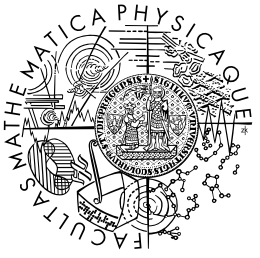

FACULTY
OF MATHEMATICS
AND PHYSICS
Charles University

BACHELOR THESIS

Aleš Manuel Papáček

Identification of Morpheme Origin

Institute of Formal and Applied Linguistics

Supervisor of the bachelor thesis: prof. Ing. Zdeněk Žabokrtský, Ph.D.

Study programme: Computer Science – Artificial Intelligence

Prague 2025

I declare that I carried out this bachelor 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 Prague 1 May 2025

Aleš Manuel Papáček

I would like to thank my supervisor, prof. Ing. Zdeněk Žabokrtský, Ph.D., for many valuable consultations and advice throughout the work on this thesis. I also thank Tomáš Sourada for his help and guidance, Vojtěch John for providing morphologically segmented and classified data, and Tomáš Janeček for dedicating his time to annotate part of the dataset. Last but not least, I would like to thank my family and my girlfriend for their continuous support.

Title: Identification of Morpheme Origin

Author: Aleš Manuel Papáček

Institute: Institute of Formal and Applied Linguistics

Supervisor: prof. Ing. Zdeněk Žabokrtský, Ph.D., Institute of Formal and Applied Linguistics

Abstract: This thesis focuses on predicting the etymological origin of individual morphemes in Czech words. Given morphologically segmented sentences, the task is to determine for each morpheme whether it is native or borrowed, and if borrowed, to identify the languages through which it entered Czech. We created a manually annotated dataset of Czech sentences with morpheme-level etymology labels for model training and evaluation. Features such as character n-grams and lexical or positional morpheme types were used to train several supervised machine learning classifiers. We also experimented with morph and word embeddings, as well as semi-supervised self-training, but these did not improve performance. The best model was an MLP trained on extracted features and enriched with etymological dictionary data. It outperformed all baselines, including predictions by OpenAI's latest reasoning large language model, o3. Although the baseline F1 score is high, predicting all morphemes as native achieves 90.1 %, the best model reached 96.8 % and reduced the baseline error by 67.7 %.

Keywords: Morphology, Morpheme, Etymology, Machine learning

Název práce: Identifikace původu morfémů

Autor: Aleš Manuel Papáček

Ústav: Ústav formální a aplikované lingvistiky

Vedoucí bakalářské práce: prof. Ing. Zdeněk Žabokrtský, Ph.D., Ústav formální a aplikované lingvistiky

Abstrakt: Tato práce se zaměřuje na predikci etymologického původu jednotlivých morfémů v českých slovech. Cílem je určit z morfologicky segmentovaných vět, zda je morfém nativní, nebo přejatý, a pokud přejatý, přes které jazyky se do češtiny dostal. Vytvořili jsme ručně anotovaný dataset českých morfologicky segmentovaných vět, určený pro trénování a vyhodnocování modelů. Z dat jsme extrahovali rysy jako znakové n-gramy a typy morfémů. Na těchto datech jsme trénovali několik modelů strojového učení. Experimentovali jsme také se samoučením (technika učení bez učitele) a embeddingy morfémů i slov, tyto metody však nepřinesly žádné zlepšení. Nejlepší model, postavený na klasifikátoru MLP a využívající dodatečná data z etymologického slovníku, překonal všechna definovaná základní řešení včetně predikcí generovaných pomocí nejnovějšího LLM modelu o3 od OpenAI. Baseline řešení dosahuje vysokého skóre, predikce nativního původu pro všechny morfémy dosahuje F1 skóre 90,1 %, náš nejlepší model dosáhl 96,8 % a snížil tak chybovost tohoto přístupu o 67,7 %.

Klíčová slova: Morfologie, Morfém, Etymologie, Strojové učení

Contents

Introduction **7**

1 Theoretical Background **10**
1.1 Morphology and Word Structure 10
1.1.1 Morphemes and Morphs 10
1.1.2 Morpheme types 11
1.1.3 Problematic affixes 13
1.2 Etymology 14
1.2.1 Languages of the World 15
1.2.2 Language groups 15
1.2.3 Influence of Cultures and Language Contact 18
1.3 Etymology of Morphemes 19
1.3.1 Calques 20

2 Related Work and Data Sources **21**
2.1 Deconstructor 21
2.2 DeriNet 22
2.3 Czech Etymological Dictionary 24
2.3.1 SIGMORPHON Shared Task 25
2.4 Segmentation 25
2.5 Morpheme Classification 25

3 Data and Evaluation Methodology and Baselines **26**
3.1 Data 26
3.1.1 CzEtyL 26
3.1.2 Manually Annotated Dataset 28
3.2 Evaluation Methodology 30
3.2.1 F1-Score 30
3.2.2 Native vs. Borrowed 30
3.2.3 Score Grouped by morphemes 31
3.2.4 Relative error reduction 31
3.2.5 Expected Bounds of Performance 31
3.3 Inter-Annotator Agreement 33
3.3.1 Results of inter annotator agreement 34
3.4 Baselines 34
3.4.1 Always-Czech Baseline 35
3.4.2 Memorization-Based Approach 35
3.4.3 Word Lemmatization Approach 35
3.4.4 Morpheme-Based Baseline 35
3.4.5 Using Large Language Model 36

4 Model for Morpheme-Level Etymology Prediction **38**
4.1 Features 38
4.1.1 Morph n-grams 38
4.1.2 Morph Types 38

4.1.3 Vowel Start and End . . . 39
4.1.4 Embeddings . . . 39
4.2 Context . . . 40
4.3 Extending the Training Data with Etymological Dictionary . . . 41
4.4 Representation of Target Classes . . . 42
4.4.1 Label Statistics from Training Data . . . 42
4.4.2 Extended Training Data . . . 43
4.5 Classification Models . . . 44
4.6 Self-Training . . . 45

5 Experiments and Results **46**
5.1 Experiments on the Development Set . . . 46
5.1.1 Baseline Results . . . 46
5.1.2 Learning Model Results . . . 47
5.2 Final Results on Test Set . . . 52
5.2.1 Baseline Results . . . 53
5.2.2 Learning Models Results . . . 53
5.2.3 Best Model Results . . . 54

6 Implementation Details **55**
6.1 Repository Structure . . . 55
6.1.1 Data . . . 56
6.1.2 Source Code . . . 56
6.2 Main . . . 57
6.3 Morph Classifier . . . 57
6.3.1 DataSentence Class . . . 57
6.3.2 Fitting . . . 58
6.3.3 Prediction . . . 58
6.3.4 Save and Load . . . 58
6.4 Reproducibility . . . 58

Conclusion **60**

Bibliography **62**

List of Figures **65**

List of Tables **66**

Introduction

According to Eberhard et al. (2025a), there are over 7,000 living languages worldwide. Despite this linguistic diversity, many languages share notable similarities in vocabulary, grammar, and syntax. These similarities typically result from historical relationships between languages, either through inheritance from a common ancestor—such as Latin evolving into Spanish, Italian, and other Romance languages—or through language contact, where languages borrow vocabulary from each other. English, for instance, has borrowed extensively from French, while Czech has incorporated numerous words from German. Both French and German have also borrowed extensively from Latin, which itself borrowed significantly from Greek.

An example of such borrowing is the Czech word "škola" (school), which traces its roots from the German "Schule", originating from Latin "schola", which itself is derived from Greek "skholē".

Traditionally, etymology has focused on tracing the history and evolution of entire words. However, linguistic borrowing and inheritance often occur at a finer level, affecting smaller meaningful units known as morphemes. Analysing the origins of individual morphemes provides a more detailed and nuanced view of how languages change and influence each other. Although linguists have long recognized the importance of morpheme-level etymology—as seen in specialized resources like lexicons of Indo-European roots and verbs (Kümmel et al., 2001)—this information is rarely made explicit in standard etymological dictionaries for modern languages.

Task and Approach

This thesis defines the task of identifying the etymological origin of individual morphemes in Czech words. Given morphologically segmented sentences, the goal is to determine whether each morpheme is native or borrowed, and if borrowed, through which languages it entered Czech. To address this challenge, we created a manually annotated dataset by assigning morpheme-level etymological labels to each morpheme in morphologically segmented Czech sentences. A total of 300 sentences from the SIGMORPHON (2022) shared task on morphological segmentation were used.[1]

We develop several baseline approaches, including a method that memorizes the most frequent origin for each morph from the training data, rule-based techniques that utilize an etymological lexicon, and predictions produced by OpenAI's latest state-of-the-art reasoning model, o3 (OpenAI, 2025).

For the main learning approach, we predict the etymology of morphs based on features such as character n-grams, morph type, and morph position within the word. Morphological classification is performed using a model by John (2024).

We experiment with a range of machine learning models, including logistic regression (LR), support vector machines (SVM), and multilayer perceptrons (MLP), to learn to predict the origin of each morph. In addition, we explore the use of morph and word embeddings for this task and experiment with self-training

[1]The dataset is available at `https://github.com/ampapacek/MorphemeOrigin/tree/main/data/annotations`

as a semi-supervised learning method.

Languages Used

The concepts explored in this thesis are relevant to all natural languages. However, the experimental part focuses on Czech, as we have suitable data in Czech for this type of research. Additionally, the author of the latest Czech Etymological Dictionary Rejzek, 2019 cooperates on this project, allowing direct access to this valuable source.

In the theoretical sections, examples are provided in English, Czech, and occasionally Spanish or German, to illustrate how the same concepts manifest across different languages. These particular languages were selected because they are among the most widely spoken globally and are also languages with which the author is personally familiar.

Structure of the Thesis

In Chapter 1, we define key terminology from etymology and morphology and provide the necessary theoretical background.

In Chapter 2, we introduce existing tools that attempt to address similar or related problems, and we describe the datasets and resources used in this thesis.

Chapter 3 presents the details of our data, outlines the evaluation methodology, and describes several baseline models for the morph-level etymology prediction task.

