importable **Python package**, which provides the model architecture as a TensorFlow Model subclass alongside a training method, with the option of loading pre-trained weights trained on the data described in this thesis.

[9] https://github.com/iml-r/PaReNT

4. Annotating compounds in DeriNet

Armed with a tool that analyzes compounds, it is now time to demonstrate how its power can be harnessed to enrich existing data sources so that both compounding and derivation can be studied in the context of each other. The following chapter thus documents the process and results of using PaReNT's output and/or training data to reannotate and find compositional parents in DeriNet 2.1 (Vidra et al., 2021a), resulting in DeriNet 2.2 (Svoboda et al., 2024b). The chapter is broken up into two sections.

Section 4.1 – Annotation scheme describes the annotation scheme and the reasoning behind the decisions made. Despite the fact that this is the only chapter that is not multi-lingual, and only deals with Czech, we tried to design the annotation scheme such that it would carry over into other languages easily. Section 4.2 – DeriNet 2.2 goes over the details of the actual annotation process of DeriNet, the way *PaReNT* was utilized to help this process, and offers some insight into compounding as considered together with other word-formation processes as covered by DeriNet, in turn shedding some light into the word formation of Czech. The data set has been publicly released.[1] (Svoboda et al., 2024b).

4.1 Annotation scheme

In this section, we will describe the decision-making process behind which parents are supposed to be mapped to a given compound. In many cases, this mapping is fairly obvious, as is the case with *garážmistr* 'garage foreman', where the only reasonable option is a mapping to *garáž* 'garage' + *mistr* 'foreman', but in many other cases, it may not be. We consider two types of compounds – standard compounds and neoclassical compounds.

4.1.1 Standard compounds

Standard compounds are compounds whose parent sequence contains exactly zero neoclassical constituents. As already delved into in Section 2.1, most standard compounds can be traced back to a syntactic phrase (Scalise and Vogel, 2010). We therefore propose that linking compounds to their ancestors should reflect this observation, in that each compound should be assigned its corresponding syntactic phrase, and the parents are to be the lemmatized words from that phrase in the order that they appear in the given compound. If two or more phrases compete, we choose the one whose sequence of lemmas is most string-similar to the target compound.

However, there are cases where no such phrase is available (*spolupacient*). This is usually because there exists a so-called compositional schema (Booij, 2010). Here, compounds are described based on the formal properties that they share with other compounds coupled with analogous meaning, since the form-meaning

[1] https://lindat.mff.cuni.cz/repository/xmlui/handle/derinet22 (CC BY-NC-SA 4.0)

correspondence is a cornerstone of construction morphology. A simple example would be the following Czech schema:

$$[spolu \quad x_{N_j}]_{N_k} \rightleftharpoons [\text{WHO PLAYS ROLE j TOGETHER}]_k,$$

where j represents the input, N_j the fact that the input must be a noun, x_{N_j} represent that the input noun is a variable, N_k is the output noun, and the left hand side represents the form and the right hand side represents the meaning of the construction.

Example of this schema include *spolupacient* 'co-patient', spoluautor 'co-author', and *spoluzakladatel* 'co-founder'. This schema is actually a subschema of the more abstract schema

$$[x_{Adv_i} \quad y_{N_j}]_{N_k} \rightleftharpoons [\text{WHO PLAYS ROLE j IN AN i WAY}]_j,$$

This schema, in addition to *spolupacient*, *spoluautor* and *spoluzakladatel*, also covers e.g. *místokrál* 'viceroy' ← *místo* 'instead' + *král* 'king' and *rádobyodborník* 'so-called expert' ← *rádoby* 'so-called' + *odborník* 'expert'. In such cases, we try and preserve consistency across the given schema – in other words, we do not link *spolupořadatel* to *spolu* and *pořádat*, because this would then either force us to link *spolupacient* to the non-existent **pacientit*, or introduce schematic inconsistency.

The annotation workflow for a given compound c therefore goes as follows. Given c, do:

1. Check if c is part of an obvious established schema. If not;
2. Find an associated syntactic phrase p, lemmatize it, delete function words, and link words according to ordering in c. If a set of more such phrases P is found;
3. Select $p_n \in P$ whose lemmas are most string-similar to c after lemmatization, deletion of grammatical words, and word order rearrangement.

Item number 3 is actually proxy for *Select phrase p_n from the set of candidate phrases P that requires the least morphological operations to get from p_n to c*. It exists because e.g. *krvotok* 'bloodflow' can be associated with the set of phrases {*tok krve* 'flow of blood', *krev teče* 'blood flows'}. After we lemmatize, delete grammatical words, and rearrange word order, we get {*krev tok*, *krev teče*}, and so we choose the latter element, that is *krev tok*. Ultimately, the goal is always to map compounds onto their parent lexemes, and the tracing of their associated phrases serves only as a tool for that purpose.

Some compounds in Czech can be traced back to their parents very easily, because there is only one reasonable phrase that they can possibly represent. For example, *dřevodomek* 'wooden house' can only be reasonably traced to *domek ze dřeva* 'house made of wood' (ergo simply *dřevo domek*). The only reasonably possible competing phrase would be *dřevěný domek* 'wooden house', but such a phrase would yield the non-existent **dřevěnodomek*. Similarly, *černopáska* (a species of snail) can only be traced back to černá páska.

In the case of *spoluposluchač* 'co-listener', the situation is a bit more complicated. The phrase **posluchač spolu* 'listener together' makes no sense, however going a step further and tracing back to *poslouchat spolu* would force us to trace *spolupacient*

'co-patient' to the non-existent **spolu pacientit* 'to be patients together'. This rule was introduced by observing that in the data, such schematic clusters occur often, and we did not want to fragment these.

Pětiotvorový 'five-holed' needs to be traced back to *pět otvorů* 'five holes', because **pět otvorový* is not grammatical. Similarly, *stejnosměrný* 'same-directional' must be traced back to *stejný směr* 'same direction', since **směrný* does not exist.

In *krvotok* 'bloodflow', we run into the competing phrases (*krev teče* → *krev téci*) 'blood flows' and *tok krve* (→ *krev tok*) 'flow of blood', we select *krev tok*, since it is semanticall and formally closer. In contrast, with *psovod* 'dog handler' we are forced to trace back to *vést psa* 'handle a dog', because no **vod* exists. We do not consider this to be a violation of item no. 1 in the annotation workflow, because *téci* and *vést* are two different verbs and therefore are not the same schema.

Sometimes, the process of tracing a compound back to syntactic phrases must be guided by the annotator's knowledge of its usage. This is the case with *jihoamerický* 'South American', are traced back to *jižní Amerika* 'South America', not to *jih Ameriky* 'south of America'. The reason is that in the overwhelming majority of cases, *jihoamerický* 'South American' specifically refers to the entire continent of South America, as opposed to the southern part of one of the Americas.

Another problem that appears is that it is sometimes unclear whether the relation between the compound constituents is subordinative or coordinative, which can influence the tracing process. In *zemědělskolesnický* 'pertaining to agricultural forestry'/'pertaining agriculture and forestry', outside of context it is unclear whether the relation is subordinative and be traced back to *zemědělské lesnictví* 'agricultural forestry', or copulative and should be traced back to *zemědělství a lesnictví* 'agriculture and forestry'.[2] In cases like this, we rely on protypicality – which option is more typical? We therefore trace back *zemědělskolesnický* to *zemědělský (a) lesnický* 'agriculture and forestry', *frýdeckomístecký* 'from Frýdek-Místek' to *Frýdek Místek* (as it is a toponym), and *americkofrancouzský* to *americký (a) francouzský* 'American and French'.

A troublesome situation arises when there is a pair of compounds which are derived from the same parents, but could also be considered derivatives of each other. That is the case with *divotvorný* 'miracle-working' and *divotvůrce* 'miracleworker', with no **divotvor* or **divotvůr*. In such cases, we select the most string-similar option, and map *divotvorný* to *tvorný div(ů)* 'miracle working' and *divotvůrce* to *div(ů) tvůrce*.

In this scheme, we only consider *primary compounds*. Secondary compounds are, in accordance with the current DeriNet API convention, understood as derivatives or conversions of compounds.

4.1.2 Neoclassical compounds

Within the languages in scope, originally Greco-Latin roots (or what are perceived as such by speakers) play a special role in the lexicon (cf. Section 2.1.4). What that sets these roots apart from most others in the languages in scope is that

[2] According to Czech orthography, except for a short list of exception such as *černobílý* 'black-and-white', coordinative compounds should be spelled with a hyphen. However, this rule is often ignored, and therefore cannot be relied upon.

these *neo-classical constituents* do not appear on their own. For example, the Greco-Roman root *-log-*, having the very vague meaning of 'pertaining to words, language, knowledge, communication', never appears as an isolated **log* with that meaning, but appears as what we call an *attested variant* with a much more specific meaning, in the case of *log* of *diary* or *list of events*. Further examples for the -log- neoclassical constituent include derivation (*logic*), or the compounds *psychology*, *logography*[3].

An interesting aspect of these neoclassical constiuents is the fact that they are shared across all of the languages in scope, and thus form a shared, mostly scientific and scholarly lexical stratum. For instance, if a neoclassical compound, such as *archaeology*, is found in English, there is a high probability that a corresponding neoclassical compound with a similar meaning can be found in the other languages in scope. This introduces a strong incentive to label them in a way that is at least somewhat language agnostic.

To achieve that, we label these constituents with hyphens on both ends, to visually demonstrate that they can serve as both suffixes or prefixes. In contrast with Ološtiak and Vojteková, who model this root as either a prefix (*logo-*) or a suffix (*-logy*), we consider the shared root in all of these cases be the same object, labelled as *-log-*. Thematic vowels, which in the terminology of Ancient Greek morphology denotes the stem-trailing vowels determining which morphological schema a given stem follows (such as -o- as in logography or -e- + -o- in teleology)are therefore usually omitted, except in cases where doing so would lead to confusion. Details on their orthography are difficult to establish, since while the meaning and rough phonology of the constituents are cross-linguistically shared, their orthography generally is not – which can occur intra-linguistically as well, compare the English spelling variants *archaeology* vs. *archeology*. We therefore prefer a strong correspondence to the original Greek spelling transcribed into the Latin script according to the customs of Roman scribes. *Archaeology* and *archeology* therefore share the same constituent -archai- coming from the ancient Greek ἀρχαῖος 'old'.

A good way of understanding the need for introducing this synchronic model stems from the observation that in general, scientifically literate native speakers of the languages in scope have the ability to at least roughly infer the meaning of a given neoclassical compound despite the fact that they may have no direct knowledge of either Greek or Latin. As an example, a reader of this dissertation might be able to infer that *autophagy* (transparently from -aut- 'itself' and -phag- 'devour') refers to eating oneself in some way – which is roughly correct, since the term refers to a biological process of cellular orderly self-degradation. Similarly, the word *logographer* (-log- 'speech' and -graph- 'write') refers to a writer of speeches; a *macrophage* is a large eater, and refers to a massive white blood cell that rids our bodies of harmful microorganisms by devouring them whole.

This precludes the possibility of capturing these neoclassical compounds as root nodes in DeriNet (since the data resources only links existing lexemes to existing lexemes), because that would imply that these words are opaque to native speakers. Since they are not root nodes, they must be linked to something – however, as we have established, the neoclassical compounds appear on their own, which in turn precludes their inclusion in a lexical data resource such as DeriNet.

The solution is a compromise. We propose including neoclassical constituents as root nodes, with Neocon as the part-of-speech tag and hyphens on both sides

[3] In combination with the neoclassical derivative suffix *-y*.

Table 4.1: Horizontal table of Czech neoclassical constituents.

Neocon	-psych-	soul, mind, sanity
Attested variant	*psycho*	trippy, trippiness
Etymology	ψυχή (Greek)	soul
psychologie 'psychology', *algopsychalie* 'algopsychalia', *psychóza* 'psychosis'		
Neocon	-log-	science, speech
Attested variant	*log*	computer log
Etymology	λόγος (Greek)	speech; word; thought
pedologie 'pedology', *logography* 'logografie', *logo* 'logo'		
Neocon	-mini-	small version of
Attested variant	*mini*	miniskirt
Etymology	minimus (Latin)	smallest
miniauto 'minicar', *minisukně* 'miniskirt'		
Neocon	-crypt-	hidden, secret
Attested variant	*krypta*	underchurch vault
Etymology	κρυπτός (Greek)	hidden
kryptografie 'cryptography', *kryptofašista* 'cryptofascist', *krypto* 'cryptocurrency'		
Neocon	-aut-	it(self), spontaneous
Attested variant	*auto*	car
Etymology	αὐτός (Greek)	it(self), spontaneous
automobil 'automobile', *autofágie* 'autophagy'		
Neocon	-tel-	over distance, end
Attested variant	–	–
Etymology	τῆλε (Greek)	end
telegraf 'telegraph', *autotelický* 'autotelic', *telomer* 'telomer'		
Neocon	-haim-	blood
Attested variant	–	–
Etymology	αἷμα (Greek)	blood
hemoglobin 'hemoglobin', *anémia* 'anaemia'		
Neocon	-hemi-	half
Attested variant	–	–
Etymology	ἥμισυς (Greek)	half
hemisféra 'hemisphere', hemikrystalický 'hemicrystallic'		
Neocon	-bio-	life
Attested variant	*bio*	organic products
Etymology	βίος (Greek)	life, force
biologie 'biology', *biogeografie biogeographics*, anabióza *anabiosis*		
Neocon	-bi-	two, both
Attested variant	*bi*	bisexual
Etymology	bini (Latin)	twice
bigamie 'bigamy', *bicykl* 'bicycle', *binaurální* 'binaural'		
Neocon	-di-	two, both
Attested variant	–	–
Etymology	δίς (Greek)	twice
dichotomie 'dichotomy', *digraf* 'digraph'		
Neocon	-dis-	into (two or more) parts
Conversion	–	–
Etymology	dis- (Latin)	asunder
diskriminace 'discrimination'		

as proposed above. Neoclassical compounds are then to be linked to all their associated constituents in much the same manner as standard compounds.

It is interesting to point out that apart from compounding, neoclassical constituents can be modeled as undergoing any other word formation process, not just compounding. As a result, many neoclassical constituents can by modeled as having undergone e.g. conversion[4] in Czech, resulting in e.g. (*-krypt-* → *krypta* 'church crypt') or derivation (*-psych-* → *psychika* 'psyche'), leading to some having attested variants (but not all – *-typhl-* meaning 'blind' has no **tyfla* or **tyflický*). While we model the lexicon of unattested neoclassical constituents as being shared cross-linguistically, their attested variants may be language-specific, as evidenced by the aforementioned English *log*, which has no equivalent in Czech.

