PBML

The Prague Bulletin of Mathematical Linguistics
NUMBER 118 APRIL 2022 55–73

Word Formation Analyzer for Czech: Automatic Parent Retrieval and Classification of Word Formation Processes

Emil Svoboda, Magda Ševčíková

Charles University, Faculty of Mathematics and Physics

Abstract

We present a deep-learning tool called *Word Formation Analyzer for Czech*, which, given an input lexeme, automatically retrieves the lemma or lemmas from which the input lexeme was formed. We call this task parent retrieval. Furthermore, based on the number of words in the output sequence and its comparison to the input, the input word is classified into one of three categories: *compound*, *derivative* or *unmotivated*. We call this task word formation classification. In the task of parent retrieval, *Word Formation Analyzer for Czech* achieved an accuracy of 71%. In word formation classification, the tool achieved an accuracy of 87%.

1. Introduction

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 has 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 coin new words to suit their communicative needs; this implies 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.

 Corresponding author: svoboda@ufal.mff.cuni.cz
Cite as: Emil Svoboda, Magda Ševčíková. Word Formation Analyzer for Czech: Automatic Parent Retrieval and Classification of Word Formation Processes. The Prague Bulletin of Mathematical Linguistics No. 118, 2022, pp. 55–73. doi: 10.14712/00326585.019.

In this paper, we 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 lexeme was 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.
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')).

For the purposes of our solution, we consider products of conversion in Czech to be derivatives. The reasoning behind this will be expanded upon further in Section 2. Similarly, we consider loanwords to be unmotivated, even in cases where they are clearly motivated in their source languages (cf. *downsizing* from English, *majstršťyk* from German, or *špageta* from Italian). Due to the retroactive nature of parent retrieval and word formation classification, all examples of word formation from here on out will be structured with the product word on the left side, followed by a leftwise arrow, with the parent(s) on the right side; cf. (1).

(1) ***product*** translation.POS ← ***parent***$_1$ translation.POS ***parent***$_2$ translation.POS

We begin this paper by briefly outlining the challenges of Czech word formation, especially derivation and compounding in Section 2. Section 3 relays the handling of these issues in natural language processing (NLP), and describes in brief the *Czech Compound Splitter* tool, which is the predecessor of *WFA.ces*. Next, in Section 4, we outline the various formal difficulties that Czech word formation presents, the data that was used to train the deep-learning model ensemble, and the evaluation metrics used to measure its performance. Section 5 presents the way the underlying ensemble was trained, how it functions, and how it ended up performing, including error analysis. Section 6 compares *WFA.ces* to its predecessor and outlines future research. Finally, Section 7 contains the summary of this paper.

2. Word Formation in Czech

The foundations of theoretical approaches toward word formation in Czech have been laid by Dokulil (1962) and, since then, broadly accepted and applied to Czech and other, particularly (but not exclusively) Slavic languages; cf. all reference grammars of Czech, including the representative volume by Dokulil et al. (1986) and the latest grammars by Štícha et al. (2013) or Štícha et al. (2018).

2.1. Derivation

The basic concept of derivation as a process of the formation of new words by adding derivational affixes to already-existing lexemes or roots is in Czech fundamentally complicated by allomorphy, homonymy, and other issues, which are difficult to model computationally. For instance, two different variants of the prefix (*vy-*, *vý-*) and three different allomorphs of the same root occur in the adjective *vybraný* 'chosen' (root *-br-*), in the noun *výběr* 'choice' (*-běr-*), and in the noun *výbor* 'committee' (*-bor-*), even if they are all motivated by the verb *vybrat* 'to choose', which is, in turn, based on *brát* 'to take'. Although verb prefixation is among the less irregular processes with a minimum of formal changes, problems can also be found here. An example is the verb *obléci* 'to dress', whose simple deprefixation yields a string that does not match any existing verb (cf. **bléci* or **léci* as both the prefix *o-* and *ob-* exist in Czech). This verb is to be traced back to the verb *vléci* 'to pull', in which the initial consonant is dropped when combined with the prefix *ob-* (because of the pronunciation: *ob+vléci*; cf. (2)), but remains in place with other prefixes ((3) and (4)).