Chapter 4 details the development and characteristics of the machine learning model used for predicting morph origins.

In Chapter 5, we discuss and analyse the results achieved by our models on both the development and test sets.

Finally, Chapter 6 covers practical implementation details and provides guidance on using and reproducing the experiments included in this thesis.

Etymological Dictionary by J. Rejzek

We will frequently refer to the Czech Etymological Dictionary by Rejzek (2019). When discussing etymologies described in this dictionary, the [R] mark will be used to indicate the source.

Acknowledgments

During the development of this thesis, I made use of several AI tools to support both the implementation and the writing. For coding assistance and suggestions, I used GitHub Copilot[2] and ChatGPT[3].

[2] `https://github.com/features/copilot`

[3] `https://chatgpt.com`

To check grammar and improve the clarity of the text, I relied on Grammarly[4], including its AI-based writing features. I also used Writefull[5], integrated with Overleaf. ChatGPT assisted in rephrasing parts of the text.

I declare that I have not used any AI tool to write any complete chapter, section, or paragraph of this thesis. AI tools were used only to help rephrase or improve the clarity of text that I had already written myself.

[4] https://www.grammarly.com
[5] https://www.writefull.com

1 Theoretical Background

In this chapter, we define the key terms used in the following chapters, primarily related to morphology and etymology. We establish fundamental concepts such as morpheme, morph, morphological segmentation, and types of language borrowings. Although many textbooks have been written on these topics—for example, Booij (2007), Haspelmath and Sims (2013), Haspelmath (2020), and Aronoff and Fudeman (2011)—the terminology is not fully unified and often varies across different publications. In this thesis, I mainly follow the definitions proposed by Haspelmath and Sims (2013). To ensure consistency, we provide definitions of the terms as they are used in this thesis, while acknowledging that some of them may remain somewhat vague due to the complexity and ambiguity of linguistic classification in certain cases.

1.1 Morphology and Word Structure

Morphology is a branch of linguistics that studies the form and the internal structure of words. The word morphology was first used by Goethe (1817) in a biological context, where it referred to the study of the structure of organisms. The etymology of the word morphology comes from Greek *morphē*, meaning "shape" or "form," and *-logía*, meaning "study" or "science" (derived from the verb *légō*, "to say, read, or collect") [R].

Morphology analyses words in a language and divides them into smaller units that carry meaning; these units together form the structure of the word. For example, the complex word *disrespectfulness* can be segmented as *dis-* + *respect* + *-ful* + *-ness*.

Each part has a defined meaning, either lexical or grammatical. The smallest indivisible units are called morphemes. Some words can be just one morpheme (e.g. *pen*).

The root *respect* could be further segmented as *re-* + *spect* based on its etymology. The word *respect* comes from Latin *re-* + *specere*, meaning "to look again" [R].

1.1.1 Morphemes and Morphs

The terms *morpheme* and *morph* are often used interchangeably, and in some cases, the term *morpheme* is applied to any smaller part of a word.

German linguist Haspelmath (2020) identifies at least three distinct uses of the term *morpheme*. For the purposes of this thesis, we adopt the following definitions for morpheme and morph:

> A **morpheme** is a set of minimal forms with identical syntacticosemantic content.
>
> A **morph** is the concrete realization of a morpheme.

To better understand the distinction, we will illustrate it with an example. In English, the plural form is created by adding *-s*, *-es*, or *-ies* to the singular noun.

The morphs *-s*, *-es*, and *-ies* are all elements of the same morpheme representing the plural form (denoted as {Plural} or {-s, -es, -ies}).

Similarly, in Spanish, the plural morpheme {Plural} can be realized through different morphs, such as *-s* (e.g., *cerveza* + *-s* → *cervezas* (beers)) or *-es* (e.g., *corazón* + *-es* → *corazones* (hearts)).

This is in case of regular nouns, for both English and Spanish there are irregularities, which we will discuss later.

Likewise, in Czech, the actor morpheme {Actor} can be realized through different morphs, such as *-el* (e.g., *učit* + *-el* → *učitel* (teach + -er)) or *-ář* (e.g., *kov* + *-ář* → *kovář* (blacksmith)).

The phenomenon where a morpheme has multiple variants is called **allomorphy**, and the morphs that serve as realizations of the same morpheme are called **allomorphs**.

1.1.2 Morpheme types

In this section, we define the root of a word, affixes, and their types.

Root

The *root* morpheme carries the core lexical meaning of a word. In some languages, such as English, roots can often function as stand-alone words. However, in languages like Czech, words typically consist of multiple morphemes, as they frequently include inflectional affixes that determine grammatical features such as case, number, gender, tense, or mood.

According to Haspelmath (2020) the **root** is a morpheme carrying meaning that cannot be analysed any further into constituent morphemes.

Morphemes that cannot stand alone and must be attached to a root are called **affixes**. These affixes serve to modify either the lexical meaning or the grammatical properties of the word.

For example, Czech is a morphologically rich language that can form complex words by adding multiple affixes to a root. The word

*nejneob**hospod**ařovávatelnějšími*

meaning "with the most impossible to continuously manage" contains multiple affixes surrounding the root *hospod*.

Although this word is grammatically correct, it is artificially created to demonstrate word length and is not used in everyday language.

Affixes

A word may consist of a root and additional elements called affixes. **Affixes** are bound morphemes that attach to a root to modify its meaning or grammatical role. Affixes cannot occur by themselves as stand-alone words (Haspelmath; Sims, 2013).

Prefixes are affixes that precede the root.

Suffixes are affixes that follow the root.

An **Ending** is a special type of suffix that carries specific grammatical features like tense, mood, person, or number. Endings are often distinguished from other suffixes by their strictly inflectional function.

We illustrate this with an example from the beginning of this chapter: the word *disrespectfulness* consists of one prefix (*dis-*), the root (*respect*), and two suffixes (*-ful* and *-ness*).

Base and Stem

The **base** of a word is the form to which affixes attach. When the attached affix is inflectional the term **stem** is often used instead of base.

The **stem** of a word is defined as its base form without inflectional affixes (Haspelmath; Sims, 2013). For example, in English, the verb form *walking* has the stem *walk*, which also serves as the basis for other forms such as *walks* and *walked*, where the inflectional suffixes *-s* and *-ed* express grammatical distinctions like tense or number.

The stem can sometimes be identical to the root, as in the previous example, but it can also be more complex. For instance, in the verb *rebuilds* (third person singular of *rebuild*), the stem is *rebuild*, while the root is just *build*.

Less common affixes

In addition to prefixes and suffixes, there are other types of affixes classified based on their relative position to the root.

A **circumfix** is an affix that consists of two parts, which attach to a root from both sides. Unlike a simple combination of a prefix and a suffix, circumfixes always function as a fixed pair, meaning their individual parts cannot be used separately. A common example is in German, where the past participle is often formed with the circumfix *ge-...-t*, as in *gesagt* ("said") from *sagen* ("to say").

An **infix** is an affix inserted within the root. Infixes are relatively rare in Indo-European languages but occur in other language families, such as Austronesian. For example, in Tagalog, the infix *-um-* is used in verb formation, as in *liwanag* ("clearness") becoming *lumiwanag* ("to become clear") (Schachter; Otanes, 1983).

A **compound** is a word that contains more than one root. The roots in a compound can be connected by **interfixes**. For example, in English, *blackboard* is a compound made up of two roots: *black* and *board*, without an interfix. In German, the word *Liebesbrief* ("love letter") is formed from the roots *Liebe* ("love") and *Brief* ("letter"), with the interfix *-s-* appearing between them.

Derivation and Inflection

A key criterion for categorizing affixes is whether they contribute to derivation or inflection. **Inflection** modifies the grammatical form of a base word without changing its core meaning, while **derivation** creates a new lexeme from an existing one, often altering its meaning or changing its part of speech; it is often done by adding a derivational affix (Aronoff; Fudeman, 2011).

The term *inflection* comes from the Latin *flectere* [R], meaning "to bend." It applies to a base word and "bends" its shape to express the desired *morphosyntactic information*, including tense, aspect, number, and case.

Inflection modifies a word's form without changing its core meaning, typically by adding an inflectional affix. However, it can also involve internal modification rather than affixation. For example, the English verb *talk* inflects to *talked* with the suffix *-ed* (past tense), while *sing* changes to *sang* (past tense) through vowel alternation, a process known as **ablaut**.

As defined by Booij (2007), a **lexeme** is an abstract representation of a word. It captures the core meaning shared across different grammatical forms. Words formed through inflection belong to the same *lexeme*, meaning they differ only in grammatical features but retain the same fundamental meaning. For example, *talk*, *talks*, *talked*, and *talking* are all word forms of the lexeme *talk*. Dictionaries typically list lexemes rather than their individual inflected forms, so a person searching for *talking* would find it under *talk*.

The distinction between derivation and inflection is not always clear-cut, as some affixes can show characteristics of both. For instance, the Czech prefix *nej-*, which forms the superlative degree of adjectives, does change the meaning to some extent, but since adjective degree is a grammatical feature, it is typically considered an inflectional affix.

According to Aronoff and Fudeman (2011), the main distinction is that derivation creates new lexemes, whereas inflection produces different forms of the same lexeme, with the specific form determined by the syntactic context. For example, *walk* and *walks* (where *-s* is an inflectional affix) share the same base meaning, and the addition of *-s* depends on subject-verb agreement. In contrast, in the pair *teach* and *teacher*, the affix *-er* is derivational—it changes the part of speech from verb to noun. Derivation can significantly change the meaning of a word. In Czech, for example, *bít* ("to beat") and *zabít* ("to kill'') are both verbs, but the prefix *za*-changes the meaning considerably.

Another distinction recognized by Aronoff and Fudeman (2011) is that inflectional affixes are generally positioned further from the root than derivational affixes. For example, in the English word *rationalizations*, the derivational suffixes *-al*, *-iz*, and *-ation* appear closer to the root *ration*, while the inflectional suffix *-s* is positioned at the very end.