The attested variants of neoclassical constituents, we propose, should be linked either to a neoclassical compound should they obviously be a clipping thereof (like the Czech *auto* 'car' should be linked to *automobil* 'motorcar'), or directly to the associated neoclassical constituent if such an association is not clear (the English *psychic* should be linked directly to *-psych-*). This would lead to a multilingually consistent and intuitively legible word-formation trees, as demonstrated in Figure 4.1. As of now, derivatives/conversions of neoclassical constituents and clippings of neoclassical compounds are not handled in DeriNet 2.2 in this way, but the figure shows how such words, e.g. *psychika* 'psyche', *pathos* 'pathos', could be handled in the future. Specifically, we would like to link these items directly to the neoclassical constituents *-psych-* and *-path-*, respectively. The figure also illustrates that judgment calls will have to be made regarding the mapping of neoclassical compounds containing more than one constiuent. On a surface level, it would perhaps make sense to map *psychopathologie* 'psychopathology' to *psychopat* 'psychopath', a person exhibiting impaired empathy and remorse (among others), and *psychopathology* denotes the study of mental illness in general and not specifically psychopathy. It therefore makes sense to link it to *-psych-* 'soul' and *patologie* 'pathology', better reflecting the form/meaning correspondence.

A table displaying a diverse set of examples can be found in Table 4.1. Each segment is its own mini-table, with four rows. The first row of each segment is the neoclassical constituent itself, the second row is an associated attested variant (if it exists in Czech), and the third row is the word or morpheme as it existed in its original language. The final row lists examples of words which contain the given neoclassical constituent. The first of the three columns is the legend; the second gives the form of the linguistic object in question; the third gives the meaning.

4.2 DeriNet 2.2

In this section, we describe DeriNet version 2.2. First we go over the practicalities that we ran into applying *PaReNT* to the annotation scheme described earlier, and then we show statistical results and interpretations pertaining to the newly-linked compounds found in the new version of the data set.

[4]Unlike in the English tradition, in Czech the sole addition of an inflectional ending, i.e. *-a*, is considered conversion.

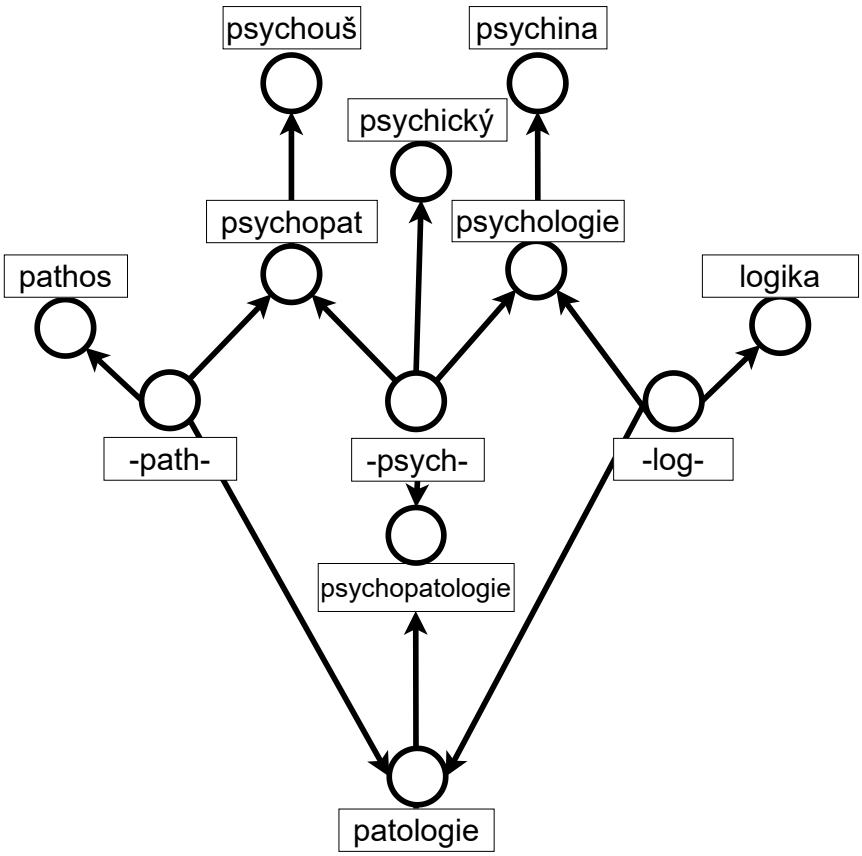


Figure 4.1: Proposal how to model neoclassical compounds, constituents, and their derivatives in a cross-linguistically consistent manner.

4.2.1 Creating version 2.2

The first problem we ran into when annotating DeriNet's compounds was that many of them had parents not present in DeriNet 2.1, because they belonged to a part-of-speech category that had previously been excluded from the data source – specifically, pronouns (such as *se* 'him/her/itself'), numerals (*dvě* 'two', *půl* 'half'), and adpositions (*mimo* 'outside of') (see 2.3). The first step in the creation of DeriNet 2.2 was therefore the rollback of that decision, and the following addition of numerals, adverbs, and pronouns. 163 pronouns, 774 numerals, and 95 adpositions were taken from the MorfFlex database (Hajič et al., 2020) – which is historically the original source of DeriNet's underlying lemmaset – and added to DeriNet alongside their respective POS. Some (but not all) numerals contained information on their numerical value, e.g. the numeral *třiadvacet* contains the value 23. We used the *num2words*[5] package to convert this value into words and, with manual correction, find the compounding parents of such numerals.

The current version of PaReNT's Retriever module supports not only returning a single parent sequence, but it can also return a list of candidate parent sequences. This allowed us to develop a *parent candidate scoring function* that takes into account

1. ordering of the candidate list (i.e. PaReNT's implicit scoring function);

[5] https://github.com/savoirfairelinux/num2words

Word type	DeriNet 2.1	DeriNet 2.2	Added
Neoclassical constituents	202	285	83
Compounds	1,952	6,336	4,384
Derivatives	782,814	782,904	90
Variants	50,533	50,511	-22
Unmotivated	203,079	199,668	-3,411
Conversions	144	135	-9
Total size	1,039,012	1,040,127	1,115

Table 4.2: Differences in word types in DeriNet 2.2 and DeriNet 2.1.

2. presence of all the candidate parents in DeriNet;
3. presence of all the candidate parents in DeriNet except the last letter.

Item number 3 is there because PaReNT tends to drop the ending of the first constituent (cf. Section 3.11). However, such minor misretrievals can still be valuable during the annotation process, because filling in a missing ending is faster than writing out a whole parent sequence.

The exact formula of the scoring function for a candidate sequence is

$$S_{cs} = \frac{1}{i} + \sum_{1}^{cs} w(cs_i),$$

where cs refers to the candidate parent sequence, S_{cs} to its score, p to a parent, i to its index in PaReNT's output, and $w(x)$ refers to a weighting function that assigns scores based on the presence in DeriNet. $w(x)$ is defined as

$$w(x) = \begin{cases} 10 & \textit{if } x \in DeriNet \\ 9 & \textit{if } x \in DeriNet[:-1] \\ 0 & \textit{if } x \notin DeriNet \wedge x \notin DeriNet[:-1] \end{cases}$$

where $DeriNet$ is the set of all lemmas in DeriNet 2.2 and $DeriNet[:-1]$ the set of all lemmas in DeriNet except their last letters.

A given parent sequence therefore receives 1 point if it's the first in PaReNT's output, and an additional 10 points for each predicted parent that is present in DeriNet, or, failing that, 9 points if it's present in DeriNet disregarding last letters. If it is present in neither, the parent receives zero points. The candidate sequence list coming from PaReNT was thus re-scored, and the best-scoring parent sequence was the one being corrected during the annotation process.

In this way, we hand-annotate $5,022$ compounds so that they are compatible with the new annotation scheme described in Section 4.1.1, plus re-annotating the ones that were already there, according to the annotation scheme presented in Section 4.1. All the new compounds were detected using PaReNT, but as shown in Table 3.10, the tool does not score a 100% in word formation classification. Therefore, some derivatives had been misclassified as compounds, and were annotated as such in case they had previously been recorded as unmotivated, which is the reason for the addition of 90 derivatives. The annotation was performed by

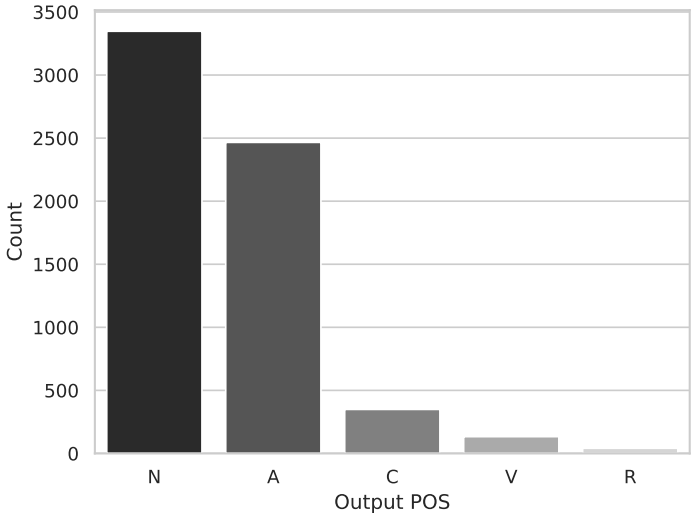


Figure 4.2: Output POS distribution of compounds in DeriNet 2.2 – N is Noun, A is Adjective, C is Numeral, V is Verb, R is Adverb.

manually correcting the best candidate sequence (according to the S_{cs} candidate scoring function) from the output of PaReNT's retrieval module. In addition to these hand-annotated compounds, we were able to find $1,398$ additional compounds by pattern-matching neoclassical constituents and other common modifiers (*sám* 'self', *půl* 'half' ...) with unmotivated words in DeriNet. This approach is lemma-based, but some lexemes in DeriNet share their lemmas with other lexemes, which leads to ambiguity when trying to link these lexemes. The final round of annotation therefore consisted of resolving these ambiguities, of which there were 820.

It is important to note than in the current version of DeriNet, the label of compounding does not propagate downward in the word-formation tree. That is, only immediate or primary compounds such as *dřevorubec* 'woodcutter' are labelled as compounds; their derivatives and other word-formation children, a.k.a secondary compounds (in this case e.g. *dřevorubecký* 'pertaining to woodcuttery'). Table 4.2, which shows the POS distribution of each compound in DeriNet reflects this, and only shows primary compounds. However, secondary compounds can be traced by starting off at each compound and recursively travelling downward in its subtree. By doing so, we discover that, apart from the $6,336$ aforementioned primary compounds, there are $14,750$ secondary compounds (and their children, and the children of their children ...) currently traced back to their parents.

4.2.2 Statistical analysis

The statistics presented here are calculated on the set of $6,336$ primary compounds that were linked to their parents by the procedure described above. Although our annotation increased the amount of compounds in DeriNet over threefold compared to the previous version, it still covers only a part of the compounds present in this resource.[6] The reason is our adherence to DeriNet's general policy

[6]We tried to estimate the coverage of the current annotation by sampling 200 lexemes from DeriNet, excluding capitalized lexemes such as given names, and hand-annotating them regarding

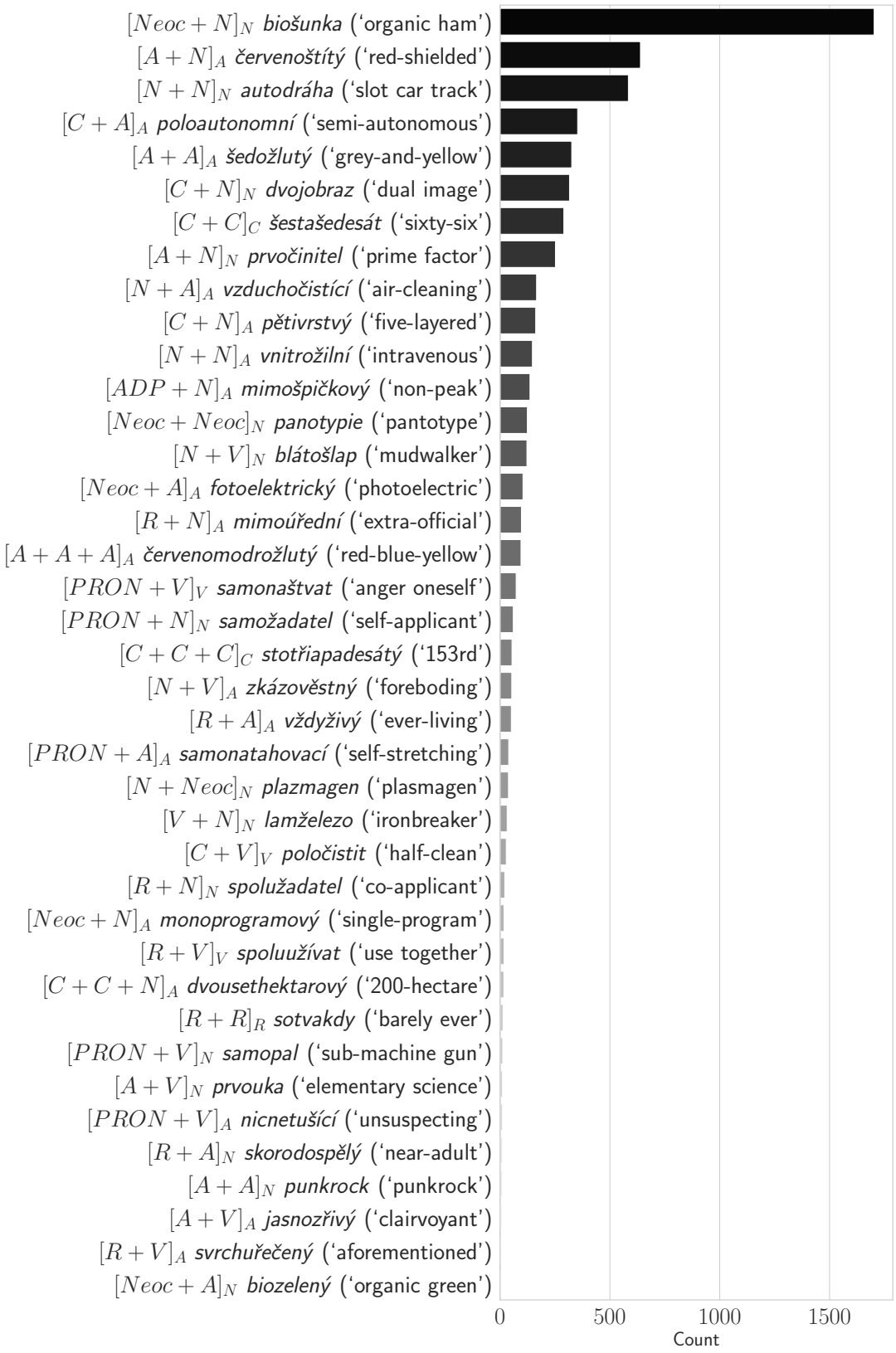


Figure 4.3: Distribution of compound patterns for number of entries with examples and translations in DeriNet 2.2, filtered for $n > 5$ occurrences in the data set.

of highly prioritizing precision over recall; meaning that it is considered much less of a problem if a word is left unlinked compared to incorrectly linking a word to the wrong parent(s). As a result, we were extremely precise, and actually went over the annotated set many times to fine-tune and tweak the annotation scheme to a high standard.