(2) ***obléci*** ← ***vléci***
dress.VERB pull.VERB

(3) ***navléci*** ← ***vléci***
pull on.VERB pull.VERB

(4) ***svléci*** ← ***vléci***
take off (clothes).VERB pull.VERB

Circumfixation, understood as prefixation and suffixation in a single step, also occurs in Czech (5). This presents difficulty for automatic solutions, because in affixation mostly a single affix is added in each step. However, if derivation of the adjective *přidrzlý* is interpreted as a sequence of derivations (cf. (6) or alternatively (7)), the product of the middle step is unattested, and therefore an incorrect retrieval. An algorithmic solution, nevertheless, has no way of inferring attestability without consulting a corpus. The implementation of a corpus lookup can mitigate this particular problem, but may introduce other issues, as demonstrated in Section 5.2.

(5) ***přidrzlý*** ← ***drzý***
a bit cheeky.ADJ cheeky.ADJ

(6) ***přidrzlý*** ← ****přidrznout*** ← ***drzý***
a bit cheeky.ADJ become cheeky.VERB cheeky.ADJ

(7) ***přidrzlý*** ← ****drzlý*** ← ***drzý***
a bit cheeky.ADJ having become cheeky.ADJ cheeky.ADJ

In Czech, conversion is formally very similar to derivation in many cases.[1] The two processes differ solely by the type of affixes used. While derivational affixes are added in derivation, conversion is assumed to be the sole addition of inflectional morphemes without adding derivational affixes. For example, the adjective in (8) is considered to be converted from the noun, despite the fact that we see a total of *two* formal changes to the parent word. First, it is vowel deletion, which can also be see as an alternation ($\emptyset \leftarrow /e/$), which is common across all of Czech word formation (cf. (17) and (19) for examples in compounding), and the addition of the adjectival ending *-í*.

(8) ***psí*** ← ***pes***
dog.ADJ dog.NOUN

However, this relatively clear distinction is very difficult for automatic analysis. An example pipeline capable of doing so would require the following:

1. reliable morphological segmentation so as to isolate the morphemes of both the input and output words;
2. reliable morpheme alignment of the input and output morphemes onto each other in order to determine which morphemes, if any, were added;
3. reliable classification of the added morphemes as either *derivational* or *inflectional*.

Additionally, the mere changing of the POS and/or the inflectional pattern of a given word without any formal changes is in the Czech linguistic tradition also considered to be conversion. This is more akin to what is considered conversion in English (cf. (9)). Such a word formation procedure cannot, however, in principle be handled by a tool like *WFA.ces* because it accepts isolated lemmas represented by a string only. From the sole lemma, it is undecidable whether we mean *raněný* 'wounded' the noun, whose parent is *raněný* 'wounded' the adjective (10), or if we mean *raněný* the adjective, whose parent is the verb *ranit* 'to wound', as these need syntactic context to be disambiguated. Therefore, when the word *raněný* is passed into *WFA.ces*, the tool is expected to return *ranit*.

[1]Most cases of conversion in Czech, as in other inflecting languages, do not conform to the central type of conversion, which is characterized by the formal identity of the input and output lexeme (word-based conversion), but rather belong to the non-central type of conversion, where the input and output share the root but may differ in inflectional markers (root-based conversion; Valera and Ruz 2021).

(9) ***raněný*** ← ***raněný***
wounded.NOUN wounded.ADJ

(10) ***raněný*** ← ***ranit***
wounded.ADJ wound.VERB

For the reasons stated in the previous paragraphs, we have decided to consider conversion as derivation and to label it as such. From a theoretical perspective, this decision can be viewed as the interpretation of conversion as derivation by zero affix.

2.2. Compounding

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).