1.1.3 Problematic affixes

While many affixes can be clearly segmented and classified, this is not always the case. Some morphological elements do not follow regular or easily distinguishable patterns. In the following section, we describe several examples of such irregularities in more detail.

Sometimes it is not straightforward to determine the type of certain morphemes. Haspelmath and Sims (2013) demonstrates this with examples such as *bioethics* and *aristocrat*. Both *bio-* and *-crat* could be classified as affixes, as they do not occur independently. On the other hand, they have a clear lexical meaning and cannot be decomposed further. According to our definition, they would be classified as roots that happen to occur only in compounds.

In cases such as irregular plural forms, the affix {plural} cannot be directly segmented out. For instance, the word *fish* remains identical in both singular and plural forms; thus, adding the plural feature to *fish* results in no visible change. Morphemes that alter meaning without an overt form are typically referred to as

zero morphemes (often denoted as {∅} or {0}).

Also falling under this category are morphemes that cause a change in the root itself, such as *woman* + {plural} → *women*. These are called **simulfixes**.

When an entire morpheme is replaced by a different form, it is referred to as a **suppletive morpheme**. This phenomenon often occurs in derivation. For example *good* + {comparative} → *better*, or *go* + {past tense} → *went*.

Cranberry Morphemes

Another type of morpheme that breaks the idea of morphemes being reusable building blocks are the so-called **cranberry morphemes**. These are morphemes that appear only in one specific word and are not used anywhere else.

The name comes from the English word *cranberry*. The part *berry* is productive and appears in many words like *blueberry*, *strawberry*, etc. But the part *cran* is only used in *cranberry* and has no meaning on its own.

Another example is *cobweb*, where *web* is meaningful, but *cob* is not used in any other word in modern English.

Sometimes, when attempting to break words down into morphemes to derive new words, the morphological segmentation can be incorrect, resulting in new words with morphemes that originally had no meaning. This occurs most often when analysing loanwords from different languages.

A widely known example is the word *hamburger*, originally derived from the German city of *Hamburg*. In English, it was reanalysed as *ham* + *burger*, which led to the creation of new words such as *cheeseburger*, *chickenburger*, and others.

Another example is the word *alcoholic*, which should be segmented as *alcohol* + *-ic*. The root *alcohol* originates from Arabic *al-kuḥl*, where *al-* is a definite article. It entered English through Latin and then French [R]. Over time, the final segment *-holic* was reinterpreted as an independent morpheme and began to be used productively in new formations such as *workaholic*, *chocoholic*, and *shopaholic*.

1.2 Etymology

Etymology is the study of the origin and historical development of words. It investigates how words were formed, how their meanings and forms have changed over time, and whether they are native or borrowed from other languages.

The word *etymology* comes from Latin *etymologia*, which itself originates from Greek *etymología*, derived from *étymos* (meaning "correct" or "truthful") and *-logia* ("study" or "science"). The term was already used by ancient Greek philosophers in discussions about whether words truthfully describe the meaning of the things they denote [R].

In the following subsections, we will take a closer look at the languages of the world and their evolution, which will help us better understand how languages influence one another.

1.2.1 Languages of the World

The source of information for this subsection is primarily from the chapter on this topic in the *Czech Etymological Dictionary* by Rejzek (2019).

There are an estimated 3,000 to 7,000 languages spoken worldwide. The wide range in this estimate exists because the distinction between a language and a dialect is often unclear.

Some languages are so similar that people who speak them can understand each other quite easily, even though they are officially classified as separate languages. This often has more to do with politics, history, or culture than actual linguistic differences. For instance, Serbian, Croatian, and Bosnian are very close, but they are treated as separate languages mostly for political reasons.

On the other hand, some dialects of a single language can be so different that even native speakers struggle to understand one another. An example of this is Mandarin and Cantonese, which are both considered dialects of Chinese but differ significantly in spoken form.

Additionally, the exact number of languages cannot be precisely determined, as there are still remote regions of the world—such as the Amazon rainforest, parts of Africa, and isolated Pacific islands—that contain undocumented or barely studied languages.

To better understand the relationships and similarities between languages, linguists classify them into language families—groups of languages that evolved from a common ancestor. By studying these families, we can better understand how modern languages have evolved over time.

1.2.2 Language groups

There are around 146 recognized language families, this number depends on the granularity with which we divide languages into family groups. The largest among them are (Eberhard et al., 2025b):

- **Niger-Congo** (1,537 languages, ~612 million speakers)
- **Austronesian** (1,225 languages, ~328 million speakers)
- **Trans-New Guinea** (476 languages, ~3.8 million speakers)
- **Sino-Tibetan** (457 languages, ~1.4 billion speakers)
- **Indo-European** (446 languages, ~3.3 billion speakers)
- **Afro-Asiatic** (377 languages, ~633 million speakers)
- **Other language groups** (2,646 languages, ~1.1 billion speakers)

We will focus more on the Indo-European language family group.

Indo-European

The Indo-European language family is a group of languages that are believed to have evolved from a common ancestor, *Proto-Indo-European*. The earliest speakers of this language probably lived between approximately 4000 and 3000 BCE in what is now Ukraine and surrounding regions.

This language group further branches into many subgroups, including Italic, Germanic, Slavic, Hellenic, Anatolian, Baltic, Celtic, Tocharian, Indo-Iranian, and others (Olander, 2022).

We will explore the branches that have had the greatest impact on modern European languages, especially those that shaped Czech and English.

- **Italic**

  From the Italic language group, Latin had the greatest impact on other languages. The Romance languages evolved from Latin and are traditionally divided into two main groups:[1]

  - **Western Romance**: Spanish, French, Portuguese, Catalan, Italian
  - **Eastern Romance**: Romanian, Dalmatian[2]

- **Slavic**

  The Slavic branch is further divided into three groups:

  - **West Slavic**: Czech, Slovak, and Polish.
  - **East Slavic**: Russian, Ukrainian, and Belarusian.
  - **South Slavic**: Serbian, Croatian, Bulgarian, Slovenian, and Macedonian.

- **Germanic**

  The Germanic branch is further divided into three groups:

  - **West Germanic**: German, English, Dutch, Afrikaans, Frisian, Yiddish
  - **North Germanic**: Swedish, Danish, Norwegian, Icelandic
  - **East Germanic**: Gothic (extinct)

Italic language group

The most influential language from this group is Latin. Originally, Latin was spoken in a small region around Rome. The oldest known inscriptions date back to the seventh - fifth centuries BCE (Rejzek, 2019).

Alongside Classical Latin, a spoken simpler variety known as Vulgar Latin developed, which later became the foundation for the Romance languages.

Latin is no longer a natively spoken language, but it is still widely used in fields such as medicine, law, and science. Additionally, Latin remains one of the formal languages of the Vatican and is still used in the Roman Catholic Church.

[1] Lists are not exhaustive

[2] Dalmatian became extinct in the 19th century. It was spoken along the coast of present-day Croatia.

Latin had a profound impact on most European languages due to the expansion of the Roman Empire and the spread of Christianity across the continent. Although many languages were not influenced by Latin directly, in many cases, this influence came through an intermediate language. Examples include the influence on English through French and on Czech through German.

Evolution of Czech language

Czech belongs to the *West Slavic branch*. Over time, it gradually began to separate and develop distinct characteristics. Significant linguistic changes occurred in the 10th century, and by the beginning of the second millennium, we can begin to refer to the emerging language as *Proto-Czech* (Kosek, 2017).

- **11th–12th century: Proto-Czech** - From this period there are not any written documents.
- **12th–15th century: Old Czech** - The first complex Czech texts appear in this period, with significant literary expansion in the 14th century.
- **16th–18th century: Middle Czech**

  - **16th–early 17th century: Humanist Era** – A period of language refinement, as Czech scholars tried to make the language more elegant, following the model of Latin.

  - The form of Czech from this time was later used by J. Dobrovský as the basis for written Czech during the National Revival, which created lasting differences between spoken and written Czech (Rejzek, 2019).

  - **Mid-17th–18th century: Baroque Era** – A time of decline for the Czech language. Due to political and historical events, German became the dominant language in administration and education, while Czech was spoken mostly informally and on the countryside.

- **Late 18th century: New Czech** - Developed as a reaction to the decline of Czech in previous centuries. - During the *Czech National Revival*, scholars and writers worked to standardize and revive the language.

Germanic Language Group

This section is primarily based on the chapter *Germanic Languages* from the Czech Etymological Dictionary (Rejzek, 2019). Only relevant parts were selected, translated from Czech, and slightly rephrased.

The earliest written records of Germanic languages are runic inscriptions dating from the 2nd to the 6th century. We will focus on the West Germanic branch, which includes languages such as German and English.

German began to develop between the 5th and 11th centuries from a variety of dialects. The oldest written records in German date back to the 8th century. Traditionally, German dialects are divided into:

- *High German* – spoken in the south, it became the foundation of modern written German.

- *Low German* – spoken in the north, it evolved from Saxon and shares a common origin with Dutch and English.
- *Franconian dialects* – had a significant influence on the development of German.

Old English, also called Anglo-Saxon, evolved from the same base as Low German. This was due to the migration of the Saxons and Angles to the British Isles between the 5th and 7th centuries.

Later, English was influenced by Scandinavian languages due to Viking presence in England. However, the most significant external influence came with the Norman Conquest in 1066, which introduced a large number of French words and also affected the grammar of English.

Modern English

Nowadays, English has two main varieties: American and British. They differ slightly in spelling, with American English using simpler forms (e.g., *color* vs. *colour*), as well as in vocabulary and pronunciation.

In recent decades, the global influence of English has been growing, establishing itself as the modern *Lingua franca*. Among its two main varieties, American English appears to be more widespread than British English, though measuring this precisely is difficult. The distinction between them is not always clear, as many non-native speakers mix elements of both in their usage.

Historically, British English had a stronger global presence. Even today, in Europe and many other parts of the world, British English is the standard taught in schools. However, as people engage more with international media, technology, and online content, they are often exposed more to American English.