The first observation we measure regarding primary compounds is their output part-of-speech distribution, illustrated in Figure 4.2. We see that nominal-output compounds are the dominant part-of-speech, but adjectival compounds are not far behind.

We attribute the relative abundance of adjectival compounds to a very common specific pattern of parasynthetic compounding on the boundary between word formation and syntax, wherein a compound coinage is forced by the necessity to use a adjective-noun noun phrase as a modifier, resulting in a $[A + N]_A$ compound. The underlying noun phrases can be

- place names as in *jihoamerický* (ex. (87)) or *lysohorský* (ex. (88); Lysá hora is a mountain in the Czech Republic), which cover ca. 40% of $[A + N]_A$ compounds,[7]

  (87) ***jihoamerický*** ← ***Jižní*** + ***Amerika*** (CZ)
  South American.A South.A America.N

  (88) ***lysohorský*** ← ***Lysá*** + ***hora*** (CZ)
  from *Lysá hora*.A Bare.A mountain.N

- terms describing scientific, scholarly, administrative, or political fields or concepts, as in *sociálněpedagogický* (ex. (89)) or *právněhistorický* (ex. (90)),

  (89) ***sociálněpedagogický*** ← ***sociální*** + ***pedagogika*** (CZ)
  socially pedagogical.A social.A pedagogics.N

  (90) ***právněhistorický*** ← ***právní*** + ***historie*** (CZ)
  pertaining to history of law.A law.A history.N

- descriptions of body parts, such as *ploskohlavý* (ex. (91)) or *červenoštítý* (ex. (92)) lit. 'red-plated', but actually the specific epithet of the species *Dinoptera collaris*),

  (91) ***ploskohlavý*** ← ***ploský*** + ***hlava*** (CZ)
  flat-headed.A flat.F.SG.A head.N

  (92) ***červenoštítý*** ← ***červený*** + ***štít*** (CZ)
  red-shielded.A red.A shield/plate.N

- or any other kind of adjective-noun phrase, as in *černoděrový* (ex. (93)) or *každodenní* (ex. (94)).

  (93) ***černoděrový*** ← ***černý*** + ***díra*** (CZ)
  pertaining to black holes.A black.F.SG.A hole.N

their compoundhood status. The estimate based on this sample is that there are roughly $53,000$ primary compounds in DeriNet 2.2. Of that, $6,336$, or about 12%, are now mapped to their parents.

[7] Estimated by checking if any parent in the compound is capitalized.

(94) ***každodenní*** ← ***každý*** + ***den*** (CZ)
everyday.A every.A day.N

Evidence for the claim that syntactic needs are responsible for a large chunk of compound coinages in Czech can be found in Figure 4.3, where the distribution of compound patterns, with respect to both the POS of their parents and their output in DeriNet 2.2, is visualized. Here we see that the $[A + N]_A$ pattern is the second most common pattern after nominal-output hybrid neoclassical compounds, which are represented by the pattern $[Neocon + N]_N$.

We found a total of 103 unique compound patterns. However, most of them have very few occurrences in DeriNet, so we show only the ones that have more than 5 occurrences in Figures 4.3 and 4.4 and Table 4.3.

The third most common compound POS is the numeral-output compound, and it lags behind adjectival-output compounds by roughly an order of magnitude. While numerals play an important role in Czech compounding, they generally only play the role of modifiers, with the only consistently productive pattern of numeral-head or numeral-output compounds being $[C + C]_C$ such as *dvaasedmdesát* 'two-and-seventy' or *třiatřicet* 'thirty-three' and $[C + C + C]_C$ 'třistaosmdesátšest'.

Verbal-output compounds are even rarer, with only four patterns reaching the cutoff of 6 or more examples in the data. Most of these fall into specific schemas, and they are as follow:

- $[Pron \quad V]_V$, whose modifier is usually the pronoun *sám* 'alone' and *se* 'self', with the meaning usually being *to do something by one self or to oneself* – e.g. *samonaštvat* 'make oneself angry', *sebeopylit* 'autopollinate',

- $[R \quad V]_V$, whose modifier is usually the adverbs *spolu* 'together' or znovu *again*, with the meaning being *to do something together/again* – e.g. *spoluužívat* 'use together', *znovustvořit* 're-create'

- $[C \quad V]_V$, whose modifier is always the numeral *půl* 'half', with the meaning being *to half-do something* – e.g. *polospát* 'to be half-asleep'.

However, in spite of their relative rarity, verbal-output compounds seem to be remarkably productive – meaning that they often undergo further word formation – compared to other types of compounds in Czech. Inspecting Table 4.3, we notice that the first two of these three patterns produce more descendants compared to all the other compound types. The reason for this is that for all the verbs in DeriNet, the average amount of word-formation descendants is 25.30 (counting in possibly unattested nodes), which is much higher than for example the average number of descendants for nouns which is 2.34. The numbers of around 14.00 seen for the verbal-output compounds, while lower, therefore simply shadows the high baseline productivity of verbs in Czech.

The situation with nouns is different. The average amount of descendants for non-proper nouns is 2.34, which falls within the general range of nominal-output compounds, lying between 0.00 – 2.72 depending on type, with the exception of the highly productive neoclassical $[Neoc + Neoc]_N$. The productivity of adjectives, similar to verbs, seems to be affected by compounding, seeing that the average number of descendants for adjectives in general (2.44) falls just out of the general

Compound type	Depth	Descendants	Example	Translation
$[PRON + V]_V$	2.99	18.70	*samonaštvat*	'anger oneself'
$[R + V]_V$	2.72	14.22	*spoluužívat*	'use together'
$[C + V]_V$	2.29	12.11	*poločistit*	'half-clean'
$[PRON + V]_A$	1.73	10.73	*nicnetušící*	'unsuspecting'
$[A + V]_N$	1.45	10.09	*prvouka*	'elementary science'
$[Neoc + Neoc]_N$	1.07	8.38	*panotypie*	'pantotype'
$[A + V]_A$	1.62	3.25	*jasnozřivý*	'clairvoyant'
$[N + V]_N$	1.12	2.72	*blátošlap*	'mudwalker'
$[PRON + V]_N$	1.17	2.58	*samopal*	'sub-machine gun'
$[R + V]_A$	1.14	2.43	*svrchuřečený*	'aforementioned'
$[N + V]_A$	1.04	2.13	*zkázověstný*	'foreboding'
$[N + A]_A$	1.06	2.09	*vzduchočistící*	'air-cleaning'
$[C + N]_A$	1.02	2.02	*pětivrstvý*	'five-layered'
$[A + A]_A$	1.02	1.98	*šedožlutý*	'grey-and-yellow'
$[PRON + N]_N$	0.90	1.92	*samožadatel*	'self-applicant'
$[A + A + A]_A$	1.00	1.92	*červenomodrožlutý*	'red-blue-yellow'
$[A + N]_A$	1.00	1.86	*červenoštítý*	'red-shielded'
$[R + N]_N$	0.95	1.86	*spolužadatel*	'co-applicant'
$[C + A]_A$	1.00	1.83	*poloautonomní*	'semi-autonomous'
$[R + A]_A$	0.90	1.71	*vždyživý*	'ever-living'
$[Neoc + A]_A$	1.04	1.67	*fotoelektrický*	'photoelectric'
$[A + A]_N$	1.00	1.67	*punkrock*	'punkrock'
$[R + N]_A$	1.01	1.61	*mimoúřední*	'extra-official'
$[ADP + N]_A$	0.97	1.55	*mimošpičkový*	'non-peak'
$[N + N]_A$	0.95	1.55	*vnitrožilní*	'intravenous'
$[N + Neoc]_N$	0.87	1.47	*plazmagen*	'plasmagen'
$[Neoc + N]_A$	0.83	1.39	*monoprogramový*	'single-program'
$[C + N]_N$	0.77	1.39	*dvojobraz*	'dual image'
$[V + N]_N$	0.75	1.34	*lamželezo*	'ironbreaker'
$[PRON + A]_A$	0.69	1.33	*samonatahovací*	'self-stretching'
$[N + N]_N$	0.68	1.32	*autodráha*	'slot car track'
$[Neoc + A]_N$	0.43	1.29	*biozelený*	'organic green'
$[C + C + N]_A$	1.00	1.18	*dvousethektarový*	'20-hectare'
$[A + N]_N$	0.63	1.12	*prvočinitel*	'prime factor'
$[Neoc + N]_N$	0.39	0.76	*biošunka*	'organic ham'
$[R + R]_R$	0.08	0.08	*sotvakdy*	'barely ever'
$[C + C]_C$	0.01	0.02	*šestašedesát*	'sixty-six'
$[C + C + C]_C$	0.00	0.00	*stotřiapadesátý*	'153rd'
$[R + A]_N$	0.00	0.00	*skorodospělý*	'near-adult'

Table 4.3: Productivity of each compound type in Czech, measured on their word-formation subtree depth and number of word-formation descendants, sorted by no. of descendants.

range adjectival compounds (1.26 – 2.43), excluding the $[Pron + V]_A$ pattern, which seems to be rare ($n = 11$) and which contains the hugely productive outlier *samostatný* 'independent'. Compounding seems to have a dampening effect on further word formation in the case of compound verbs and compound adjectives, whereas compound nouns seem to behave much like any other Czech word in that regard. This observation may be somewhat skewed, however, by the fact that DeriNet is as of now incomplete, in the sense that some lexemes are not yet linked to their derivational ancestors.

DeriNet additionally allows us to gauge how far from its root a given word is – in other words, how tall a given word 'supertree' is. We call this the *height*) a given word. For example, the height of *spinelessness* is 2, because it is a derivative of *spineless* (height 1), which in turn is a derivative of *spine*, which is an unmotivated word (height 0) We calculate the average height of each parent for each compound pattern. This is shown in Figure 4.4. We observe that a given part-of-speech category may exhibit very different heights depending on which position or in which compound type it is included.

For example, adjectives in the most common pattern of multi-parent compounding ($[A+A+A]_A$) seems to be composed almost exclusively of unmotivated parents. This is because this pattern almost exclusively involves colors (e.g. *červenomodrožlutý* 'red-blue-yellow'), which tend to be unmotivated words.

In general, there seems to be an overall preference for the modifier to have lower height than the head of the compound across the board, although there are three[8] notable exceptions.

The first is the already discussed $[A + N]_A$ highly common parasynthetic pattern. This is at least partially because the modifier is often a color, basic quality (big/small), or cardinal direction (north/south/west/east), which are all usually unmotivated words. The second is yet another parasynthetic pattern, $[N + V]_A$. Here, the difference is caused by the fact that Czech verbs entering parasynthetic compounding seem to be mostly unmotivated across the board. In contrast, the verbs in $[X + V]_V$ compounds seem to be much deeper.

[8]Strictly speaking, there are more, but these are uninteresting – e.g. anything with a neoclassical constituent in the head (=for Czech, almost exclusively rightmost) position, since neoclassical constituents always have height zero, as they are unmotivated by definition; or the $[R + R]_R$ type, where the difference is negligible.

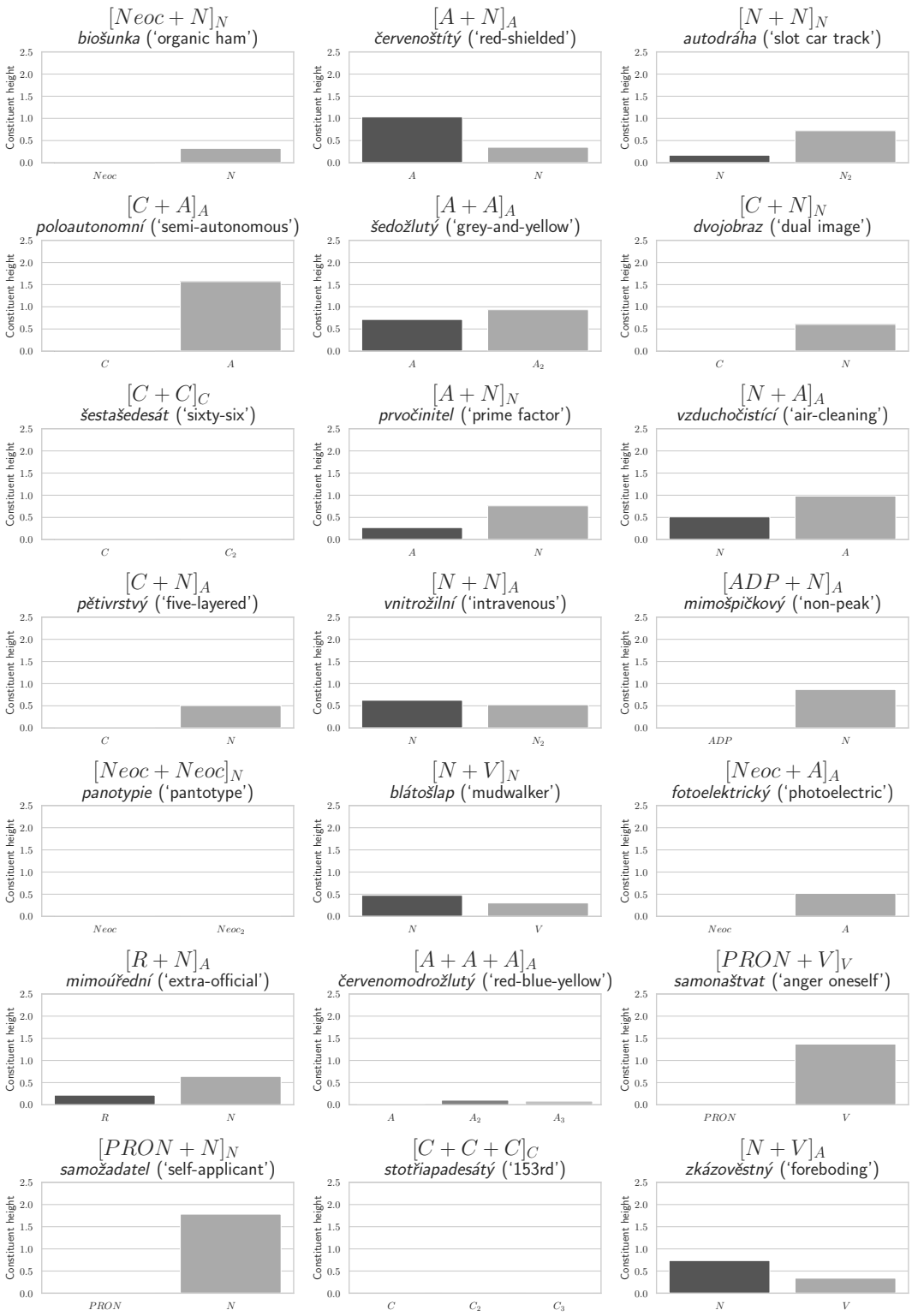


Figure 4.4: Average word-formation history (height) of each parent for the compound types in DeriNet for $n > 5$.