In a recent paper on compounding in West Slavic languages, Ološtiak and Vojteková (2021) focus on compounds partially or fully motivated by elements of Greek-Latin origin (from here: *neoclassical compounds*). Four types of word formation formants are distinguished, namely bases, baseoids, affixoids, and affixes. Bases are items that can appear freely and are lexically specific (*terapie* 'therapy', like in *ergoterapie* 'occupational therapy'); baseoids are items that do not appear freely, but are lexically specific regardless (*ergo-*, in *ergoterapie* 'occupational therapy'); affixoids are non-independent items which have gradually lost their lexical specificity (*-nácť* like in *třinácť* 'thirteen' – originally from *na deset* 'to_ten'); and affixes are items which carry lexically non-specific meaning, referencing a group of referents within a given part of speech, like "object", "place", "tool", "agent" for nouns (*-ář* in *hodinář* "clockmaker").

Ološtiak and Vojteková (2021) delimit three types of compounds according to the type of formants involved. *Proper compounds*[2] are characterized as being composed of two bases, as in (11). *Semi-compounds* are composed of one base and one baseoid (12). Finally, *quasi-compounds* are composed of two baseoids (13).

(11) ***sér|-o-|pozitivní*** ← ***sérum*** ***pozitivní***
seropositive.ADJ serum.NOUN positive.ADJ

(12) ***krypto|politika*** ← ***krypto-*** ***politika***
cryptopolitics.NOUN crypto-.BASEOID politics.NOUN

(13) ***eko|logie*** ← ***eko-*** ***-logie***
ecology.NOUN eco-.BASEOID -logy.BASEOID

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

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 4.1 for more details.

From the perspective of the parent retrieval task, the simplest case of Czech compounding seems to be compounds formed by simple concatenation of two words, which typically originate in a syntactic phrase and satisfy Bozděchová's definition of *compounding improper*. For instance, the adjective in (14) corresponds directly to the syntactic phrase *vždy zelený* 'always green'. In (15), neither parent word undergoes any morphological change during the compounding procedure, which is characteristic for *compounding improper*, but the resulting noun can be associated with no such phrase, which is typical of *compounding proper*.

(14) ***vždy\|zelený*** ← ***vždy*** ***zelený***
evergreen.ADJ always.ADV green.ADJ-NOM.SG

(15) ***garáž\|mistr*** ← ***garáž*** ***mistr***
garage supervisor.NOUN garage.NOUN master.NOUN

An interfix is added between the two input words in other compounds, usually *-o-* or *-i-*. This interfix replaces the inflectional ending of any non-final parent; cf. the ending *-a* in the feminine noun *ryba* 'fish' is dropped in (16a). Additionally, stem allomorphy often appears; cf. ∅ ← /e/ in (17).

(16) a. ***ryb\|-o-\|lov*** ← ***ryba*** ***lov***
fishery.NOUN fish.NOUN hunt.NOUN

b. ***ryb\|-o-\|lov*** ← ***ryba*** ***lovit***
fishery.NOUN fish.NOUN hunt.VERB

(17) a. ***krv\|-o-\|tok*** ← ***krȩv*** ***tok***
bloodflow.NOUN krev.NOUN flow.NOUN

b. ***krv\|-o-\|tok*** ← ***krȩv*** ***téci***
bloodflow.NOUN krev.NOUN flow.VERB

Compounding and derivation in one step (18) as well as compounding and conversion in one step (19) are possible, often accompanied by vowel and consonant changes; for instance, in (19) two cases of stem vowel alternation (∅ ← /e/ in *ps* ←

pes and /o/ ← /e:/ in *vod* ← *vést*), a stem consonant alternation (/d/ ← /s/ in *vod* ← *vést*), and an interfix insertion all occur at the same time. Note that in parallel to (19), the compounds in (16a) and (17a) can also be analysed as outputs of compounding and conversion in one step if a noun and a verb are considered as inputs (cf. (16b) and (17b)). In contrast, for *psovod* such an alternative is not available because **vod* is not attestable as a separate noun in Czech. In the data we use in our experiments, both analyses are captured (see Section 4.1).