The internet is strongly influenced by American English, shaped by multiple factors. It is widely used in global entertainment, including movies, TV, and music, and is also more common on social media platforms like Instagram, Facebook, and X (formerly Twitter), where a major proportion of content is in American English rather than British. Additionally, many major technological companies are based in the USA, which contributes to the widespread use of American English in technology and digital communication.

In recent years, the presence of American English has increased even further due to large language models (LLMs), which are primarily trained on internet sources, most of which are written in American English. This trend is likely to continue as the use of AI models grows, exposing more people to American English rather than British English.

1.2.3 Influence of Cultures and Language Contact

When two nations interact over a long period, especially as neighbors, their languages often influence each other. This is particularly true when there are strong ties through trade, science, religion, or politics. Throughout history, larger and more influential nations have shaped the languages of smaller surrounding ones, often leaving a lasting impact.

One example is the expansion of the Roman Empire, which spread Latin across much of Europe. Over time, Latin gave rise to the Romance languages and influenced many other linguistic groups, including Germanic and Slavic languages.

The Czech language has also been shaped by historical contact with German, especially during its long association with the Holy Roman Empire and later the Austro-Hungarian monarchy. Many German words entered Czech, particularly in areas like administration, trade, and urban life, and many of these loanwords are still used today.

Language contact is not just a historical phenomenon—it continues to shape languages today. Globalization, migration, and the dominance of English in international communication have led to the borrowing of many English words into other languages, including Czech.

Indirect borrowings

For borrowing to occur, there must have been some form of contact between the languages. This is why, for example, English could not have borrowed words directly from Greek—by the time the English language was forming, the period of Greek cultural dominance had already passed.

Loanwords from languages without direct contact can still enter a language, typically through an intermediary language. This process is known as **indirect borrowing**. A word may pass through several languages before reaching its final form in the recipient language, with each transfer potentially modifying its pronunciation, structure, or meaning.

To illustrate this, consider the word *admiral*, which is ultimately of Arabic origin. The Arabic word *amīr* (commander), combined with the definite article *al* in expressions such as *amīr al-mā* (commander of the fleet) or *amīr al-baḥr* (commander of the waters), was adopted into Latin as *admirallus*. From Latin, it passed into French as *amiral*, and was later borrowed into English and German, eventually making its way into Czech as well [R].

1.3 Etymology of Morphemes

Sometimes, only part of a word is borrowed from another language. For example, the Greek prefix *anti-* is used productively in many languages with native roots.

In the word *antivirus*, the prefix *anti-* comes from Greek (*anti-*, meaning "against"), while *virus* originates from Latin (*vīrus*, meaning "poison, slime, venom") [R].

The word *antivirus* is a hybrid, combining a Greek-derived prefix with a Latin root. In Czech, adding the native suffix *-ový* yields *antivirový* (antiviral). To determine the complete etymological origin of such words, we must break them down into their individual components. This is the core idea behind *morpheme-level etymology*—it allows for a more fine-grained analysis. While some words are entirely native or fully borrowed, many are composed of morphemes from multiple sources.

Other examples of words with mixed etymology include *television*, *sociology*, and *hyperactive*. The word *television* combines the Greek prefix *tele-* with the

Latin root *vision*. Similarly, *sociology* is formed from the Latin *socius* and the Greek *-logy*. The word *hyperactive* follows the same pattern, with the Greek prefix *hyper-* and the Latin root *active*.

One Czech example is *kopírovat* ("to copy"), which originates from the German *kopieren*, itself derived from the Latin *copiare*. The root *kop-* is borrowed from Latin, *-ír-* reflects the German verb-forming element *-ieren*, and *-ovat* is a native Czech verb-forming suffix.

1.3.1 Calques

Sometimes a word is not borrowed in its original form, but instead its structure or meaning is translated with the use of native morphemes. This process is called a **calquing**, or loan translation. A **calque** is typically a morpheme-by-morpheme translation of a word from another language, transferring meaning without borrowing actual morphemes (Thomason, 2001).

An example from Czech is the word *předseda* ("chairman"), which is a calque of the German *Vorsitzer*, itself based on the Latin *praesidēns*, from the verb *praesidēre*, composed of *prae-* ("before") and *sedēre* ("to sit")—literally meaning "the one who sits in front."

The Czech word follows the same structure, combining *před-* ("before") and *seda*, derived from the verb *sedět* ("to sit").

Another example is *časopis* ("magazine"), a calque of the German word *Zeitschrift*, which comes from *Zeit* ("time") and *Schrift* ("writing"). The Czech equivalent mirrors this by combining *čas* ("time") and *pis* ("writing").

An example borrowed from English is *mrakodrap* ("skyscraper"). It is a calque of the English compound *skyscraper*, made up of *sky* and *scraper*. The Czech version uses *mrak* ("cloud") and *drap* (from *drápat*, "to scrape") — literally something that "scrapes the clouds."

These and many more examples are described in the Czech Etymological Dictionary by (Rejzek, 2019).

Calques and Etymological Ambiguity

With calques, it becomes difficult to clearly determine whether a word should be considered a loan. While the structure and meaning are borrowed, the actual morphological material (the individual morphemes) is native. On the word level, it is reasonable to consider such words as borrowed since they would not exist without the influence of the donor language.

However, on the morpheme level, the situation is more complex. For instance, in the Czech word *časopis* ("magazine"), both morphemes—*čas* ("time") and *pis* ("writing")—are native Czech, both tracing back to Proto-Slavic. Even though the word is a calque of the German *Zeitschrift*, the individual components are not borrowed in form, only in conceptual structure.

In such cases, we do not take the borrowed structure into account when determining the etymology of individual morphemes. Instead, we evaluate each morpheme on its own. If the morphemes are of native origin, we classify them as native, regardless of the fact that the overall word may be a calque or structurally borrowed.

2 Related Work and Data Sources

To the best of our knowledge, there is no existing research in computational linguistics that specifically focuses on identifying the etymological origin of individual morphs. At present, there is no established or widely accepted methodology for this task, nor is there a dataset explicitly designed for morph-level etymology prediction.

Most research on etymological classification has focused on the word level. While linguists are aware of the etymology of individual morphs, this information is typically not explicitly included in etymological dictionaries or is only mentioned implicitly.

There is extensive work in computational linguistics on morphological segmentation and classification, with several tools available that automate these processes. However, we have not found any computational approach that explicitly combines these areas with etymological analysis to study the origin of individual morphs.

2.1 Deconstructor

One related tool that attempts a similar task is the online application *Deconstructor*[1]. It performs morphological segmentation and tries to reconstruct the etymology of individual morphs within a word. It also visualizes how the word may have been formed from its components. The tool supports multiple languages and uses a large language model (LLM) to generate its output.

The interface presents the analysis in a clear, graphical format, showing a step-by-step construction of the word. However, it outputs only a single origin language for each morph, without including any intermediate borrowing stages or full etymological chains.

The tool performs quite well for English. For Czech, however, the results are often less reliable, mainly due to inaccurate morphological segmentation.

It is not possible to directly compare this system to the approaches used in this thesis. Deconstructor is a web-based tool that processes one word at a time and uses its own segmentation and output format, which differs from the morphological segmentation in our dataset.

In the figure 2.1 is an example of output to entry for word *telefonní*.

[1] `https://deconstructor.ayush.digital`

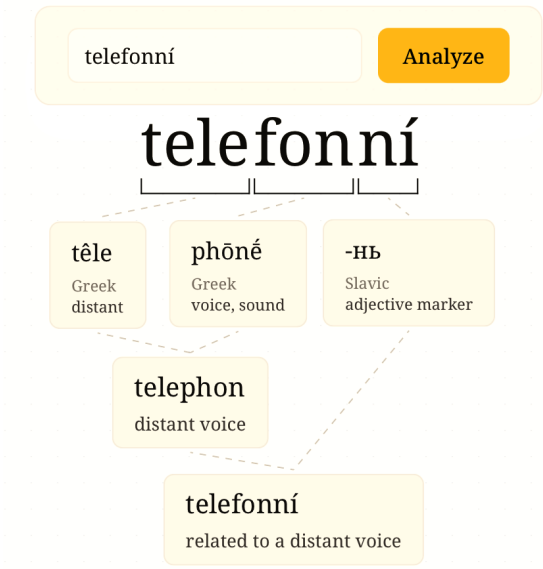

Figure 2.1 Example of the Deconstructor web interface output
Source: `https://deconstructor.ayush.digital`

While the tool itself cannot be directly used in this work, the idea of using a large language model (LLM) for morph-level etymology is promising, as LLMs have proven effective in solving many NLP tasks. This direction will be explored further in later chapters, where we will use a model from OpenAI to perform the task and evaluate it as one of the baseline solutions.

The following sections describe the tools and datasets used throughout this work.

2.2 DeriNet

DeriNet is a large-scale lexical network that models derivational and compositional relations in Czech. Each node represents a lexeme, while edges capture word-formation links, either connecting derived words to their base forms or linking compounds with their components. The dataset is based on the MorfFlex CZ dictionary (Hajič; Hlaváčová, et al., 2024) and includes linguistic annotations such as part-of-speech tags, segmentation, morphological classification, corpus frequency, and etymological information sourced from the Czech Etymological Lexicon (CzEtyL).

The latest version, DeriNet 2.3, to which I also contributed by adding etymological information, was developed by multiple authors from ÚFAL, MFF UK (Olbrich et al., 2025). It comprises 1,040,126 lexemes, 791,771 derivational relations, and 7,598 compound relations. In addition, it includes 5,781 derivational trees containing loanwords, enriched with etymological data from CzEtyL.