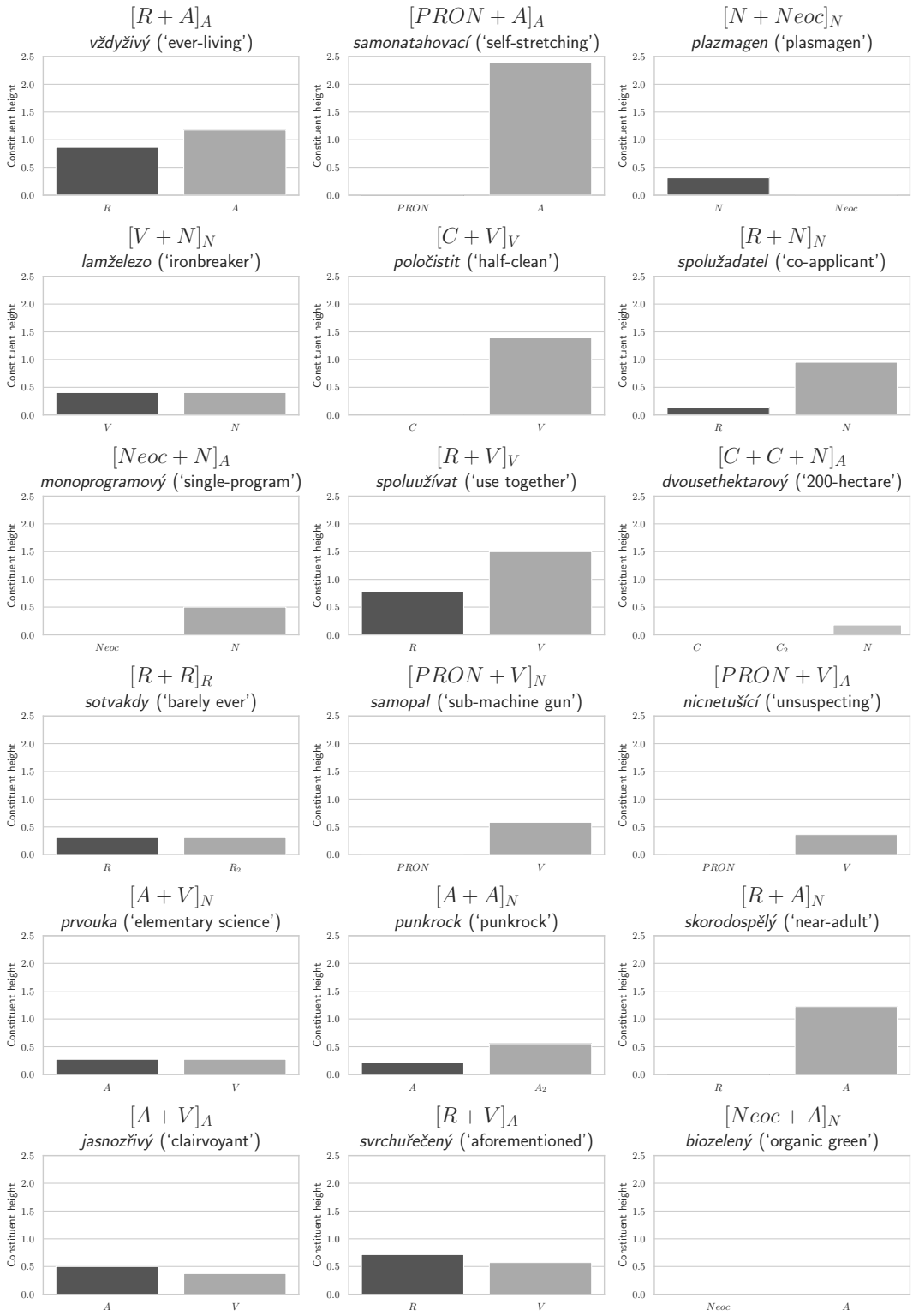


Figure 4.4: (continued)

5. Incorporating compounds into Universal Dependencies

So far, this dissertation has only dealt with modelling compounds in the context of word-formation data resources. But word-formation data resources such as DeriNet generally only capture the parents of a given compound, not the relation between them. Instead of forcing a syntactic label into DeriNet, we instead venture into placing compounds directly into a syntactic data resource. This thesis therefore wraps with a vision to explore shores unknown, and proposes a way to fit compounds into a syntactic data resource that **can** capture the relations between constituents, specifically Universal Dependencies (UD), which is a multilingual dependency treebank. The data resource represents an opportunity to make explicit the claim that compounds encode syntactic phrases.

In Universal Dependencies, compounds are represented according to tokenization, which reflects the orthographic conventions of the language. A closed compound corresponds to a single word in Universal Dependencies (e.g. *waterfall*) while a hyphenated compound (*father-in-law*) and an open compound (*apple pie*) to multiple words. The aim of this chapter is to open a discussion on how to move towards a more consistent annotation of compounds. The solution we argue for is to represent the internal structure of all compound types analogously to syntactic phrases, which would not only increase the comparability of compounding within and across languages, but also allow comparisons of compounds and syntactic phrases.

The chapter is structured as follows. We first describe how compounds are currently handled in UD, exemplifying the general and language-specific problems of compounds. Then we discuss steps that can be taken to make the annotation of compounds more coherent and to bring it closer to the way syntactic relations are annotated, but without losing the difference between compounding and syntax. Future directions regarding the automation of compound identification and annotation are outlined to some extent. The version of Universal Dependencies we are operating with throughout the chapter is v2.12 (Zeman et al., 2023).

5.1 Current annotation

We start by introducing how words considered as compounds in the literature are treated according to the UD annotation principles (de Marneffe et al., 2021).[1] The application of these rules to each of the languages under survey is described in the following subsections. Syntactic annotation in UD is based on tokenization, which in turn follows the spelling conventions of individual languages. Since the term compound covers words spelled in several ways, compounds are not annotated uniformly in UD.

[1] See also https://universaldependencies.org/guidelines.html

Dataset	Language	Compounds	Entries
CELEX2 (2014)	English	6,267	52,447
CELEX2 (2014)	German	19,304	51,728
GermaNet (2011)	German	121,655	215,000
DeriNet 2.1 (2021a)	Czech	45,473	431,857
Word Formation Latin (2016)	Latin	3,198	36,258
Golden Compound Analyses (2021)	Russian	1,699	1,699

Table 5.1: The databases employed in the present survey for identification of compounds in the Universal Dependencies treebanks of the five languages. The last two columns specify the number of lemmas (types).

5.1.1 Guidelines

Closed compounds, appearing in the text as continuous orthographic words, are handled as discrete, internally unstructured (= atomic) items which enter into relations with other items of the sentence structure. Although the compound's components are linked by similar relations as the constituents of syntactic phrases, these intra-word relations are not captured in UD because "there is no attempt at segmenting words into morphemes".[2]

Open compounds, which are spelled as two (or more) separate words, are treated as two (or more) items that are arranged into a subtree with the head component as the root and the less prominent item(s) as dependent node(s). The relation between the head and the other component is labeled with the dedicated syntactic relation `compound`. This relation is assigned to open compounds regardless of the semantic relation between the components (cf. *apple pie* = "pie made from apples" vs. *coffee cup* = "cup for coffee" vs. *water mill* = "mill powered by water", etc.). Besides the bare `compound` relation, there are 22 subtypes of this relation intended for language-specific phenomena,[3] of which only `compound:prt` is used in some languages under analysis, namely in English and German. The `compound:prt` is used for "[p]article verbs where the particle is realized as a separate word (which may alternate with affixed particles), for example Swedish *byta ut* ('exchange'; cf. *utbytt* 'exchanged')".

Hyphenated compounds are treated in the same way as in open compounds. The hyphen is attached to the head, with the relation label `punct`.[4]

Annotation of compounds is explored for English, Czech, Russian, German, and Latin. In each language, it is performed based on all treebanks available in the UD collection (i.e. ten treebanks for English with a total of 46K sentences, four German treebanks containing 208K sentences, six treebanks for Czech with 208K sentences, five Latin treebanks with 59K sentences, and five treebanks for Russian with 111K sentences). The language set is different from the language set pertaining to the rest of this thesis, because the task at hand requires much more familiarity with the given languages, and the author's knowledge of Spanish, French, and Dutch

[2] https://universaldependencies.org/u/overview/tokenization.html

[3] https://universaldependencies.org/ext-dep-index.html

[4] This is the case for the languages in scope, but the claim does not hold for all languages in UD. Swedish hyphenated compounds are for instance handled the same way as closed compounds.

Lang	compound relations	Sentences w/ compound	compound:prt relations	Sentences w/ compound:prt	Total words	Total sent.
English	22K (3,0%)	13.5K (29.3%)	2.5K (0.3%)	2.3 (5.0%)	726K	46K
German	1.8K (0.1%)	1.4K (0.7%)	22.4K (0.6%)	21.8K (10.5%)	3.8K	208K
Czech	2.7K (0.1%)	1,3K (1.1%)	0 (0.00%)	0 (0.0%)	2.2K	128K
Latin	85 (0.01%)	82 (0.1%)	0 (0.00%)	0 (0.0%)	983K	59K
Russian	1.9K (0.1%)	1,8K (1.6%)	0 (0.00%)	0 (0,0%)	1.8K	111K

Table 5.2: The number of sentences containing a compound or compound:prt relation.

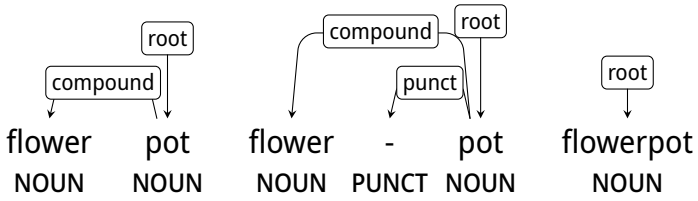
was insufficient for this purpose.

The number of sentences containing the compound relation in the languages' UD treebanks is listed in Table 5.2. The compound:prt relation is used only in English and German; it will not be further commented upon.

The UD treebanks for English

Out of the languages analyzed, English treebanks contain the highest number of compound relations, both in absolute numbers and in percentages, owing to the fact that in this language, $[N + N]$ sequences are analyzed as compounds. English is also a language where these $[N + N]$ compounds can alternatively be spelled with a hyphen or even without a space as a single graphical word (cf, Table 5.3), resulting in different tree structures; cf. the textbook example *flower pot* as an open compound with the hyphenated (*flower-pot*) and closed spelling alternative (*flowerpot*) annotated in line with the UD guidelines in ex. (95).

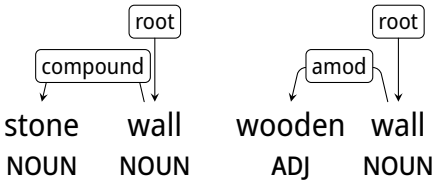
(95)



The compound relation is also assigned to NOUN+ADJ phrases (*emerald green*, *labour intensive*), as well as complex open numerals such as *twenty one*.

Even though the relationship between the components of the open compound *stone wall*, which can be paraphrased as "wall of stones", is the same as the relationship between the adjective *wooden* and the noun *wall* ("wall of wood"), the syntactic relations within these sequences are labeled differently, namely compound in the first sequence while amod in the second; cf. ex. (96).

(96)



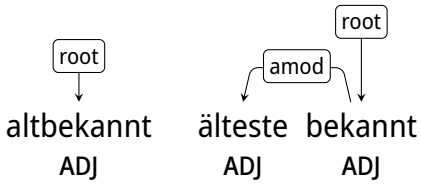
If there were an adjective to the noun *stone* (**stonen*) or if *stone* were considered also as an adjective in English, the annotation would have been no different from *wooden wall*. This is encountered in the phrase *west side*, where *west* is interpreted as an adjective (while the formally identical noun *west* and the formally different adjective *western* exist) and therefore handled as an adjectival modifier (amod) of the nominal governor.

The UD treebanks for German

German is a language where compounding is widely used, but compounds are typically spelled as compact strings. Nevertheless, both hyphenated compounds (cf. the Anglicism *Trackpad-Click*) and open compounds ($[N + N]$ sequences, often with proper names; e.g. *Präsident Franjo* 'President Franjo') are documented in the treebanks, both types assigned the compound relation.

In German we also find cases of (here, closed) compounds with the components' relations analogous to those between words in syntactic phrases, but these analogies are not obvious in the current annotation; cf. the compound *altbekannt* 'well-known', which is represented by a single node, and the phrase *älteste bekannt* 'oldest known', which is represented as a tree headed by the second word with the first element linked by the amod relation in ex. (97).

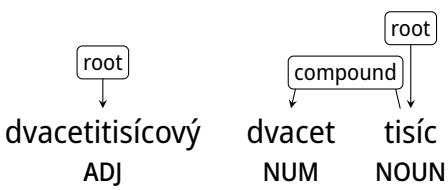
(97)



The UD treebanks for Czech

Also in Czech, compounds are commonly written as continuous strings, still a hyphen may connect the components in coordinate compounds. In the data, however, the compound relation appears not only with hyphenated compounds (*indo-australský* 'Indo-Australian'), but also with numeral expressions, which in Czech are separated by spaces.[5] The rightmost component is taken as the head and the other parts are depending on it as modifiers; cf. the right structure in ex. (98). When a numeral construction enters derivation, the output is a closed compound and it is represented by a single node; cf. the adjective *dvacetitisícový* 'twenty-thousand' on the left in ex.(98) which is traced back to the phrase *dvacet tisíc* 'twenty thousand'.

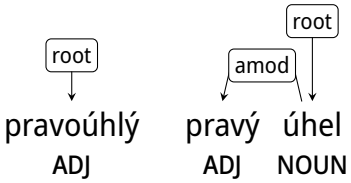
(98)



[5]The interpretation of numerals as compounds, though, does not conform to the Czech linguistic tradition.

Similarly, nouns modified by adjectival modifiers can give rise to adjectives with two roots and closed spelling. Cf. the noun phrase *pravý úhel* 'right angle' and the adjectival compound *pravoúhlý* 'right-angled' in ex. (99), which is close to the German adjective *blauäugig* 'blue-eyed' mentioned in the introductory section in that the right component does not exist as a separate adjective (**úhlý* 'angled' similar to **äugig* '*eyed').

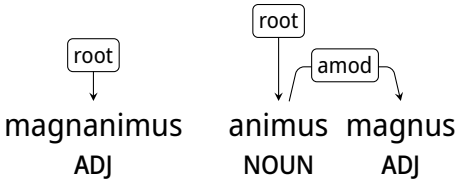
(99)



The UD treebanks for Latin

Latin treebanks contain the lowest number of compound relations, as documented in Table 5.2. Its current usage is limited to numeral expressions if they are spelled as separate words in a way described above for Czech, with the addendum that sometimes one of the words is *unus* 'one' labeled as a determiner and not a numeral. Example (100) is also analogous to Czech, documenting an adjectival compound (*magnanimus* 'high-spirited') that is based on a noun phrase (here, more specifically, on a phrase with the head noun preceding the adjectival modifier: *animus magnus* lit. 'spirit high' = 'high spirit').