(18) ***modr|-o-|oký*** ← ***modrý oko***
blue-eyed.ADJ blue.ADJ eye.NOUN

(19) ***ps|-o-|vod*** ← ***pes vést***
dog handler.NOUN dog.NOUN lead.VERB

In (20), the compound is traced back to the noun phrase *chtivý holek* 'wanting of girls', with its original ordering switched. Additionally, there are compounds that cannot be meaningfully split into two parents; cf. the compound in (21) 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.noun' ← *-letý* '-year.adj').

(20) ***holek|chtivý*** ← ***chtivý holek***
wanting girls.ADJ wanting.ADJ girl.NOUN.GEN.PL

(21) ***dva|a|půl|letý*** ← ***dvě a půl léto***
two-and-a-half-year-old.ADJ two.NUM and.CONJ half.NUM year.ADJ

Neoclassical compounds, under our interpretation, constitute what Ološtiak and Vojteková (2021) consider *semi-composition* and *quasi-composition*. The noun *sociologie* 'sociology' in (22) is an example of *quasi-composition* in their framework. In a broader sense, chemical compounds satisfy the definition of semi-composition, as in (23).

Products of reduplication are considered to be compounds for the purposes of this paper, because they formally tend to behave very similarly to compounds (24).[3]

(22) ***soci|-o-|logie*** ← ***-soci- -log-***
sociology.NOUN -soci-.NEOCON -log-.NEOCON

(23) ***tetra|chlor|ethylen*** ← ***-tetra- chlor ethylen***
tetrachlorethylene.NOUN -tetra-.NEOCON chlorine.NOUN ethylene.NOUN

(24) ***čern|-o-|černý*** ← ***černý černý***
pitch black.ADJ black.ADJ black.ADJ

It is worth noting that in spite of all of these formal peculiarities, Czech native speakers tend to find it easy to correctly determine the parents of a given compound. Opportunities for folk etymologies similar to the English *cockroach* (apparently from

[3]Cf. also Hoeksema (2012) who proposes a category of elative compounds.

cock + *roach*, actually from the Spanish *cucaracha*) are few and far between. One such example is *medvěd* ('bear') from *med* ('honey') + *jíst* ('eat'), whose etymology is obfuscated by diachronic sound changes. This may lead to a Czech speaker wrongly analyzing the word either as unmotivated or as *med* ('honey') + *vědět* ('know').

3. NLP approaches toward word formation

Unlike the long-lasting attention of theoretical linguists, Czech word-formation has come into focus of NLP rather recently. The topic has been addressed primarily by capturing it using static data resources. Additionally, the word formation of other languages has been in the scope of NLP for a much longer time than Czech word formation has.

3.1. Derivation trees

Derivancze, which stands for Derivational Analyzer of Czech (Pala and Šmerk, 2015), is a static data resource that can be used to return not only the derivational parents of a given word, but also its derivatives. The tool does not seem to contain compounding relations.

A similar word formation resource for the language, DeriNet, maps derivation by means of linking words to the words they are respectively derived from all the way to their roots, which should canonically be unmotivated. DeriNet has additionally been equipped for handling compounding as well since version 2.0, in that its data format allows for a single lexeme to have multiple parents, and it contains an optional flag for each lexeme signaling whether or not the given lexeme is a compound. Similarly, it is equipped with the possibility of including an *unmotivated* flag (Vidra et al., 2019).

DeriNet version 2.1 (Vidra et al., 2021) contains 33,938 compounds, of that 2,691 compounds with linked parents,[4] and a total of words 13,611 labelled as unmotivated. Furthermore, it contains 664,430 lexemes which have a single parent, are not roots of a derivation tree, and are lowercase. These items can be assumed to be derivatives or products of conversion.