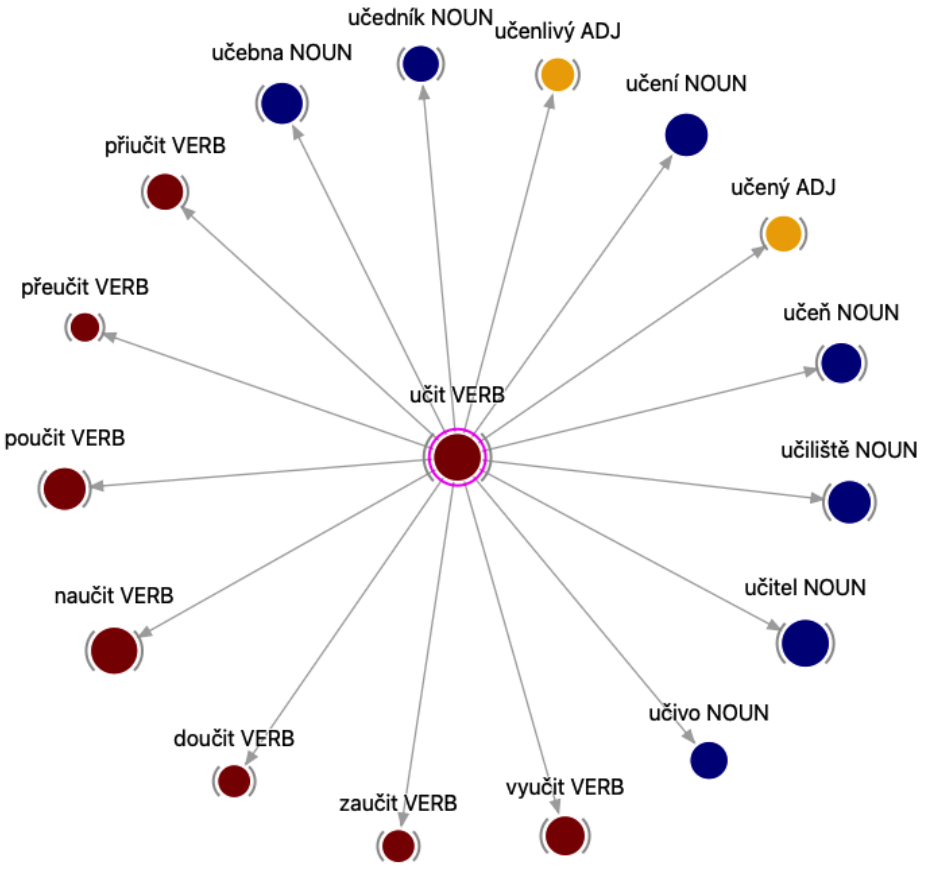

Figure 2.2 Example of DeriNet tree for the word *učit* ("to teach").
Source: `https://quest.ms.mff.cuni.cz/derisearch2/v2/databases/Czech-DeriNet-2.1/dcql`

Figure 2.2 shows an example of a DeriNet tree for the word *učit* ("to teach"), expanded only to its direct derivations. Each of these derivations can itself be the root of a larger subtree. If the entire tree were fully expanded, it would contain over 500 lexemes. Figure 2.3 presents the same tree, but with several additional nodes expanded. For clarity and readability, some parts of the tree are intentionally hidden.

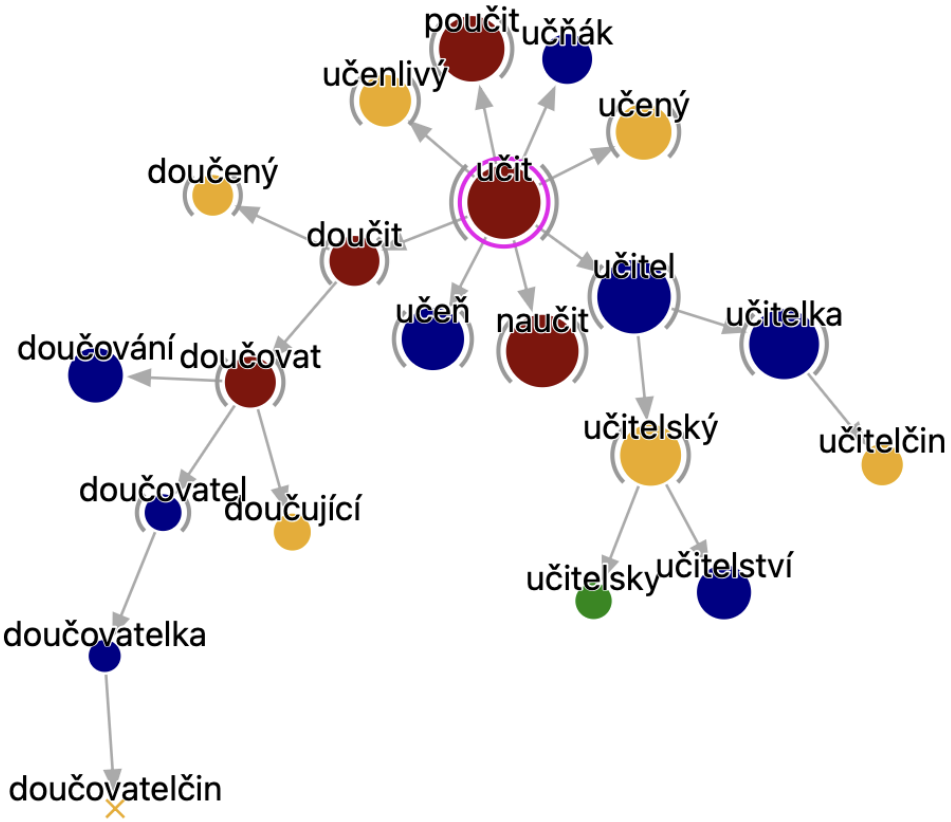

Figure 2.3 Example of partially expanded DeriNet tree for the word *učit* ("to teach").
Source: `https://quest.ms.mff.cuni.cz/derisearch2/v2/databases/Czech-DeriNet-2.1/dcql`

In addition, DeriNet provides detailed morphological segmentation and classification of morphs, making it a valuable resource for this work.

Borrowed words often become bases for new derivations that include native affixes. Even when a native affix is added to a borrowed stem, the resulting word is still considered a loanword. However, if we classify each morpheme individually, some will be identified as borrowed and others as native.

2.3 Czech Etymological Dictionary

The most recent Czech etymological dictionary by Rejzek (2019) contains over 11,000 main entries. In addition, it provides examples of derived words for many entries, totaling around 21,000 derived forms. The dictionary also includes nearly 64,000 references to words in other languages, either as sources of borrowing or to illustrate shared borrowing paths across different languages.

The dictionary also features an introductory chapter on general etymology, the development of world languages, and the evolution of Czech from its Indo-European roots to its modern form. It includes a classification of related languages across the world.

2.3.1 SIGMORPHON Shared Task

SIGMORPHON (Special Interest Group on Computational Morphology and Phonology) regularly organizes shared tasks to support progress in morphological analysis. This thesis uses data from the *SIGMORPHON 2022 Shared Task on Morpheme Segmentation* (SIGMORPHON, 2022).

The dataset for this task was constructed by integrating several major morphological resources, including UniMorph for inflectional morphology (McCarthy et al., 2020), MorphyNet for derivational morphology (Batsuren et al., 2021), Universal Dependencies (Nivre; Agić, et al., 2017), and multiple editions of Wiktionary for compounds and root words (Wiktionary, 2025).

This work focuses on the Czech portion of the dataset, which contains full sentences segmented into individual morphemes. These annotations provide the morphological basis for the etymological task addressed in this thesis.

2.4 Segmentation

Morphological segmentation is the task of splitting words in a sentence into individual morphemes. This thesis uses data provided by the *SIGMORPHON 2022 Shared Task on Morpheme Segmentation*, which focuses specifically on this task. In this work, the manually segmented sentences from the shared task are used.

If other datasets were to be used, a morphological segmentation step would be required beforehand, as morpheme-level etymology prediction depends on having the words already segmented into morphemes.

2.5 Morpheme Classification

Classification of morphemes serves as a valuable feature for morpheme-level etymology prediction. As discussed in Section 1.1.2, there are several ways to categorize morphemes. In this work, the primary distinction is between *roots* and *affixes*.

Affixes are further categorized based on two criteria: their function—either *derivational* or *inflectional*—and their position relative to the root within the word. According to position, affixes are labelled as *prefixes*, *suffixes*, or *interfixes*.

The morphological classification used in this thesis was generated automatically using a model developed by John (2024).

3 Data and Evaluation Methodology and Baselines

In this chapter, we describe the data used and created for this work and explain how the manually annotated dataset was constructed. We then define the evaluation methodology used to measure the quality of model predictions. After that, we present inter-annotator agreement as a reference point for human-level consistency. Finally, we introduce several baseline methods for etymology prediction to establish a lower bound for model performance.

3.1 Data

Currently, there is no publicly available resource that provides words annotated with etymological information at the morpheme level. The Czech Etymological Dictionary by Rejzek (2019) is a valuable source at the word level, and its digital form, CzEtyL (Rejzek et al., 2025), is suitable for automated processing.

3.1.1 CzEtyL

The Czech Etymological Dictionary by Rejzek (2019) is a rich and detailed resource, but it is primarily intended for human readers and not designed for computational use. The structure of entries and the categories of information provided are not always consistent or machine-readable.

To address this, the Czech Etymological Lexicon (**CzEtyL**) (Rejzek et al., 2025) was created in collaboration with the original author of the dictionary and researchers from MFF UK, ÚFAL—including myself as one of the co-authors. This digitalized version extracts the essential etymological information from the original dictionary and stores it in a simple format.

CzEtyL focuses on identifying the source languages from which a given Czech word was borrowed, omitting other details such as the original form in the source language, the time of borrowing, or cross-references to similar words. Version 1.0 of the dataset includes approximately 10,500 Czech words, each annotated with a sequence of ISO 639-3 language codes that represent the etymological path into Czech.

The data is organized in a tab-separated format with three columns:

- **First column:** Lists the lemma.
- **Second column:** Provides the corresponding language codes, separated by commas.
- **Third column:** Specifies whether the word is classified as a loanword ("loan") or a native word ("native"). In this classification, "native" refers to words that have naturally evolved in the language rather than being borrowed from another.

Example entry:

```
architekt   deu,lat,ell   loan
```

The word *architekt* originated from Greek and entered Czech through Latin and German.