(100)



The UD treebanks for Russian

In the Russian treebanks, the compound relation is – unlike in Czech – applied to "noun compounds (e.g., *стресс менеджмент* 'stress management, *Жар птица* 'Fire bird'), but also adjective compounds (e.g., *бэд блоки* 'bad blocks', *мини колонка* 'mini speaker', *Гранд отель* 'Grand hotel') and some other types ('+ 1', '№ 1')".[6] Such $[N + N]$ compounds and ADJ+NOUN compounds are often loanwords or direct translations of foreign expressions.

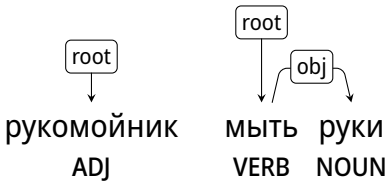
In addition, now similarly to Czech and also Latin, the compound relation appears also with numerals (*две тысячи* 'two thousand') and hyphenated constructions (*город-государство*; 'city-state').

Noteworthy are compounds which are analyzed as $[N + V]$ structures in the Golden Compound Analyses database. Since they are closed compounds, they are currently represented by a single node in the treebanks, but the relationship between the components resembles the obj relation of the object noun to its

[6] https://universaldependencies.org/ru/dep/compound.html

governing verb; cf. *рукомойник* ‘washbasin’ and the phrase *мыть руки* ‘to wash hands’ in ex. (101), or *короед* ‘bark beetle’ traced back to *есть кору* ‘to eat bark’ and *травосеяние* ‘grass sowing’ related to *сеять траву* ‘to sow grass’.

(101)



5.2 Syntax-based annotation of compounds

As we have tried to show, the current annotation does not allow to get a complex picture of compounds (as multi-root items) either within one language or across languages. On the one hand, the compound relation only applies to open and hyphenated compounds while closed compounds are not marked in any way. On the other hand, the compound relation is underspecified, without capturing the different relations observed between the components in individual compounds – the exact same label is used for English $[N + N]$ compounds, which themselves document a variety of internal relationships, and for relations between numerals in Czech, for example.

5.2.1 Covering all compound types

We now roughly outline a preliminary proposal for a new annotation of compounds in UD that should overcome these issues. Rather than offering an ultimate solution to each individual aspect of compound annotation, we present in our proposal one or more possible solutions based on what we have encountered in the literature or in existing language resources, with our primary goal being to initiate a discussion on this topic.

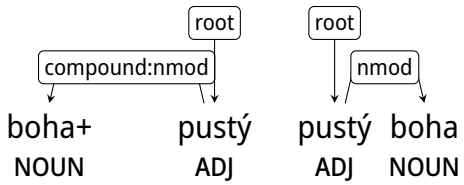
Compounds with all types of spelling should be approached as complex structures that consist of components which are linked by a relationship that is often similar to syntactic relations between words in syntactic phrases:

(a) Closed compounds should be split into their respective constituents for this purpose, and further handled in the same manner as open and hyphenated compounds. Compounds with three and more components will be divided into individual parts (e.g. the above German example *Umfrageteilnehmer* ‘survey participant’ into *Umfrage*+ *Teil*+ *Nehmer*) and their relationships will be captured by arranging them into a tree structure (see the next points). As illustrated, in closed compounds a “+” sign may be used on the first (or on all non-final) components to indicate the original morphological boundary, so that the information on their orthography is retained. An interfix, if contained in a compound, will be part of the preceding component (cf. *Umgangssprache* ‘colloquial language’ as *Umgangs*+ *Sprache*).

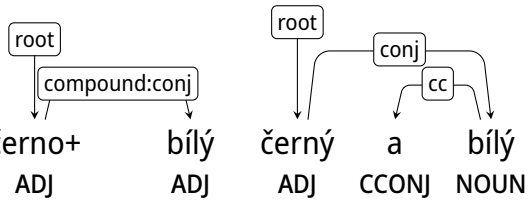
(b) Since such an approach would yield strings that do not exist as separate words (cf. **Abbiege* in *Abbiegeassistent*), we propose – in accordance with the fact that the words in syntactic phrases are treated in this way – to assign a lemma to each component. It can be a full word that is identical with the component (i.e. *Umgang* 'dealing' or *umgehen* 'to deal' and *Sprache* 'language' for *Umgangs+ Sprache*) or close to it (*abbiegen* 'to turn' and *Assistent* 'assistant' for *Abbiege+ Assistent*). Derivatives of compounds would share this lemmatization with their ancestors, e.g. *domorodec* 'native man' would be lemmatized as *domo+ rodý* 'native' (i.e. *dům* 'house' and *rod* 'kin').

(c) All types of compounds should be organized into subtrees in a way analogous to syntactic phrases in UD, making a distinction between subordinate compounds (with the compound's head as the governor and its modifier as its dependent; cf. *bohapustý* 'godless' in ex. (102) and coordinate compounds (with the first component as the root of the subtree and all the other conjuncts depending on it; cf. *černobílý* 'black-and-white' in ex. (103)).

(102)



(103)



(d) Though the subtree modeling the syntactic structure of a compound's components is proposed to be as close an analogy as possible to the subtrees of syntactic phrases, the relation may retain the compound/phrase distinction. As bare `compound` relations are not informative, the relations within compounds could be tagged with a `compound:<relation>` label, where `<relation>` is an already-existing UD syntactic relation. This restriction regarding forcing compound subtypes into established relations should pertain solely to a) currently bare `compound` relations and b) closed compounds currently treated as atomic units, **not** to established, already-subtyped relations such as the `compound:prt` mentioned in Section 5.1.1. These should not be overwritten, their further usage is neither blocked nor discouraged by our proposal.

How these individual pieces of annotation could be brought into the data is discussed in the next section.

5.2.2 Towards the proposed annotation

1. **Identification of closed compounds.** To get a preliminary idea of which part of the treebank data for individual languages would be affected by the

Language	Closed compounds	Total words	Sentences w/ cl comp	Total sent
English	5,934 (0.82%)	726K	5,286 (11.57%)	46K
German	156,629 (4.11%)	3,810K	87,104 (50.14%)	208K
Czech	47,103 (2.11%)	2,222K	34,775 (27.27%)	128K
Latin	26,271 (2.62%)	983K	18,353 (31.27%)	59K
Russian	4,803 (0.27%)	1,830K	4,460 (4.00%)	111K

Table 5.3: A lower bound estimate of the amount of closed compounds (tokens) in Universal Dependencies.

proposed annotation, the number of closed compounds in the UD treebanks needs to be estimated in addition to the number of the compound relations (which are in Table 5.2). In this study, we used the lists of compounds contained in the language resources discussed above in Section 2. The figures in Table 5.3 are heavily conditioned by the size of the resources used. The figures represent a lower bound for the actual amount of closed compounds contained in UD, since none of the data sources list the compounds from their respective languages exhaustively. They are based on searching for the known compounds (and their derivatives) extracted from the data sources listed in Table 5.1

With these limitations in mind, Table 5.3 suggests that the influence of such a change would be substantial, especially in German, where more than 156 thousand closed compounds were identified, which are part of 87 thousand sentences (i.e. 50% of all sentences). The least affected language by our current estimate would be Russian with less than 5 thousand closed compounds distributed over 4 thousand (4%) sentences; this is due to the relatively low coverage of the Golden Compound Analyses database used as the Russian compound data source in this study (see Table 5.1). The utilization of resources with higher coverage or another more sophisticated approach could render these numbers substantially higher.

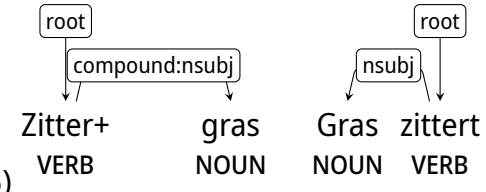
The gap between the percentage of words that are closed compounds and the percentage of trees that contain them is caused by the fact that many often used words happen to be closed compounds. Taking Latin, where this pheonemon is pronounced, as an example, *Word Formation Latin* (Litta et al., 2016) contains compound words such as *horsum* 'hither; this way' from *hic* 'this' *versum* 'side' or *quasi* 'as if' from *qui* 'as' and *si* 'if'. Similarly, *CELEX* contains *draußen* 'out there' from *da* 'there' and *außen* 'außen' or *woher* 'where from' from *wo* 'where' and *her* 'here'.

2. For **splitting of compounds and lemmatization of the components**, the language data sources reviewed above can be taken as a starting point, because they contain high-quality, linguistically adequate material. Whereas CELEX both divides the compounds into substrings and assigns representative forms to its individual parts (cf. *geh* for *gang* above), the other resources provide full-fledged ancestors for compounds that would fit our idea of components' lemmas. Even if the resources for some languages are limited, the existing data can – after unifying the annotation according to the proposal – be used for training automatic tools. *PaReNT* (Svoboda and Ševčíková,

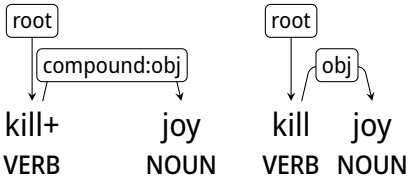
2022), performs both compound splitting and component lemmatization with decent results on Czech.

3. **Specifying the syntactic structure and assigning syntactic relation labels** is another important step for which existing sources provide only very limited data (cf. the bracketed structure in CELEX). Since the pilot manual annotation was based around a mostly mechanical process of finding compound-associated phrases, feeding them into UDPipe (Straka et al., 2016), and observing the relation within the phrase, a semi-automatic procedure is being developed that follows this approach. For example, the German compound *Zittergras* 'quaking-grass' encodes the phrase *das Gras zittert*. The syntactic annotation provided for this phrase by UDPipe is then replicated in the compound, cf. the structures of the compound and of the underlying phrase both with *Gras* as `nsubj` in ex.(104). The English example *killjoy* with the `obj` relation follows in ex. (105).

(104)



(105)



In addition to the examples provided in this section (ex. (102) through ex. (105)), the envisioned annotation scheme is applied to the examples that were presented above in Section 5.1 – see Table 5.4, where the annotation according to the current UD guidelines is shown on the left-hand side and the proposed annotation on the right.

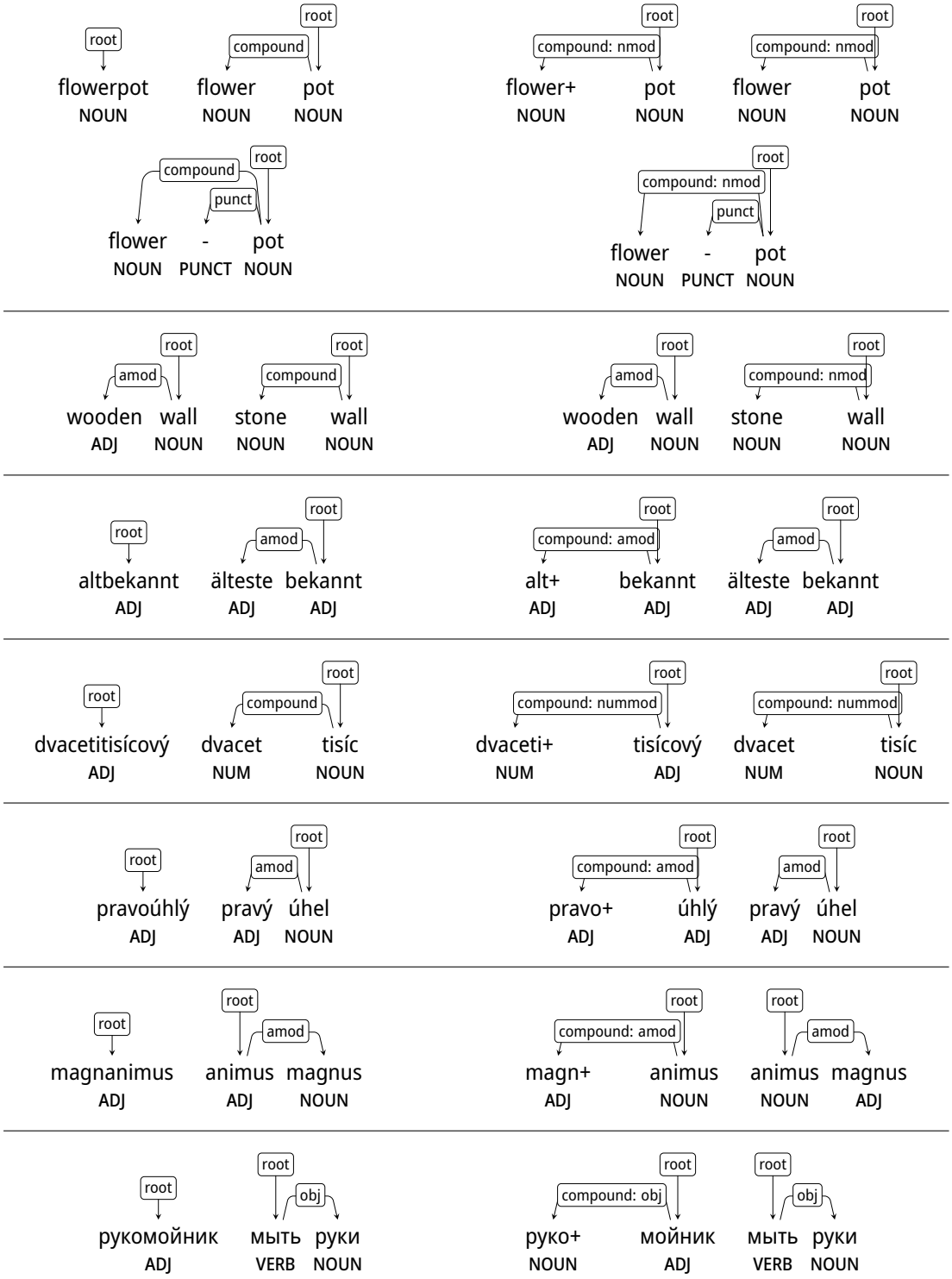


Table 5.4: A comparison between how compounds are handled currently in UD (left) and how they would be handled by the proposed annotation scheme (right).

6. Conclusion

In this dissertation, we described and evaluated the development of a series of three tools – *Czech Compound Splitter*, *Word Formation Analyzer for Czech* and *PaReNT* – capable of static modeling of compounds, meaning that the tool accepts a compound word as input and returns its parents. We then offer a practical demonstration how such a tool can be used to enrich DeriNet, and by extension allow the study of compounding in the context of other word-formation processes such as derivation. Finally, we also show how, in the future, compounding can additionally be studied not only in the context of word formation, but also in the context of syntax, by surveying the way compounding is handled at the moment in Universal Dependencies, and proposing a way to handle them consistently.