3.2. Compound splitting

Splitting of Czech compounds has been addressed by *Czech Compound Splitter* (Svoboda and Ševčíková, 2021), which is the predecessor of *WFA.ces*. Its primary capability, compound splitting, is parent retrieval limited to confirmed compounds. Analogically, it also performed compound identification, which is word formation classification limited to a binary set of classes – *compounds* and *non-compounds*. The performance and versatility of the tool was what ultimately inspired us to take a new

[4]Manually annotated and added as part of the creation of *Czech Compound Splitter* (Svoboda and Ševčíková, 2021).

direction in word formation analysis and generalize its utility. As there is no other compound splitting tool available for Czech, this task has been demonstrated to be feasible in several other languages.

Henrich and Hinrichs (2011) linked German nominal compounds to their respective parents in GermaNet (Hamp and Feldweg, 1997) 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.

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%.

As for other languages, 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.

3.3. Stemming

The closest widely used procedural task related to parent retrieval is *stemming*, already mentioned in Section 1. 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 paper, 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 (25) 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, (26) or alternatively (27) is what we would expect a parent retriever to do.

(25) ***unhappiness*** ← ****happi***

(26) ***unhappiness*** ← ***unhappy***

(27) ***unhappiness*** ← ***happiness***

Of course, one can use a parent retriever for a purpose similar to that of a stemmer by calling it repeatedly, like in (28) or alternatively (29), which is how a parent retriever can be used for purposes similar to a stemmer. Parent retrieval does *not* handle inflection, so inputting *happiest* into *WFA.ces* may in practice result in unexpected behavior.

(28) ***unhappiness*** ← ***unhappy*** ← *happy*

(29) ***unhappiness*** ← ***unhappy*** ← *happy*

4. Data and evaluation methodology

Word Formation Analyzer for Czech is a deep-learning based tool, and as such it required data to be trained, tuned, and tested. The following section describes where this data was taken, how it was augmented and preprocessed, and how it was used to fine-tune and test the tool's performance.

4.1. Golden data set

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 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 1.

4.1.1. 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 (30) 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, but formally correctly formed, like in the example.

(30) ***důležit*|*-o-*|*neomylný*** ← ***důležitý*** ***neomylný***
important-infallible.ADJ important.ADJ infallible.ADJ

For the purposes of training *WFA.ces*, we simulated a number of such compound formation procedures in Python using randomly selected lexemes from DeriNet weighted by their corpus frequency, creating a data set of 280,000 synthetic compounds. We did not synthesize any derivatives, because the available number of derivative items was deemed sufficient for the purposes of training deep-learning models.

4.2. Evaluation methodology

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]

[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.

Formation class	train	test	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 1. 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

In the case of neoclassical compounds, we strictly require the predicted constituents to be correctly hyphenated, as in (31), otherwise the prediction counts as incorrect, cf. (32) and (33).

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

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

(33) ***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. (34a), (34b)). 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.

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

(35) b. ***ryb|-o-|lov*** ← ***ryba*** ***lovit*** ✓
fishery.NOUN fish.NOUN hunt.VERB

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 2. Description of the configurations in the model ensemble used in Word Formation Analyzer for Czech

5. Building and testing the tool

5.1. Model ensemble training and tuning

The core of *WFA.ces* was built using the *Marian* framework developed by Junczys-Dowmunt et al. (2018), utilizing an ensemble of three models described in Table 2. All of the models in the ensemble were then trained on the dataset described in Table 1 with layer regularization. The model 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. The hyperparameters of the model ensemble, such as the dropout rate and number of training iterations, were fine-tuned manually on the test set.

One interesting obstacle that had to be overcome was the fact that, as the FAQ page of the *Marian* project explicitly states:[6] "Convolutional character-level NMT models are not yet supported." Since nothing but isolated lemmas was supported to the model, character-level learning was strictly necessary. We solved this by replacing all spaces (which were only present in the parent sequences of compounds) with an underscore character, and by adding spaces between each character in the string. Thus, *zelenočerný* 'green-black' became *z e l e n o č e r n ý*, and its corresponding parents *zelený černý* became *z e l e n ý _ č e r n ý*. This forced the models to consider each grapheme as a separate word, solving the problem of the models being word-level only.