The morphemes are derived from Greek:

- **Archi-** – meaning "main, leader," from *árkhō* ("I command")
- **-téktōn** – meaning "craftsman, artist"

Affix entries

Although the lexicon is word-based, it does include a few examples of affixes with annotated origins. For instance:

- Prefixes:
  - Greek prefixes: *aero-* (*aerodynamika*), *anti-* (*antivirus*), *astro-* (*astronomie*), *elektro-* (*elektromagnetismus*)
  - Latin prefixes: *ab-* (*abdikace*), *ad-* (*administrace*), *dis-* (*disfunkce*), *per-* (*perforace*), *re-* (*rekonstrukce*)
- Suffixes:
  - English: *-bal* (*fotbal*, *handbal*)
  - Latin: *-ace* (*rekreace*, *prezentace*), *-iz* (*organizace*, *realizace*)
  - Czech: *-náct*, used in numerals from eleven to nineteen (e.g., *jedenáct*, *devatenáct*)

Expanded Version

The expanded version of CzEtyL builds on the original lexicon by incorporating derivational relations from DeriNet. For each entry, all lexemes that belong to the same derivational tree are included and assigned the same etymological label. This significantly increases coverage, expanding the number of annotated words from approximately 10,500 to around 511,000.

Root and Affix Dictionaries

Using DeriNet, I extracted morphological segmentation and morpheme classification for all words present in CzEtyL. This resulted in a dataset of approximately 10,500 words, each annotated with segmentation, morpheme classification, and word-level etymology.

Assuming that the etymology provided in CzEtyL corresponds to the root of the word, we can construct a dictionary of roots and their etymological origins by assigning the etymology of the whole word to its root morpheme. While this approach is a simplification, it offers a rough but practical approximation of morpheme-level etymology for a large number of root morphemes.

CzEtyL also includes a list of approximately *250* affixes with known etymologies, covering both borrowed and native affixes. Together with the root dictionary constructed from CzEtyL and DeriNet, this serves as a useful basis for simple baseline predictions or as an extension to the training data.

3.1.2 Manually Annotated Dataset

To create a dataset of sentences annotated with etymological information at the morpheme level, I used data from the SIGmorphemeON 2022 Shared Task. This dataset contains sentences that are already segmented into individual morphemes. I then manually annotated each morpheme with a sequence of language origins. The annotations were based primarily on information from the Czech Etymological Dictionary by Rejzek (2019), with additional reference to other etymological sources like Wiktionary (2025) when needed.

I divided the annotated dataset into three parts: training, development, and test sets. Table 3.1 provides an overview of each subset, including the number of sentences, words, and morphemes it contains.

Dataset	**Sentences**	**Words**	**morphemes**
Training set	200	2,774	7,016
Development set	50	599	1,460
Test set	50	609	1,485

Table 3.1 Size of the annotated dataset used for training, development, and testing

Only morphemes that are candidates for etymological classification are counted. This excludes punctuation, numerals, abbreviations, and special symbols.

The training sentences were selected from the Sigmorphemeon shared task train set for Czech, while the development and test sets were derived from the original development set.

The training data consists of approximately 170 sentences from the beginning and about 30 sentences from the end of the original training file.

The development and test sets were selected from the development file, which contains 500 sentences in total. Every 10^{th} sentence (1st, 11th, 21st, ...) was assigned to the test set, and every 10^{th} sentence starting from the second (2nd, 12th, 22nd, ...) was assigned to the development set.

The remaining 400 sentences from the development file were not used in this work and are kept for possible future evaluation or additional training data. Similarly, the 500 sentences from the official test file were also left unused.

It would be beneficial to have more annotated sentences available for both training and evaluation. However, the annotation process is time-consuming and requires either a certain level of expertise or considerable effort spent searching for etymological information in various sources. This naturally limits how much annotated data can be realistically produced.

Example annotation:

Sentence: *Faxu škodí především přetížené telefonní linky*

- **Faxu**
  - *Fax* — `R` — `eng,lat`
  - *u* — `I` — `ces`

- **škodí**
  - *škod* — R — gmh
  - *í* — I — ces
- **především**
  - *přede* — D — ces
  - *vš* — R — ces
  - *í* — I — ces
  - *m* — I — ces
- **přetížené**
  - *pře* — D — ces
  - *tíž* — R — ces
  - *en* — D — ces
  - *é* — I — ces
- **telefonní**
  - *tele* — R — ell
  - *fon* — R — ell
  - *n* — D — ces
  - *í* — I — ces
- **linky**
  - *lin* — R — deu,lat
  - *k* — D — ces
  - *y* — I — ces

The annotation uses the following abbreviations:

- R – Root
- D – Derivational affix
- I – Inflectional affix

  Language codes follow ISO 639-3:

  - ces – Czech
  - deu – German
  - ell – Greek
  - lat – Latin
  - eng – English

- `fra` – French
- `gmh` – Middle High German
- ...

The morpheme types (root, derivational affix, inflectional affix) are included in the dataset and were obtained by automatic prediction. The language origin sequences were manually annotated.

3.2 Evaluation Methodology

The goal is to predict, for each morpheme, a sequence of languages starting from the original source language, through any intermediate languages, before getting to Czech. Since the order of languages in the sequence is usually fixed or clear from context, the prediction is evaluated as an unordered set. For example, if the correct languages are Latin and German, it is almost always the case that the borrowing path was from Latin through German, not the other way around, so the order is not considered in the evaluation.

3.2.1 F1-Score

To evaluate the quality of predictions, we use the F1-score, which provides a more balanced evaluation than simple accuracy. We don't want to just check whether the prediction is exactly the same as the target or not—we also want to measure how close it is, even if it's only partially correct. It's important that the model predicts as many correct languages from the sequence as possible (recall), but also avoids adding incorrect ones (precision). Because both of these aspects are equally important for our purposes, we decided to use the F1-score as the main evaluation metric.

We calculate the F1-score for each individual morpheme occurrence and then take the average across all morphemes in the dataset. This means that morphemes which appear more frequently have a bigger influence on the final score.

Alternatively, we could compute micro F1-score by summing up the total number of correct predictions, total predicted languages, and total gold labels across all morphemes—and calculating precision, recall, and F1 from those aggregate counts.

As another approach, we can split the dataset into two subsets—native and borrowed morphemes—and compute separate F1 scores for each group. This provides a more detailed view of how the model performs across different categories of morphemes, especially since borrowed ones are generally more challenging to classify due to their lower frequency and more complex etymological paths.

3.2.2 Native vs. Borrowed

Because the majority of morphemes in the dataset are native, computing the F1-score across all morpheme instances can lead to results biased toward this dominant class. To address this imbalance and gain deeper insight into model behavior, we additionally evaluate performance separately for two categories of morphemes: native and borrowed.

We define native morphemes as those whose target etymology is `ces`, representing morphemes of Czech or Slavic origin. All other morphemes are considered borrowed. By reporting F1-scores for each group independently, we can better observe how well the model performs on native morphemes and how well on borrowed morphemes. Borrowed morphemes typically present a greater challenge for the model, as they can originate from a wide variety of languages and may follow complex borrowing paths involving multiple intermediate languages.

3.2.3 Score Grouped by morphemes

In the dataset, many morphemes appear multiple times—on average about 7 times—but some affixes, especially inflectional endings, occur dozens of times. In the training set, there are 7,205 total morphs, of which only 972 are unique. The 10 most frequent morphs alone account for 1,965 occurrences; the top 20 cover 2,864, and the top 50 together make up 3,997 occurrences—more than half of the dataset.

To reduce the bias introduced by these highly frequent morphs, we also report an additional metric where morphs are grouped by their surface form. For each unique morph, we compute the average F1-score across all of its occurrences. These per-morph scores are then averaged to obtain the final result.

3.2.4 Relative error reduction

When the score of the baseline solutions is high, absolute values of the F1 score or accuracy do not fully reflect the improvements made by better models. A model might outperform the baseline by only a few percentage points, even though it significantly reduces the number of actual errors. To better highlight these improvements, we report the *relative error reduction* compared to the baseline.

The formula for computing the relative error reduction is:

$$\text{Error Reduction} = \frac{\text{Error}_{\text{baseline}} - \text{Error}_{\text{model}}}{\text{Error}_{\text{baseline}}}$$

where error is defined as $1 - \text{F1-score}$.

3.2.5 Expected Bounds of Performance

To properly interpret the results of this task, it is important to establish both a lower and an upper bound—defining an interval within which realistic performance can be expected. The lower bound is represented by simple baselines such as always predicting the most frequent class, memorizing morphemes from the training data, or applying basic rule-based methods using available etymological sources.

The upper bound, on the other hand, is more difficult to define. A model with 100 % error reduction would be perfect; however, such performance is not achievable. Even a human annotator can struggle to consistently determine the correct etymology of morphemes. Without access to reference materials like etymological dictionaries, most people would not perform better than the simpler baselines.

Problems with annotation

Even when such resources are available, inconsistencies or differing interpretations can occur, especially in complex or ambiguous cases. Etymological dictionaries may sometimes offer conflicting explanations, or lack coverage for certain words.

For example, in the Czech Etymological Dictionary (Rejzek, 2019), the entry for *bendžo* ("banjo") provides two possible etymological explanations. One interpretation traces the word as a Black English modification of Old English *bandore* or Spanish *bandurria*, ultimately derived from Late Latin *pandura* and Greek *pandoũra*. The second explanation suggests that the term may have originated from an African language.

The author of the Czech Etymological Dictionary (Rejzek, 2019), which serves as the primary source in this work, often presents multiple possible etymological explanations for ambiguous words. In such cases, he also refers to interpretations given in other etymological dictionaries such as Holub and Kopečný (1952) and Machek (1968).

Etymology is often uncertain, especially when it comes to the path a word or morpheme took through multiple languages before entering the target language. While the goal of this work is to annotate the full borrowing chain—including intermediate languages, not just the ultimate source—such detailed information is not always available. Many etymological resources focus only on the original source, and in many cases, we simply do not know which languages served as intermediaries.