Czech Compound Splitter is the result of the first attempts to systematically identify and split Czech compounds. While there has been a lot of attention invested into automatic compound splitting in languages such as German or Sanskrit, in the Slavic languages, the topic had largely, though not completely, been overlooked. We have attempted to tackle the problem using three approaches – one that uses simple heuristics, another based on an asymmetric word similarity metric based on finding the shortest path through a matrix of phonological similarity, and another utilizing a deep learning model partially trained on synthetic data, using the Marian Translation framework. Despite a surprising amount of irregularity and difficulty in Czech compounds, the *Czech Compound Splitter* tool achieved an accuracy of 54% in the task of compound splitting and an accuracy of 84% in compound identification.

Word Formation Analyzer for Czech (*WFA.ces*) is an extension of *CCS* into derivation, the result being a computational tool capable of parent retrieval and word formation classification. It is based around an ensemble of deep-learning models built using the *Marian* framework, equipped with output analysis and reranking. It is able to perform word formation classification with 87% balanced accuracy, specifically excelling in discerning compounds from non-compounds, in which it achieves an F1-score of 94%, and parent retrieval with 71% accuracy, as measured on a separate data set. It outperforms its predecessor, *Czech Compound Splitter*, in every regard.

As the final entry in the series, we presented *PaReNT* (Parent Retrieval Neural Tool). It is an extension of *WFA.ces* into six more languages – English, German, Dutch, Russian, Spanish, and French. Unlike its predecessors built within the Marian framework, which are character-based only, *PaReNT* is based on the Tensorflow framework and runs on a dual-representation input, custom-designed encoder/decoder with a classification module RNN neural network coupled with a Transformer-like block and two output modules, one for parent retrieval and one for word formation classification. *PaReNT* has achieved a total Accuracy of 0.62 in parent retrieval and a Balanced Accuracy of 0.74 in word-formation classification on an independent validation data set. *PaReNT* is now freely available as both a command-line tool and an importable Python package.[1]

To demonstrate the usage of *PaReNT* in enriching data sources, we used it to detect and retrieve nearly $4,400$ compounds – increasing the original amount of compounds threefold – and with the help of a manual check, map these compounds to their parents as part of the DeriNet word-formation network, releasing the

[1] https://github.com/iml-r/PaReNT

version 2.2 as a result. In order to perform this mapping, numerals, pronouns, and appositions had to be added into DeriNet, since they previously weren't present and many compounds have these as their parents. As part of the update, we also revisited the annotation scheme that had been originally implemented in DeriNet, and on which *PaReNT* had been trained on, and developed an improved one, based around the idea that compound words encode phrases. The new annotation scheme is based around mapping compounds to the content words contained in these phrases. The scheme is designed to easily carry over into the other languages in scope. In addition, it can handle neoclassical compounds by mapping them to neoclassical constituents, which are assumed to be shared among the languages in scope, and their handling is designed with this fact in mind. Using DeriNet 2.2 released as part of this thesis[2], we presented some statistical findings regarding the productivity of various compound types with respect to their input and output parts-of-speech ($[POS_1 + POS_2]_{POS_3}$), as well as examining the derivative productivity of the various compound types and the derivation height of the parents of the various compound types.

In the last chapter of this dissertation, we explored the current treatment of compounds in Universal Derivations in five languages. Compounding in many cases straddles the line between word formation and syntax and the exact boundary between the two is sometimes dependent on linguistic tradition, which is in turn often dependent on a specific language or language area. We proposed an annotation scheme that handles closed, hyphenated and open compounds in the same vein. The proposal approaches compounds analogously to how Universal Derivations currently handle syntax. We observed that the handling of open and hyphenated compounds varies widely according to the particular language in question, and that closed compounds are taken into account in none of them. Based on these observations and also the long-standing tradition of describing compounds from a syntactic perspective present in the linguistic literature, the objective of the paper was to open a discussion on whether a multilingual annotation scheme for compounds in Universal Dependencies that employs the dependency relations already in use is useful and what features it should have.

In conclusion, this dissertation contributes to the study of compounding by releasing a multi-lingual model of word formation capable of compound analysis, showing how its usage can help the enrichment of existing data resources in order to study compounds in the context of other word-formation processes, and paving the way to systematically study compounds in the context of syntax, across languages, by proposing an annotation scheme that captures compounds in Universal Dependencies.

[2] https://lindat.mff.cuni.cz/repository/xmlui/handle/11234/1-5538

Bibliography

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2015. TensorFlow: Large-scale Machine Learning on Heterogeneous Systems. Software available from tensorflow.org.

Peter Ackema and Ad Neeleman. 2010. *The Role of Syntax And Morphology in Compounding*, pages 21–36. John Benjamins Pub. Co.

Valerie Adams. 1977. *An Introduction to Modern English Word-formation*, 1st edition. Routledge.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer Normalization. *arXiv preprint arXiv:1607.06450*.

RH Baayen, R Piepenbrock, and L Gulikers. 2014. CELEX2 LDC96L14, 1995. *URL https://doi. org/10*, 35111.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural Machine Translation by Jointly Learning to Align And Translate. *arXiv preprint arXiv:1409.0473*.

Charles Bally. 1944. *Linguistique Générale Et Linguistique Francaise*. A. Francke, Tübingen.

Khuyagbaatar Batsuren, Gábor Bella, Aryaman Arora, Viktor Martinovic, Kyle Gorman, Zdeněk Žabokrtský, Amarsanaa Ganbold, Šárka Dohnalová, Magda Ševčíková, Kateřina Pelegrinová, Fausto Giunchiglia, Ryan Cotterell, and Ekaterina Vylomova. 2022a. The SIGMORPHON 2022 Shared Task on Morpheme Segmentation. In *Proceedings of the 19th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 103–116, Seattle, Washington. Association for Computational Linguistics.

Khuyagbaatar Batsuren, Omer Goldman, Salam Khalifa, Nizar Habash, Witold Kieraś, Gábor Bella, Brian Leonard, Garrett Nicolai, Kyle Gorman, Yustinus Ghanggo Ate, Maria Ryskina, Sabrina Mielke, Elena Budianskaya, Charbel El-Khaissi, Tiago Pimentel, Michael Gasser, William Abbott Lane, Mohit Raj, Matt Coler, Jaime Rafael Montoya Samame, Delio Siticonatzi Camaiteri, Esaú Zumaeta Rojas, Didier López Francis, Arturo Oncevay, Juan López Bautista, Gema Celeste Silva Villegas, Lucas Torroba Hennigen, Adam Ek, David Guriel, Peter Dirix, Jean-Philippe Bernardy, Andrey Scherbakov, Aziyana Bayyr-ool, Antonios Anastasopoulos, Roberto Zariquiey, Karina Sheifer, Sofya Ganieva, Hilaria Cruz, Ritván Karahóǧa, Stella Markantonatou, George Pavlidis, Matvey Plugaryov,

Elena Klyachko, Ali Salehi, Candy Angulo, Jatayu Baxi, Andrew Krizhanovsky, Natalia Krizhanovskaya, Elizabeth Salesky, Clara Vania, Sardana Ivanova, Jennifer White, Rowan Hall Maudslay, Josef Valvoda, Ran Zmigrod, Paula Czarnowska, Irene Nikkarinen, Aelita Salchak, Brijesh Bhatt, Christopher Straughn, Zoey Liu, Jonathan North Washington, Yuval Pinter, Duygu Ataman, Marcin Wolinski, Totok Suhardijanto, Anna Yablonskaya, Niklas Stoehr, Hossep Dolatian, Zahroh Nuriah, Shyam Ratan, Francis M. Tyers, Edoardo M. Ponti, Grant Aiton, Aryaman Arora, Richard J. Hatcher, Ritesh Kumar, Jeremiah Young, Daria Rodionova, Anastasia Yemelina, Taras Andrushko, Igor Marchenko, Polina Mashkovtseva, Alexandra Serova, Emily Prud'hommeaux, Maria Nepomniashchaya, Fausto Giunchiglia, Eleanor Chodroff, Mans Hulden, Miikka Silfverberg, Arya D. McCarthy, David Yarowsky, Ryan Cotterell, Reut Tsarfaty, and Ekaterina Vylomova. 2022b. UniMorph 4.0: Universal Morphology. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 840–855, Marseille, France. European Language Resources Association.

Laurie Bauer. 1998. Is There a Class of Neoclassical Compounds, And If So Is It Productive? *Linguistics*, 36(3):403–422.

Laurie Bauer. 2001. Compounding. In M. Haspelmath, editor, *Language Typology and Language Universals: An International Handbook*, pages 695–707. De Gruyter.

Laurie Bauer. 2019. Compounds And Multi-word Expressions in English. In Barbara Schlücker, editor, *Complex Lexical Units. Compounds and Multi-Word Expressions*, pages 45–68. De Gruyter, Berlin.

Marenglen Biba and Eva Gjati. 2014. Boosting Text Classification Through Stemming of Composite Words. In *Recent Advances in Intelligent Informatics*, pages 185–194. Springer.

Antonietta Bisetto and Chiara Melloni. 2008. Parasynthetic Compounding. *Lingue e linguaggio*, 7(2):233–260.

Antonietta Bisetto and Sergio Scalise. 2005. The Classification of Compounds. *Lingue e linguaggio*, 4(2):319–332.

Leonard Bloomfield. 1933. *Language*. A. Francke.

Geert Booij. 2005. Compounding and Derivation: Evidence for Construction Morphology. *Amsterdam Studies in the Theory and History of Linguistic Science Series 4*, 264:109–132.

Geert Booij. 2010. Construction Morphology. *Language and linguistics compass*, 4(7):543–555.

Geert Booij. 2019. Compounds And Multi-word Expressions in Dutch. In Barbara Schlücker, editor, *Complex Lexical Units. Compounds and Multi-Word Expressions*, pages 95–126. De Gruyter, Berlin.

Ivana Bozděchová. 1997. *Tvoření slov skládáním*. Institut sociálních vztahů, Praha.

Johan Carlberger, Hercules Dalianis, Martin Duneld, and Ola Knutsson. 2001. Improving Precision in Information Retrieval for Swedish Using Stemming. In *Proceedings of the 13th Nordic Conference of Computational Linguistics (NODALIDA 2001)*, pages 17–22.

Petr Chmelař, David Hellebrand, Michal Hrušecký, and Vladimír Bartík. 2011. Nalezení slovních kořenů v češtině. In *Znalosti 2011: Sborník příspěvků 10. ročníku konference*, pages 66–77. VŠB-Technical University of Ostrava.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning Phrase Representations Using RNN Encoder–Decoder for Statistical Machine Translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar. Association for Computational Linguistics.

Elizaveta L. Clouet and Béatrice Daille. 2014. Splitting of Compound Terms in Non-Prototypical Compounding Languages. In *Workshop on Computational Approaches to Compound Analysis*, pages 11–19.

Ryan Cotterell, Arun Kumar, and Hinrich Schütze. 2016. Morphological Segmentation Inside-out. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2325–2330, Austin, Texas. Association for Computational Linguistics.

Mathias Creutz and Krista Lagus. 2002. Unsupervised Discovery of Morphemes. *arXiv preprint cs/0205057*.

George Cybenko. 1989. Approximation by Superpositions of a Sigmoidal Function. *Mathematics of Control, Signals and Systems*, 2(4):303–314.

Marie-Catherine de Marneffe, Christopher D. Manning, Joakim Nivre, and Daniel Zeman. 2021. Universal Dependencies. *Computational Linguistics*, 47(2):255–308.

Ljiljana Dolamic and Jacques Savoy. 2009. Indexing And Stemming Approaches for the Czech Language. *Information Processing & Management*, 45(6):714–720.

Nigel Fabb. 1998. *Compounding*, pages 66–83. John Wiley & Sons, Inc.

Rita Finkbeiner and Barbara Schlücker. 2019. Compounds And Multi-word Expressions in the Languages of Europe. In Barbara Schlücker, editor, *Complex Lexical Units. Compounds and Multi-Word Expressions*, pages 1–43. De Gruyter, Berlin.

Fabienne Fritzinger and Alexander Fraser. 2010. How to Avoid Burning Ducks: Combining Linguistic Analysis And Corpus Statistics for German Compound Processing. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*, pages 224–234, Uppsala, Sweden. Association for Computational Linguistics.

Livio Gaeta, Davide Ricca, et al. 2009. Composita Solvantur: Compounds as Lexical Units or Morphological Objects? *Rivista di Linguistica*, 21(1):35–70.

Stig-Arne Grönroos, Sami Virpioja, and Mikko Kurimo. 2018. Cognate-aware Morphological Segmentation for Multilingual Neural Translation. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 386–393, Belgium, Brussels. Association for Computational Linguistics.

Emiliano Guevara, Sergio Scalise, Antonietta Bisetto, and Chiara Melloni. 2006. Morbo/Comp: a Multilingual Database of Compound Words. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)*, Genoa, Italy. European Language Resources Association (ELRA).

Jan Hajič, Jaroslava Hlaváčová, Marie Mikulová, Milan Straka, and Barbora Štěpánková. 2020. MorfFlex CZ 2.0. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (
'UFAL), Faculty of Mathematics and Physics, Charles University.

Birgit Hamp and Helmut Feldweg. 1997a. GermaNet – a lexical-semantic net for German. In *Automatic Information Extraction and Building of Lexical Semantic Resources for NLP Applications*, pages 9–15.

Birgit Hamp and Helmut Feldweg. 1997b. germanet – a Lexical-semantic Net for German. In *Automatic information extraction and building of lexical semantic resources for NLP applications*, pages 9–15.

Heidi Harley. 2011. Compounding in Distributed Morphology. In *The Oxford Handbook of Compounding*, pages 129–141. Oxford University Press, Oxford.

Peter E. Hart, Nils J. Nilsson, and Bertram Raphael. 1968. A Formal Basis for the Heuristic Determination of Minimum Cost Paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2):100–107.

Martin Haspelmath. 2002. *Understanding Morphology*. Arnold, London.

Martin Haspelmath. 2017. The Indeterminacy of Word Segmentation And the Nature of Morphology And Syntax. *Folia Linguistica*, 51:31–80.

Martin Haspelmath and Andrea Sims. 2013. *Understanding Morphology*. Routledge, Abingdon.

Benjamin Heinzerling and Michael Strube. 2017. BPEMB: Tokenization-free Pretrained Subword Embeddings in 275 Languages. *arXiv preprint arXiv:1710.02187*.

Oliver Hellwig and Sebastian Nehrdich. 2018. Sanskrit Word Segmentation Using Character-level Recurrent And Convolutional Neural Networks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2754–2763.

Verena Henrich and Erhard Hinrichs. 2010. GernEdit – the Germanet Editing Tool. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, pages 2228–2235.

Verena Henrich and Erhard Hinrichs. 2011. Determining Immediate Constituents of Compounds in GermaNet. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing 2011*, pages 420–426.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural computation*, 9(8):1735–1780.

Kurt Hornik. 1991. Approximation Capabilities of Multilayer Feedforward Networks. *Neural networks*, 4(2):251–257.