5.2. 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.

[6] https://marian-nmt.github.io/faq

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*.
- *Frequency*: *WFA.ces* uses a list of relative corpus frequencies[7] 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*.

5.3. Performance evaluation and error analysis

The performance of *WFA.ces* in parent retrieval can be viewed in Table 3. 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 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

[7] Acquired from DeriNet 2.0 for the purposes of this paper.

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. 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 $n = 4$.

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

Table 4. The Precision, Recall and F1 scores achieved by Word Formation Analyzer for Czech for each word formation class, as measured on the validation set for $n = 4$.

regards to each class can be viewed in Table 4, 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 5). 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 5). 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 5 – *malý* 'small'). Additionally, the tool has no way of leveraging semantics to its advantage, leading it to analyze *siný* 'light

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 5. A sample of the various errors that WFA.ces made in parent retrieval under different reranking methods. Some of the errors were made under all of them.

blue' as a derivative of *sít* 'to sow' (the penultimate line of Table 5). 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 5).

6. Discussion

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 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.

While *WFA.ces* shows promising results, there is still much to be improved and expanded upon. One of the easiest improvements would be the ability to discriminate between native compounds and neoclassical compounds, since *WFA.ces*'s model ensemble is trained to detect neoclassical constituents by marking them with hyphens. The classification of neoclassical compounds could therefore be implemented without adjusting the deep-learning model ensemble at all. The granularity of this classification could be easily increased even further by discriminating between what

Ološtiak and Vojteková consider to be *semi-compounds* (formed from a neoclassical constituent and a native word) and *quasi-compounds* (formed solely from neoclassical constituents).

Furthermore, products of conversion and derivatives have been grouped into a single class in this study, but it could potentially be valuable to be able to automatically discriminate between the two as well. Since in Czech, conversion is linguistically distinct from derivation by the addition of inflectional affixes as opposed to the addition of derivational affixes, this could hypothetically be achieved by using two lists, one of word formation affixes and another of inflectional affixes. Perhaps the most interesting future development of *WFA.ces* would be its generalization into other languages.

7. Conclusions

We presented *Word Formation Analyzer for Czech*, a computational tool for 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 discriminating 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. In the future, it would be valuable if *WFA.ces* could be made to distinguish between native and neoclassical compounds, as well as between derivatives and products of conversion. Furthermore, we would like to see the tool generalized into more languages.

Acknowledgements

This work was supported by the Grant No. GA19-14534S of the Czech Science Foundation, the LINDAT/CLARIAH-CZ project of the Ministry of Education, Youth and Sports of the Czech Republic (project LM2018101), and by the Grant No. START/HUM/010 of Grant schemes at Charles University (Reg. No. CZ.02.2.69/0.0/0.0/19_073/0016935).

Bibliography

Biba, Marenglen and Eva Gjati. Boosting text classification through stemming of composite words. In *Recent Advances in Intelligent Informatics*, pages 185–194. Springer, 2014. doi: 10.1007/978-3-319-01778-5_19.

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

Carlberger, Johan, Hercules Dalianis, Martin Duneld, and Ola Knutsson. Improving precision in information retrieval for Swedish using stemming. In *Proceedings of the 13th Nordic Conference of Computational Linguistics (NODALIDA 2001)*, pages 17–22, 2001.

Chmelař, Petr, David Hellebrand, Michal Hrušecký, and Vladimír Bartík. Nalezení slovních kořenů v češtině. In *Znalosti 2011: Sborník příspěvků 10. ročníku konference*, pages 66–77. VŠB-Technical University of Ostrava, 2011. URL `https://www.fit.vut.cz/research/publication/9473`.

Clouet, Elizaveta L. and Béatrice Daille. Splitting of compound terms in non-prototypical compounding languages. In *Workshop on Computational Approaches to Compound Analysis*, pages 11–19, 2014. doi: 10.3115/v1/W14-5702.

Dokulil, Miloš. *Tvoření slov v češtině 1: Teorie odvozování slov*. Academia, Praha, 1962.