Furthermore, some words have been borrowed simultaneously from different languages, making it difficult to trace a single, clear etymological path. This is especially true for modern scientific and technical vocabulary, which was often created in the 19th or 20th century and then adopted across multiple European languages. For instance:

- *ozon* – introduced into Czech in the 19th century from modern European languages (e.g., German *Ozon*, French and English *ozone*), originally derived from Ancient Greek *ózōn* {giving off a smell}, from the verb *ózō* {to smell}.

- *virus* – entered Czech in the 20th century through modern European languages (German *Virus*, French and English *virus*), which in the 19th century had reintroduced the term from Latin *vīrus* {poison, slime, toxic substance}.

- *turbulence* – entered Czech in the 20th century from modern European languages (e.g., German *Turbulenz*, French *turbulence*), where it was coined based on Late Latin *turbulentia* {disorder, agitation}, derived from Latin *turbulentus* and ultimately from *turba* {crowd, confusion}, *turbāre* {to disturb}.

These examples from Rejzek (2019) show how some words were created in a shared scientific or cultural context and then adopted by many languages at roughly the same time. Because of that, it's hard to tell the full borrowing path.

Identifying the etymological sequence at the morpheme level is even more challenging. While linguistic experts would be able to assign origins to individual morphemes, this kind of information is rarely stated explicitly in etymological

dictionaries or linguistic studies, which typically focus on whole words. As a result, there are no standard guidelines for morpheme-level annotation, and decisions sometimes come down to personal interpretation. This makes the annotation process inherently difficult and ambiguous, and currently, there is no existing dataset dedicated to this task.

3.3 Inter-Annotator Agreement

Inter-annotator agreement measures how consistently different annotators label the same data. It provides a useful way to assess both the subjectivity of the task and the reliability of the annotation process.

This is particularly relevant in the context of etymological annotation for individual morphemes, which can be unclear. Measuring agreement can help identify borderline or problematic cases and ensure greater consistency across the dataset.

Moreover, inter-annotator agreement serves as a practical upper bound for the performance of automatic models. If human annotators cannot consistently agree on the correct etymology of a morpheme, it is unrealistic to expect a model to achieve significantly better accuracy.

Cohen's kappa

One commonly used metric for measuring inter-annotator agreement is *Cohen's kappa*, introduced by Cohen (1960). It evaluates the agreement between two annotators assigning categorical labels to a dataset.

Unlike simple percentage agreement, Cohen's kappa also accounts for the agreement that might occur purely by chance, which is especially important when one category is much more frequent than the others.

The formula for computing Cohen's kappa is:

$$\kappa = \frac{p_o - p_e}{1 - p_e} \tag{3.1}$$

where:

- p_o is the observed agreement between the annotators,
- p_e is the expected agreement by chance.

The expected agreement p_e is calculated based on the marginal probabilities for each category assigned by the annotators:

$$p_e = \sum_{i=1}^{k} p_i^{(1)} \cdot p_i^{(2)} \tag{3.2}$$

where:

- k is the number of categories,
- $p_i^{(1)}$ is the proportion of items that annotator 1 assigns to category i,
- $p_i^{(2)}$ is the proportion of items that annotator 2 assigns to category i.

The value of Cohen's kappa ranges from -1 to 1. A higher value indicates stronger agreement between the two annotators, with 1 representing perfect agreement and 0 corresponding to agreement equal to chance. Negative values suggest less agreement than would be expected by chance. However, there is no universally accepted interpretation of specific kappa values, and their evaluation often depends on the context and nature of the task.

3.3.1 Results of inter annotator agreement

Annotation on the development set was also performed by a second annotator, a final-year high school student with no prior experience in linguistic annotation. Before beginning, he was introduced to the task and provided with a thorough explanation of the annotation scheme. He was given access to the Czech Etymological Dictionary by Rejzek (2019) and directed to additional resources such as Wiktionary[1]. Although not a trained linguist, his annotation offers a valuable reference point for estimating human performance on this task.

We compute Cohen's kappa and the percentage of exactly matched annotations.

- Cohen's kappa: 0.82
- Exact match: 95.96%

These results show a high level of agreement between the two annotators. The differences were sometimes caused by clear mistakes made by one of the annotators, which, once corrected, helped improve the overall quality of the dataset. In other cases, the disagreement arose from differences in the granularity of annotation. For example, one annotator might have labelled some morph as `gmh` (Middle High German), while the other used the more general `deu` (German).

There were also cases where inconsistencies between different etymological sources led to disagreement. This highlights the difficulty of the task—even professional linguists do not always agree on the etymology of certain words. In this work, the challenge is even greater, as the goal is to determine etymology at the level of individual morphemes rather than whole words.

3.4 Baselines

To evaluate how well the model performs, we first need to define a few baselines. These serve as reference points, helping us understand whether the model actually learns something useful or if similar results could be achieved using much simpler approaches.

In this work, we define several baselines: a trivial one that always predicts native origin (Czech), a memorization-based baseline, a word-level lemmatization approach using CzEtyL, and a root-based approach using CzEtyL.

[1] `https://www.wiktionary.org`

3.4.1 Always-Czech Baseline

A very simple baseline is to always predict the most frequent target (Czech) as the etymological origin for all morphs. Despite its simplicity, this approach gets a high score, which is due to the fact that the majority of morphs in the dataset are of Czech origin.

3.4.2 Memorization-Based Approach

This approach simply memorizes all morphs seen during training and assigns them their most frequent etymology. If a morph appears in the test data and was seen in training, its stored etymology is used. In cases where a morph was annotated with multiple different etymologies in training, the most frequent one is selected.

This method can perform surprisingly well, especially when the training data covers a large portion of the vocabulary in the test set.

However, its main limitation is handling unseen morphs. If a morph is not present in the training data, the system defaults to predicting the most frequent class, here the native Czech label.

3.4.3 Word Lemmatization Approach

This baseline uses a morphological analyser to obtain the lemma for each word, which is then looked up in the expanded version of CzEtyL. The retrieved etymology is assigned to the root morph. We use the *MorphoDiTa* morphological dictionary and tagger by Straková et al. (2014).

Affixes are matched against a list of known borrowed affixes from CzEtyL; those not found in the list are assumed to be native. Inflectional endings are always treated as Czech. If a word's lemma is not present in CzEtyL, the default prediction is Czech.

3.4.4 Morpheme-Based Baseline

For this baseline, we assume that the etymology provided in CzEtyL corresponds to the root of the word. Affixes are considered native unless they appear in a predefined list of known borrowed affixes, which is also part of CzEtyL. All inflectional affixes are assumed to be native by default.

To build this baseline, we iterate through all words in CzEtyL, extract their roots, and assign the full word-level etymology to those root morphemes. This produces a mapping from root morphemes to a multi-set of possible etymological origin sequences based on their occurrences.

The algorithm then applies the following logic:

- If the morpheme is classified as a root, it is assigned the most frequent origin sequence from its associated multi-set.
- If the morpheme is a derivational affix, we check whether it appears in the list of known borrowed affixes and assign its origin accordingly.
- For inflectional affixes, or in cases where the morpheme is not found in the root or affix dictionaries, we default to Czech (`ces`), the most frequent class.

3.4.5 Using Large Language Model

Large language models (LLMs) have demonstrated the ability to solve a wide range of NLP tasks without requiring additional training on the specific task. To explore their potential in morpheme-level etymology prediction, we experimented with OpenAI's `o3` reasoning model.

The model was prompted to act as an etymological annotator for Czech morphemes, assigning ISO 639-3 language codes to each morpheme to indicate its etymological origin and the languages through which it entered Czech. The complete prompt used in the experiment is presented below:

Prompt

- You are an etymological morph annotator for Czech, assigning ISO-639-3 origin codes to each morph.
- For each morph in the input, append a language code indicating its origin and the languages through which the morph entered Czech.
- The codes are based on the ISO-639-3 standard.
- Use the provided training data to learn patterns and rules for assigning these codes.
- Use `ces` if the morpheme is native Czech or evolved from Old Czech/Slavic.
- If multiple source languages apply, separate the codes with a comma (e.g., `deu,lat`).

- **Input Format:**
  - Sentence text.
  - Indented word lines.
  - Under each word, a list of morphemes is shown with a tab-separated tag:
    - D – derivational affix
    - I – inflectional affix
    - R – root
- **Example Output Format:**

```
Jak konstatoval premiér
    Jak
        Jak R ces
    konstatoval
        kon D lat
        stat R lat
        ova D ces
        l D ces
```

```
premiér
    prem R fra,lat
    iér D fra,lat
```

- **Rules:**
  - Preserve all original text, indentation, morphs, and D/R/I tags.
  - Add exactly one tab character, followed by the ISO 639-3 code(s).
  - Use `ces` for native Slavic/Czech forms.
  - Do not add any other columns or comments.
  - Numerical values are not to be annotated.

- **Hints:**
  - Majority of the morphs in the data are native (`ces`).
  - Inflectional affixes are almost always native.

- **Learning and Prediction:**
  - Use `train.tsv` as data for learning. Use also your own knowledge to make predictions.
  - Predict on `dev.tsv`.
  - Take time to analyse before making predictions.

We tried out several different prompts, and the one shown above gave us the best results. Telling the model that most morphs are Czech turned out to be really important—without that, it often assigned some non-Czech origins to many native morphs. Also, when we didn't include the training data as examples, the model's predictions got noticeably worse, since it didn't have any clear patterns to follow.

4 Model for Morpheme-Level Etymology Prediction

This chapter describes the approaches explored to develop a model capable of predicting the etymological origin of individual morphemes in Czech. The goal is to assign each morpheme a sequence of languages from which it was borrowed into Czech.

4.1 Features

To successfully predict the etymological origin of each morpheme, it is essential to extract useful features from the annotated data. These features serve as input to the classification model and help it learn patterns associated with language origin.

Several types of features were used in this work, each capturing different aspects of the morpheme and its context. These include character-level n-grams, morphological classification, position within the word, and others. Additionally, abstract representations such as morpheme and word embeddings were also tested.

The following subsections describe each feature type in more detail.

4.1.1 Morph n-grams

From the text of each morph, we extract character n-grams and convert them into sparse vectors. This is done by building a vocabulary of all n-gram combinations seen during training. Then, for each morph, a vector is created where each dimension counts how many times a specific n-gram from the vocabulary appears in the morph.