Gérard Huet. 2005. A Functional Toolkit for Morphological And Phonological Processing, Application to a Sanskrit Tagger. *Journal of Functional Programming*, 15(4):573–614.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. 2018. Marian: Fast Neural Machine Translation in c++. In *Proceedings of ACL 2018, System Demonstrations*, pages 116–121.

M.R. Kale. 1931. *Higher Sanskrit Grammar*, 7th edition. Open Source.

F. Katamba. 1993. *Morphology*. Modern linguistics series. Macmillan.

Sanjeet Khaitan, Arumay Das, Sandeep Gain, and Adithi Sampath. 2009. Data-driven Compound Splitting Method for English Compounds in Domain Names. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, CIKM '09, page 207–214, New York, NY, USA. Association for Computing Machinery.

Diederik P Kingma and Jimmy Ba. 2014. ADAM: A Method for Stochastic Optimization. *arXiv preprint arXiv:1412.6980*.

Oskar Kohonen, Sami Virpioja, and Krista Lagus. 2010. Semi-Supervised Learning of Concatenative Morphology. In *Proceedings of the 11th Meeting of the ACL Special Interest Group on Computational Morphology and Phonology*, pages 78–86.

Michal Křen, Václav Cvrček, Tomáš Čapka, Anna Čermáková, Milena Hnátková, Lucie Chlumská, Tomáš Jelínek, Dominika Kováříková, Vladimír Petkevič, Pavel Procházka, Hana Skoumalová, Michal Škrabal, Petr Truneček, Pavel Vondřička, and Adrian Jan Zasina. 2016. SYN2015: Representative Corpus of Contemporary Written Czech. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 2522–2528.

Irina Krotova, Sergey Aksenov, and Ekaterina Artemova. 2020. A Joint Approach to Compound Splitting And Idiomatic Compound Detection. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 4410–4417, Marseille, France. European Language Resources Association.

Lukáš Kyjánek, Zdeněk Žabokrtský, Jonáš Vidra, and Magda Ševčíková. 2021. Universal Derivations V1.1. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Vladimir I. Levenshtein. 1966. Binary Codes Capable of Correcting Deletions, Insertions and Reversals. *Soviet Physics-Doklady*, 10(8):707–710.

Rochelle Lieber. 2004. *Morphology and Lexical Semantics*, volume 104. Cambridge University Press, Cambridge.

Rochelle Lieber and Pavol Štekauer. 2009. *The Oxford Handbook of Compounding*. Oxford Handbooks Series. OUP Oxford.

Rochelle Lieber and Pavol Štekauer. 2011. *The Oxford Handbook of Compounding*, first published in paperback 2011 edition. Oxford handbooks in linguistics. Oxford University Press, Oxford.

Eleonora Litta, Marco Passarotti, and Chris Culy. 2016. Formatio Formosa Est. Building a Word Formation Lexicon for Latin. In *Proceedings of the 3rd Italian Conference on Computational Linguistics*, pages 185–189.

Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective Approaches to Attention-based Neural Machine Translation. *arXiv preprint arXiv:1508.04025*.

Jianqiang Ma, Verena Henrich, and Erhard Hinrichs. 2016. Letter Sequence Labeling for Compound Splitting. In *Proceedings of the 14th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 76–81.

Hugo Mailhot, Maximiliano A Wilson, Joël Macoir, S Hélène Deacon, and Claudia Sánchez-Gutiérrez. 2020. Morpholex-FR: A Derivational Morphological Database for 38,840 French Words. *Behavior research methods*, 52:1008–1025.

Shie Mannor, Dori Peleg, and Reuven Rubinstein. 2005. The Cross Entropy Method for Classification. In *Proceedings of the 22nd international conference on Machine learning*, pages 561–568.

Hans Marchand. 1960. *The Categories And Types of Present-day English Word-formation : A Synchronic-Diachronic Approach*. Harrassowitz, Wiesbaden.

Hans Marchand. 1967. Expansion, Transposition, And Derivation. *La Linguistique*, 3(1):13–26.

Hans Marchand. 1969. *The Categories And Types of Present Day English Word Formation*. Verlag C.H.Beck, Munich.

Chris Biemann Martin Riedl. 2016. Unsupervised Compound Splitting With Distributional Semantics Rivals Supervised Methods. In *Proceedings of The 15th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologie*, pages 617–622, San Diego, CA, USA.

Chiara Melloni. 2023. Neoclassical word formation. In Peter Ackema, Sabrina Bendjaballah, Eulàlia Bonet, and Antonio Fábregas, editors, *The Wiley Blackwell Companion to Morphology*, pages 1–33. John Wiley & Sons, Ltd, Chichester, UK.

Fiammetta Namer. 2003. Automatiser L'analyse Morpho-sémantique Non Affixale: Le Système Dérif. *Cahiers de grammaire*, 28:31–48.

Yurii Nesterov. 1983. A Method for Solving the Convex Programming Problem with Convergence Rate o(1/k2). In *Dokl Akad Nauk SSSR*, volume 269, page 543.

Diarmuid Ó Séaghdha. 2008. Learning Compound Noun Semantics. Technical Report UCAM-CL-TR-735, University of Cambridge, Computer Laboratory.

Martin Ološtiak and Marta Vojteková. 2021. Kompozitnosť a kompozícia: príspevok k charakteristike zložených slov na materiáli západoslovanských jazykov. *Slovo a slovesnost*, 82(2):95–117.

Susan Olsen. 2000. Copulative Compounds: A Closer Look at the Interface Between Syntax And Morphology. In *Yearbook of morphology 2000*, pages 279–320. Springer.

Susan Olsen. 2001. Copulative Compounds: A Closer Look at the Interface Between Syntax And Morphology. In *Yearbook of Morphology 2000*, pages 279–320. Springer.

OpenAI. 2021. ChatGPT.

Karel Pala and Pavel Šmerk. 2015. Derivancze – Derivational Analyzer of Czech. In *International Conference on Text, Speech, and Dialogue*, pages 515–523.

Renáta Panocová and Pius ten Hacken. 2020. *Neoclassical Compounds Between Borrowing And Word Formation*, page 32–48. Edinburgh University Press, Edinburgh.

Boris T Polyak. 1964. Some Methods of Speeding Up the Convergence of Iteration Methods. *USSR Computational Mathematics and Mathematical Physics*, 4(5):1–17.

Martin F Porter. 1980. An Algorithm for Suffix Stripping. *Program: Electronic Library and Information Systems*, 14:130–137.

Martin F. Porter. 2001. Snowball: A Language for Stemming Algorithms. Published online. Accessed 21.01.2022, 15.00h.

Virginia Pulcini. 2019. Internationalisms, Anglo-Latinisms And Other Kinship Ties Between Italian And English. *Studia Linguistica Universitatis Iagellonicae Cracoviensis*, 136(2):121–142.

Pāṇini. 1987. *The Aṣṭādhyāyī of Pāṇini*. Munshiram Manoharlal Publishers, New Delhi. Translation and Commentary.

Jan Radimský. 2015. *Noun+Noun Compounds in Italian*. University of South Bohemia in České Budějovice.

Herbert Robbins and Sutton Monro. 1951. A Stochastic Approximation Method. *The Annals of Mathematical Statistics*, pages 400–407.

David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. 1986. Learning Representations by Back-propagating Errors. *Nature*, 323(6088):533–536.

Sergio Scalise and Antonietta Bisetto. 2009. The classification of compounds. In Rochelle Lieber and Pavol Štekauer, editors, *The Oxford Handbook of Compounding*, pages 49–82. Oxford University Press.

Sergio Scalise and Irene Vogel. 2010. *Cross-disciplinary Issues in Compounding*. John Benjamins.

Barbara Schlücker, editor. 2019a. *Complex Lexical Units. Compounds and Multi-word Expressions*. De Gruyter, Berlin.

Barbara Schlücker. 2019b. Compounds And Multi-word Expressions in German. In Barbara Schlücker, editor, *Complex Lexical Units. Compounds and Multi-Word Expressions*, pages 69–94. De Gruyter, Berlin.

Leslie N Smith. 2017. Cyclical Learning Rates for Training Neural Networks. In *2017 IEEE winter conference on applications of computer vision (WACV)*, pages 464–472. IEEE.

Andrew Spencer. 1991. *Morphological Theory*. Blackwell, Oxford.

Pavel Štichauer. 2013. Je možná nová klasifikace českých kompozit? *Časopis pro moderní filologii*, 95(2):113–128.

Milan Straka, Jan Hajič, and Jana Straková. 2016. UDPipe: Trainable Pipeline for Processing CONLL-U Files Performing Tokenization, Morphological Analysis, Pos Tagging And Parsing. In *Proceedings of the 10th International Conference on Language Resources and Evaluation*, pages 4290–4297.

Kyoko Sugisaki and Don Tuggener. 2018. German Compound Splitting Using the Compound Productivity of Morphemes. In *14th Conference on Natural Language Processing*, pages 141–147.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to Sequence Learning with Neural Networks. *CoRR*, abs/1409.3215.

Emil Svoboda, Tomáš Bořil, Jan Rusz, Tereza Tykalová, Dana Horáková, Charles R.G. Guttmann, Krastan B. Blagoev, Hiroto Hatabu, and Vladimir I. Valtchinov. 2022. Assessing Clinical Utility of Machine Learning And Artificial Intelligence Approaches to Analyze Speech Recordings in Multiple Sclerosis: A Pilot Study. *Computers in Biology and Medicine*, 148:105853.

Emil Svoboda, Andrzej Marciniak, Alfredo Morales Pinzon, Tomáš Bořil, Noriaki Wada, Boyan Alexandrov, Hiroto Hatabu, Charles RG Guttmann, and Vladimir Valtchinov. 2024a. Building And Validating a Digital Health Pipeline for Speech Analyses: Can Smartphones Be Used to Extract Voice Features Correlated with Multiple Sclerosis? *In preparation*.

Emil Svoboda and Magda Ševčíková. 2021. Splitting And Identifying Czech Compounds: A Pilot Study. In *Proceedings of the Third International Workshop on Resources and Tools for Derivational Morphology (DeriMo 2021)*, pages 129–138.

Emil Svoboda and Magda Ševčíková. 2024. Compounds in Universal Dependencies: A Survey in Five European Languages. In *Proceedings of the 6th Workshop on Research in Computational Linguistic Typology and Multilingual NLP*, pages 88–99, St. Julian's, Malta. Association for Computational Linguistics.

Emil Svoboda, Jonáš Vidra, Magda Ševčíková, and Zdeněk Žabokrtský. 2024b. DeriNet 2.2. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Emil Svoboda and Magda Ševčíková. 2022. Word Formation Analyzer for Czech: Automatic Parent Retrieval And Classification of Word Formation Processes. *Prague Bulletin of Mathematical Linguistics*, 118:55–72.

Emil Svoboda and Magda Ševčíková. 2024. PaReNT (Parent Retrieval Neural Tool): A Deep Dive Into Word Formation Across Languages. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 12611–12621, Torino, Italia. ELRA and ICCL.

Pius ten Hacken. 2011. Neoclassical Word Formation in English And the Organization of the Lexicon. In *Selected Papers from the 10th International Conference on Greek Linguistics*, pages 78–88.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. *Advances in neural information processing systems*, 30.

Jonáš Vidra, Zdeněk Žabokrtský, Lukáš Kyjánek, Magda Ševčíková, Šárka Dohnalová, Emil Svoboda, and Jan Bodnár. 2021a. DeriNet 2.1. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Jonáš Vidra, Zdeněk Žabokrtský, Lukáš Kyjánek, Magda Ševčíková, Šárka Dohnalová, Emil Svoboda, and Jan Bodnár. 2021b. DeriNet 2.1.

Jonáš Vidra, Zdeněk Žabokrtský, Magda Ševčíková, and Lukáš Kyjánek. 2019. DeriNet 2.0: Towards an All-in-one Word-formation Resource. In *Proceedings of the 2nd Workshop on Resources and Tools for Derivational Morphology*, pages 81–89. Charles University.

Sami Virpioja, Peter Smit, Stig-Arne Grönroos, and Mikko Kurimo. 2013. Morfessor 2.0: Python Implementation And Extensions for Morfessor Baseline.

Daniil Vodolazsky and Hermann Petrov. 2021. Compound Splitting and Analysis for Russian. *Resources and Tools for Derivational Morphology (DeriMo 2021)*, pages 145–153.

Paul Wexler. 1969. Towards a Structural Definition of 'internationalisms'. *Linguistics*, 7(48):77–92.

Zdeněk Žabokrtský, Niyati Bafna, Jan Bodnár, Lukáš Kyjánek, Emil Svoboda, Magda Ševčíková, and Jonáš Vidra. 2022. Towards Universal Segmentations: UniSegments 1.0. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 1137–1149, Marseille, France. European Language Resources Association.

Britta Zeller, Sebastian Padó, and Jan Šnajder. 2014. Towards Semantic Validation of a Derivational Lexicon. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1728–1739, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.