Dokulil, Miloš, Karel Horálek, Jiřina Hůrková, Miloslava Knappová, and Jan Petr. *Mluvnice češtiny 1. Fonetika, fonologie, morfonologie a morfematika, tvoření slov*. Academia, Praha, 1986.

Dolamic, Ljiljana and Jacques Savoy. Indexing and stemming approaches for the Czech language. *Information Processing & Management*, 45(6):714–720, 2009. doi: 10.1016/j.ipm.2009.06.001.

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

Hellwig, Oliver and Sebastian Nehrdich. 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, 2018. doi: 10.18653/v1/D18-1295.

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

Hoeksema, Jack. Elative compounds in Dutch: Properties and developments. In *Intensivierungskonzepte bei Adjektiven und Adverben im Sprachenvergleich*, pages 97–142. Kovač Verlag, Hamburg, 2012.

Huet, Gérard. A functional toolkit for morphological and phonological processing, application to a Sanskrit tagger. *Journal of Functional Programming*, 15(4):573–614, 2005. doi: 10.1017/S0956796804005416.

Junczys-Dowmunt, Marcin, 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. Marian: Fast Neural Machine Translation in C++. In *Proceedings of ACL 2018, System Demonstrations*, pages 116–121, 2018. doi: 10.18653/v1/P18-4020.

Křen, Michal, 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. SYN2015: Representative Corpus of Contemporary Written Czech. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 2522–2528, 2016.

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

Ma, Jianqiang, Verena Henrich, and Erhard Hinrichs. Letter sequence labeling for compound splitting. In *Proceedings of the 14th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 76–81, 2016. doi: 10.18653/v1/W16-2012.

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

Pala, Karel and Pavel Šmerk. Derivancze – derivational analyzer of Czech. In *International Conference on Text, Speech, and Dialogue*, pages 515–523, 2015. doi: 10.1007/978-3-319-24033-6_58.

Porter, Martin F. An algorithm for suffix stripping. *Program: electronic library and information systems*, 14:130–137, 1980. doi: 10.1108/eb046814.

Porter, Martin F. Snowball: A language for stemming algorithms. Published online, October 2001. URL `http://snowball.tartarus.org/texts/introduction.html`. Accessed 21.01.2022, 15.00h.

Sugisaki, Kyoko and Don Tuggener. German compound splitting using the compound productivity of morphemes. In *14th Conference on Natural Language Processing*, pages 141–147, 2018.

Svoboda, Emil and Magda Ševčíková. 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, 2021.

Štícha, František, Miloslav Vondráček, Ivana Kolářová, Jana Bílková, and Ivana Svobodová. *Akademická gramatika spisovné češtiny*. Academia, Praha, 2013.

Štícha, František, Ivana Kolářová, Miloslav Vondráček, Ivana Bozděchová, Jana Bílková, Klára Osolsobě, Pavla Kochová, Zdeňka Opavská, Josef Šimandl, Lucie Kopášková, and Vojtěch Veselý. *Velká akademická gramatika spisovné češtiny 1: Morfologie: Druhy slov / Tvoření slov*. Academia, Praha, 2018.

Valera, Salvador and Alba Ruz. Conversion in English: homonymy, polysemy and paronymy. *English Language and Linguistics*, 25(1):181–204, 2021. doi: 10.1017/S1360674319000546.

Vidra, Jonáš, Zdeněk Žabokrtský, Magda Ševčíková, and Lukáš Kyjánek. 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, 2019.

Vidra, Jonáš, Zdeněk Žabokrtský, Lukáš Kyjánek, Magda Ševčíková, Šárka Dohnalová, Emil Svoboda, and Jan Bodnár. DeriNet 2.1, 2021. URL `http://hdl.handle.net/11234/1-3765`. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Address for correspondence:
Emil Svoboda
`svoboda@ufal.mff.cuni.cz`
Malostranské náměstí 25, 118 01 Praha 1, Czech Republic