Daniel Zeman, Joakim Nivre, Mitchell Abrams, Elia Ackermann, Noëmi Aepli, Hamid Aghaei, Željko Agić, Amir Ahmadi, Lars Ahrenberg, Chika Kennedy Ajede, Salih Furkan Akkurt, Gabrielė Aleksandravičiūtė, Ika Alfina, Avner Algom, Khalid Alnajjar, Chiara Alzetta, Erik Andersen, Lene Antonsen, Tatsuya Aoyama, Katya Aplonova, Angelina Aquino, Carolina Aragon, Glyd Aranes, Maria Jesus Aranzabe, Bilge Nas Arıcan, Hórunn Arnardóttir, Gashaw Arutie, Jessica Naraiswari Arwidarasti, Masayuki Asahara, Katla Ásgeirsdóttir, Deniz Baran Aslan, Cengiz Asmazoğlu, Luma Ateyah, Furkan Atmaca, Mohammed Attia, Aitziber Atutxa, Liesbeth Augustinus, Mariana Avelãs, Elena Badmaeva, Keerthana Balasubramani, Miguel Ballesteros, Esha Banerjee, Sebastian Bank, Verginica Barbu Mititelu, Starkaður Barkarson, Rodolfo Basile, Victoria Basmov, Colin Batchelor, John Bauer, Seyyit Talha Bedir, Shabnam Behzad, Kepa Bengoetxea, İbrahim Benli, Yifat Ben Moshe, Gözde Berk, Riyaz Ahmad Bhat, Erica Biagetti, Eckhard Bick, Agnė Bielinskienė, Kristín Bjarnadóttir, Rogier Blokland, Victoria Bobicev, Loïc Boizou, Emanuel Borges Völker, Carl Börstell, Cristina Bosco, Gosse Bouma, Sam Bowman, Adriane Boyd, Anouck Braggaar, António Branco, Kristina Brokaitė, Aljoscha Burchardt, Marisa Campos, Marie Candito, Bernard Caron, Gauthier Caron, Catarina Carvalheiro, Rita Carvalho, Lauren Cassidy, Maria Clara Castro, Sérgio Castro, Tatiana Cavalcanti, Gülşen Cebiroğlu Eryiğit, Flavio Massimiliano Cecchini, Giuseppe G. A. Celano, Slavomír Čéplö, Neslihan Cesur, Savas Cetin, Özlem Çetinoğlu, Fabricio Chalub, Liyanage Chamila, Shweta Chauhan, Ethan Chi, Taishi Chika, Yongseok Cho, Jinho Choi, Jayeol Chun, Juyeon Chung, Alessandra T. Cignarella, Silvie Cinková, Aurélie Collomb, Çağrı Çöltekin, Miriam Connor, Daniela Corbetta, Francisco Costa, Marine Courtin, Mihaela Cristescu, Ingerid Løyning Dale, Philemon Daniel, Elizabeth Davidson, Leonel Figueiredo de Alencar, Mathieu Dehouck, Martina de Laurentiis, Marie-Catherine de Marneffe, Valeria de Paiva, Mehmet Oguz Derin, Elvis de Souza, Arantza Diaz de Ilarraza, Carly Dickerson, Arawinda Dinakaramani, Elisa Di Nuovo, Bamba Dione, Peter Dirix, Kaja Dobrovoljc, Adrian Doyle, Timothy Dozat, Kira Droganova, Puneet Dwivedi, Christian Ebert, Hanne Eckhoff, Masaki Eguchi, Sandra Eiche, Marhaba Eli, Ali Elkahky, Binyam Ephrem, Olga Erina, Tomaž Erjavec, Farah Essaidi, Aline Etienne, Wograine Evelyn, Sidney Facundes, Richárd Farkas, Federica Favero, Jannatul Ferdaousi, Marília Fernanda, Hector Fernandez Alcalde, Amal Fethi, Jennifer Foster, Cláudia Freitas, Kazunori Fujita, Katarína Gajdošová, Daniel Galbraith, Federica Gamba, Marcos Garcia, Moa Gärdenfors, Fabrício Ferraz Gerardi, Kim Gerdes, Luke Gessler, Filip Ginter, Gustavo Godoy, Iakes Goenaga, Koldo Gojenola, Memduh Gökırmak, Yoav Goldberg, Xavier Gómez Guinovart, Berta González Saavedra, Bernadeta Griciūtė, Matias Grioni, Loïc Grobol, Normunds Grūzītis, Bruno Guillaume, Céline Guillot-Barbance, Tunga Güngör, Nizar Habash, Hinrik Hafsteinsson, Jan Hajič, Jan Hajič jr., Mika Hämäläinen, Linh Hà Mỹ, Na-Rae Han, Muhammad Yudistira Hanifmuti, Takahiro Harada, Sam Hardwick, Kim Harris, Dag Haug, Johannes Heinecke, Oliver Hellwig, Felix Hennig, Barbora Hladká, Jaroslava Hlaváčová, Florinel Hociung, Petter Hohle, Marivel Huerta Mendez, Jena Hwang, Takumi Ikeda, Anton Karl Ingason, Radu Ion, Elena Irimia, Ọlájídé

Ishola, Artan Islamaj, Kaoru Ito, Siratun Jannat, Tomáš Jelínek, Apoorva Jha, Katharine Jiang, Anders Johannsen, Hildur Jónsdóttir, Fredrik Jørgensen, Markus Juutinen, Hüner Kaşıkara, Nadezhda Kabaeva, Sylvain Kahane, Hiroshi Kanayama, Jenna Kanerva, Neslihan Kara, Ritván Karahóǧa, Andre Kåsen, Tolga Kayadelen, Sarveswaran Kengatharaiyer, Václava Kettnerová, Jesse Kirchner, Elena Klementieva, Elena Klyachko, Arne Köhn, Abdullatif Köksal, Kamil Kopacewicz, Timo Korkiakangas, Mehmet Köse, Alexey Koshevoy, Natalia Kotsyba, Jolanta Kovalevskaitė, Simon Krek, Parameswari Krishnamurthy, Sandra Kübler, Adrian Kuqi, Oğuzhan Kuyrukçu, Aslı Kuzgun, Sookyoung Kwak, Kris Kyle, Veronika Laippala, Lorenzo Lambertino, Tatiana Lando, Septina Dian Larasati, Alexei Lavrentiev, John Lee, Phương Lê Hồng, Alessandro Lenci, Saran Lertpradit, Herman Leung, Maria Levina, Lauren Levine, Cheuk Ying Li, Josie Li, Keying Li, Yixuan Li, Yuan Li, Kyung-Tae Lim, Bruna Lima Padovani, Yi-Ju Jessica Lin, Krister Lindén, Yang Janet Liu, Nikola Ljubešić, Olga Loginova, Stefano Lusito, Andry Luthfi, Mikko Luukko, Olga Lyashevskaya, Teresa Lynn, Vivien Macketanz, Menel Mahamdi, Jean Maillard, Ilya Makarchuk, Aibek Makazhanov, Michael Mandl, Christopher Manning, Ruli Manurung, Büşra Marşan, Cătălina Mărănduc, David Mareček, Katrin Marheinecke, Stella Markantonatou, Héctor Martínez Alonso, Lorena Martín Rodríguez, André Martins, Cláudia Martins, Jan Mašek, Hiroshi Matsuda, Yuji Matsumoto, Alessandro Mazzei, Ryan McDonald, Sarah McGuinness, Gustavo Mendonça, Tatiana Merzhevich, Niko Miekka, Aaron Miller, Karina Mischenkova, Anna Missilä, Cătălin Mititelu, Maria Mitrofan, Yusuke Miyao, AmirHossein Mojiri Foroushani, Judit Molnár, Amirsaeid Moloodi, Simonetta Montemagni, Amir More, Laura Moreno Romero, Giovanni Moretti, Shinsuke Mori, Tomohiko Morioka, Shigeki Moro, Bjartur Mortensen, Bohdan Moskalevskyi, Kadri Muischnek, Robert Munro, Yugo Murawaki, Kaili Müürisep, Pinkey Nainwani, Mariam Nakhlé, Juan Ignacio Navarro Horñiacek, Anna Nedoluzhko, Gunta Nešpore-Bērzkalne, Manuela Nevaci, Lương Nguyễn Thị, Huyền Nguyễn Thị Minh, Yoshihiro Nikaido, Vitaly Nikolaev, Rattima Nitisaroj, Alireza Nourian, Hanna Nurmi, Stina Ojala, Atul Kr. Ojha, Hulda Óladóttir, Adédayọ̀ Olúòkun, Mai Omura, Emeka Onwuegbuzia, Noam Ordan, Petya Osenova, Robert Östling, Lilja Øvrelid, Şaziye Betül Özateş, Merve Özçelik, Arzucan Özgür, Balkız Öztürk Başaran, Teresa Paccosi, Alessio Palmero Aprosio, Anastasia Panova, Hyunji Hayley Park, Niko Partanen, Elena Pascual, Marco Passarotti, Agnieszka Patejuk, Guilherme Paulino-Passos, Giulia Pedonese, Angelika Peljak-Łapińska, Siyao Peng, Siyao Logan Peng, Rita Pereira, Sílvia Pereira, Cenel-Augusto Perez, Natalia Perkova, Guy Perrier, Slav Petrov, Daria Petrova, Andrea Peverelli, Jason Phelan, Jussi Piitulainen, Yuval Pinter, Clara Pinto, Tommi A Pirinen, Emily Pitler, Magdalena Plamada, Barbara Plank, Thierry Poibeau, Larisa Ponomareva, Martin Popel, Lauma Pretkalniņa, Sophie Prévost, Prokopis Prokopidis, Adam Przepiórkowski, Robert Pugh, Tiina Puolakainen, Sampo Pyysalo, Peng Qi, Andreia Querido, Andriela Rääbis, Alexandre Rademaker, Mizanur Rahoman, Taraka Rama, Loganathan Ramasamy, Joana Ramos, Fam Rashel, Mohammad Sadegh Rasooli, Vinit Ravishankar, Livy Real, Petru Rebeja, Siva Reddy, Mathilde Regnault, Georg Rehm, Arij Riabi, Ivan Riabov, Michael Rießler, Erika Rimkutė, Larissa Rinaldi, Laura Rituma, Putri Rizqiyah, Luisa Rocha, Eiríkur Rögnvaldsson, Ivan Roksandic, Mykhailo Romanenko, Rudolf Rosa, Valentin Roșca, Davide Rovati, Ben Rozonoyer, Olga Rudina, Jack Rueter, Kristján Rúnarsson, Shoval Sadde, Pegah Safari, Aleksi Sahala, Shadi Saleh, Alessio Sa-

lomoni, Tanja Samardžić, Stephanie Samson, Manuela Sanguinetti, Ezgi Sanıyar, Dage Särg, Marta Sartor, Mitsuya Sasaki, Baiba Saulīte, Yanin Sawanakunanon, Shefali Saxena, Kevin Scannell, Salvatore Scarlata, Nathan Schneider, Sebastian Schuster, Lane Schwartz, Djamé Seddah, Wolfgang Seeker, Mojgan Seraji, Syeda Shahzadi, Mo Shen, Atsuko Shimada, Hiroyuki Shirasu, Yana Shishkina, Muh Shohibussirri, Maria Shvedova, Janine Siewert, Einar Freyr Sigurðsson, João Silva, Aline Silveira, Natalia Silveira, Sara Silveira, Maria Simi, Radu Simionescu, Katalin Simkó, Mária Šimková, Haukur Barri Símonarson, Kiril Simov, Dmitri Sitchinava, Ted Sither, Maria Skachedubova, Aaron Smith, Isabela Soares-Bastos, Per Erik Solberg, Barbara Sonnenhauser, Shafi Sourov, Rachele Sprugnoli, Vivian Stamou, Steinþór Steingrímsson, Antonio Stella, Abishek Stephen, Milan Straka, Emmett Strickland, Jana Strnadová, Alane Suhr, Yogi Lesmana Sulestio, Umut Sulubacak, Shingo Suzuki, Daniel Swanson, Zsolt Szántó, Chihiro Taguchi, Dima Taji, Fabio Tamburini, Mary Ann C. Tan, Takaaki Tanaka, Dipta Tanaya, Mirko Tavoni, Samson Tella, Isabelle Tellier, Marinella Testori, Guillaume Thomas, Sara Tonelli, Liisi Torga, Marsida Toska, Trond Trosterud, Anna Trukhina, Reut Tsarfaty, Utku Türk, Francis Tyers, Sveinbjörn Þórðarson, Vilhjálmur Þorsteinsson, Sumire Uematsu, Roman Untilov, Zdeňka Urešová, Larraitz Uria, Hans Uszkoreit, Andrius Utka, Elena Vagnoni, Sowmya Vajjala, Socrates Vak, Rob van der Goot, Martine Vanhove, Daniel van Niekerk, Gertjan van Noord, Viktor Varga, Uliana Vedenina, Giulia Venturi, Veronika Vincze, Natalia Vlasova, Aya Wakasa, Joel C. Wallenberg, Lars Wallin, Abigail Walsh, Jonathan North Washington, Maximilan Wendt, Paul Widmer, Shira Wigderson, Sri Hartati Wijono, Seyi Williams, Mats Wirén, Christian Wittern, Tsegay Woldemariam, Tak-sum Wong, Alina Wróblewska, Mary Yako, Kayo Yamashita, Naoki Yamazaki, Chunxiao Yan, Koichi Yasuoka, Marat M. Yavrumyan, Arife Betül Yenice, Olcay Taner Yıldız, Zhuoran Yu, Arlisa Yuliawati, Zdeněk Žabokrtský, Shorouq Zahra, Amir Zeldes, He Zhou, Hanzhi Zhu, Yilun Zhu, Anna Zhuravleva, and Rayan Ziane. 2023. Universal Dependencies 2.12. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

List of Figures

List of Tables

List of Publications by Emil Svoboda

This segment chronologically lists all publications where the author of this thesis, Emil Svoboda, either authored or co-authored. The citation count is taken from Google Scholar on the 30th of July, 2024 and excludes self-citations.

Emil Svoboda, Andrzej Marciniak, Alfredo Morales Pinzon, Tomáš Bořil, Noriaki Wada, Boyan Alexandrov, Hiroto Hatabu, Charles RG Guttmann, and Vladimir Valtchinov. 2024a. Building And Validating a Digital Health Pipeline for Speech Analyses: Can Smartphones Be Used to Extract Voice Features Correlated with Multiple Sclerosis? *In preparation*

Collaboration of Charles University, Harvard University, and Los Alamos National Laboratory.

Emil Svoboda and Magda Ševčíková. 2024. PaReNT (Parent Retrieval Neural Tool): A Deep Dive Into Word Formation Across Languages. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 12611–12621, Torino, Italia. ELRA and ICCL

CITATIONS: 0
One of 10 papers to receive the **Outstanding Paper Award** at LREC-COLING 2024 out of ca. 3.500 submissions.

Emil Svoboda and Magda Ševčíková. 2024. Compounds in Universal Dependencies: A Survey in Five European Languages. In *Proceedings of the 6th Workshop on Research in Computational Linguistic Typology and Multilingual NLP*, pages 88–99, St. Julian's, Malta. Association for Computational Linguistics

CITATIONS: 0

Zdeněk Žabokrtský, Niyati Bafna, Jan Bodnár, Lukáš Kyjánek, Emil Svoboda, Magda Ševčíková, and Jonáš Vidra. 2022. Towards Universal Segmentations: UniSegments 1.0. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 1137–1149, Marseille, France. European Language Resources Association

CITATIONS: 5

Emil Svoboda, Tomáš Bořil, Jan Rusz, Tereza Tykalová, Dana Horáková, Charles R.G. Guttmann, Krastan B. Blagoev, Hiroto Hatabu, and Vladimir I. Valtchinov. 2022. Assessing Clinical Utility of Machine Learning And Artificial Intelligence Approaches to Analyze Speech Recordings in Multiple Sclerosis: A Pilot Study. *Computers in Biology and Medicine*, 148:105853

CITATIONS: 15
Collaboration of Charles University, Czech Technical University, Harvard University, and Johns Hopkins University.
Journal IF: 7.7

Emil Svoboda and Magda Ševčíková. 2022. Word Formation Analyzer for Czech: Automatic Parent Retrieval And Classification of Word Formation Processes. *Prague Bulletin of Mathematical Linguistics*, 118:55–72

CITATIONS: 3

Jonáš Vidra, Zdeněk Žabokrtský, Lukáš Kyjánek, Magda Ševčíková, Šárka Dohnalová, Emil Svoboda, and Jan Bodnár. 2021b. DeriNet 2.1

CITATIONS: 8

Emil Svoboda and Magda Ševčíková. 2021. Splitting And Identifying Czech Compounds: A Pilot Study. In *Proceedings of the Third International Workshop on Resources and Tools for Derivational Morphology (DeriMo 2021)*, pages 129–138

CITATIONS: 0