The idea behind using character n-grams is that different languages tend to favor specific letter combinations, and the model can learn to associate these patterns with particular language origins. In this work, we focus on 1-grams and 2-grams.

Using 3-grams would increase the dimensionality of the feature space, and since the average morph length is only about 2.2 characters, most morphs do not even contain three characters, making 3-grams less useful in practice.

One drawback of this method is that morphs containing character combinations unseen during training will result in a zero vector. In such cases, the model lacks meaningful input for prediction.

However, this is less of a problem with enough training data. The model may also learn to default to the most frequent class when encountering unknown n-grams or rely more heavily on other features.

4.1.2 Morph Types

Each morph is classified into one of three basic categories: *Root*, *Derivational affix*, or *Inflectional affix*. This classification is encoded using a one-hot representation.

In addition, a positional classification is also extracted. Each morph is categorized as either *Root*, *Prefix*, *Suffix*, or *Interfix*, depending on its position relative to the root(s) in the word. Typically, a word contains a single root, but in the case of compounds, multiple roots may be present. Affixes that appear before the first root are labelled as prefixes, those that come after the last root as suffixes, and those occurring between two roots are labelled as interfixes.

In rare cases, there are words that do not contain any identifiable root. This situation often occurs with certain prepositions or conjunctions, which may consist of only a single morpheme, making it unclear how to classify them.

There are also examples of multi-morpheme words that lack a root entirely. For instance, the words přední and zadní ("front" and "rear") are formed from two affixes: před- / zad- and the adjectival suffix -ní without a clear root morpheme present.

In these cases, the first (often the only) morpheme is classified as a *root* with respect to the position type, while the remaining morphemes are classified as *suffixes*.

Just approximately 3 % of words in the dataset do not contain a root and consist of more than one morpheme.

4.1.3 Vowel Start and End

This feature captures whether a morph starts and/or ends with a vowel. Two binary values are used: one indicating if the first character is a vowel, and the other if the last character is a vowel.

Some languages tend to favor specific phonological patterns, such as vowel-final or vowel-initial affixes. Even though this is a very simple feature, it can still help the model notice language-specific patterns.

This feature is especially helpful when combined with information about whether the morph is a root or an affix. For example, many Proto-Indo-European roots follow a consonant-vowel-consonant (CVC) structure. On the other hand, affixes often connect to roots by starting or ending with a vowel for better pronunciation. So, prefixes more often end with vowels, and suffixes often begin with them (Gamkrelidze; Ivanov, 1995).

4.1.4 Embeddings

Using embeddings is a way to represent words or sub-words as numerical vectors that capture aspects of their meaning. The core idea is that words with similar meanings are mapped to vectors that are close to each other in the embedding space.

The first widely adopted word embeddings were Word2Vec embeddings by Mikolov et al. (2013). An extension of this approach is FastText by Bojanowski et al. (2017), which works at the sub-word level rather than treating words as atomic units. In FastText, each word is represented as a bag of character n-grams, and its final embedding is computed as the sum of the embeddings of its n-grams. This allows FastText to generate embeddings even for words that were not seen during training.

This property is particularly useful, as it enables the generation of meaningful vector representations for individual morphs. This makes FastText a good fit for the task of predicting morpheme etymology, where we need embeddings for affixes and roots that are not standalone words.

In this work, we use FastText embeddings trained on the Czech language, provided by Grave et al. (2018). These embeddings were trained on Czech Wikipedia and Common Crawl data and have 300 dimensions. To reduce dimensionality and improve how effectively the classification model can learn from these features, the embeddings can optionally be compressed to a lower dimension using Principal Component Analysis (PCA).

4.2 Context

In configurations where word embeddings are not used, the model relies solely on n-grams, morph position, and morph type within the word. It lacks access to broader contextual information from the sentence or the word the morph is in. This limits the model's ability to disambiguate morphs with identical surface forms but different meanings and etymological origins.

For instance, compare *muzeum* ("museum") and *rozum* ("mind, reason"). In *muzeum*, the suffix *-um* is of Latin origin, serving as a neuter singular ending. In *rozum*, by contrast, *um* is the root of Slavic origin, derived from *um* (ability) [R]. In this case, the morph type (suffix vs. root) distinguishes the two.

This problem is even more apparent in morphologically ambiguous or homonymous words. For example, *kolej* can mean either "railway" or "college." The first derives from *kolo* (wheel), a native Slavic root, while the second comes from Latin *collēgium* [R]. Other examples of homonyms with distinct etymologies include [R]:

- *džin* — either the mythological spirit (from Arabic *jinn*) or the alcoholic drink (from French *genièvre*, ultimately from Latin *iūniperus*);
- *golf* — either the sport (related to German *Kolben*, meaning "club") or a sea inlet ("gulf", from Italian *golfo*, Latin *colpus*, Greek *kólpos*).

These cases highlight the potential benefits of incorporating broader contextual information. Doing so could help resolve such ambiguities and improve the quality of etymological predictions.

On the other hand, such ambiguous cases are relatively rare, so even perfect disambiguation would not significantly improve the overall evaluation score. Moreover, it is not clear what kind of contextual representation would actually help improve predictions, since for most morphs, the surrounding context is not necessary to determine their origin. Incorporating context meaningfully without causing overfitting is a non-trivial task and would likely require more complex models.

4.3 Extending the Training Data with Etymological Dictionary

The root and affix dictionaries previously extracted for use in baseline models can also serve as a valuable source of additional training data for the learning model. By adding these entries into the training set, the model has more examples of morphs with known etymology.

The original training set contains 7,205 morphs, of which 972 are unique.

The tables 4.1, 4.2, 4.3 present the most frequent morphs observed in the annotated dataset with the frequency, grouped by their function: roots, affixes, and affixes composed of more than one character.

Root	Frequency
je	41
bank	37
kter	36
stroj	29
fax	25
jedn	22
by	22
klad	19
js	19
tele	18
fon	17
práv	17

Table 4.1 Most common root morphs

Affix	Frequency
n	317
í	297
a	194
u	176
e	158
i	156
y	151
o	134
ě	104
é	99

Table 4.2 Most common affixes

Affix	Frequency
en	94
po	90
ne	60
ou	51
ost	50

Table 4.3 Most common multi-character affixes

After extending the data with dictionary entries, the training set increases to 19,430 morphs, covering 12,211 unique morphs. The dictionary entries consist of roots and affixes, which typically do not repeat, so the added morphs are almost all unique.

4.4 Representation of Target Classes

We used two different strategies for modeling the target classes in the morpheme level etymology prediction:

- **Whole-sequence classification:** The entire sequence of languages is treated as a single label. This reduces the task to a standard classification problem where each unique language sequence is a distinct class. The model predicts exactly one of the predefined sequences (sequence that appeared in the train set) for each morph.
- **Multi-label classification:** Each language is treated as an independent label. The model predicts a subset of languages (from languages which appeared in the train set), determining for each one whether it should be included in the sequence. This allows the model to predict language sequences which were not seen in training.

The first approach is simple but limited to the set of language sequences present in the training data. The second one is more flexible and can produce combinations of languages not explicitly seen during training, but it also increases the complexity of the classification task. We have to train one classifier for each language. This can be viewed as a binary classification for each language, determining whether or not it should be included in the output sequence.

4.4.1 Label Statistics from Training Data

To help choose an appropriate strategy for representing the target labels, it is useful to examine the number of unique language sequences and individual languages present in the training data. In the manually annotated Czech sentences used for training, the following statistics were observed:

- **Unique language sequences:** 35
- **Unique languages:** 11

Top Etymology Sequences

The following table shows the 10 most common etymological origin sequences found in the training data. The dataset contains approximately 7,100 annotated morphs. As shown, the majority of morphs are of Czech origin.

Language Sequence	Count
`ces` (Czech)	6,229
`lat` (Latin)	280
`ell` (Greek)	100
`deu,lat` (German, Latin)	84
`eng,lat` (English, Latin)	41
`lat,ell` (Latin, Greek)	38
`ita,deu` (Italian, German)	38
`eng` (English)	36
`deu,lat,ell` (German, Latin, Greek)	29
`fra,lat` (French, Latin)	26

Table 4.4 Top 10 most frequent etymological sequences in the training data

4.4.2 Extended Training Data

To provide more data for the model, we can extend the dataset with entries extracted from root and affix etymology dictionaries. This significantly increases label diversity:

- **Total morphs in the extended dataset:** 19 430
- **Total unique morphs in the extended dataset:** 12 211
- **Unique language sequences:** 406
- **Unique languages:** 67

To reduce noise from very rare classes and simplify the model—especially in the multilabel setting—we consider frequency cutoffs to filter out infrequent classes.

Minimum Frequency	Unique Sequences	Unique Languages
All	406	67
> 1 occurrence	216	54
> 2 occurrences	141	43
> 3 occurrences	111	35
> 4 occurrences	95	33
> 5 occurrences	81	27

Table 4.5 Number of unique language sequences and languages in the training data based on minimum frequency thresholds.

If we limit the classes to only those that appear with a frequency greater than 1 per 1,000 morphs (i.e., more than 19 occurrences), we are left with:

- **13 most frequent languages**
- **40 most frequent language sequences**

This shows that although the overall label space is large, the majority of the data is concentrated in a relatively small number of classes.

While such filtering helps reduce noise and simplify the label space, we avoid applying it too aggressively. The model itself should learn which labels are rare or unlikely based on the training data, rather than having these decisions hard-coded during preprocessing.

The high number of low-frequency language sequences may also be partly due to incorrectly parsed etymological information from the Czech Etymological Dictionary. In some cases, the dictionary lists multiple languages not as part of the actual borrowing chain, but rather to indicate similar borrowings in other languages or to illustrate certain phenomena. This can introduce noise into the dataset.

Most Frequent Language Origins

The extended training set contains 19,430 annotated morphs. While there is a wide variety of etymological origin sequences, a large portion of the data is concentrated in just a few frequent classes. The table below lists the most common language origin sequences, showing that the majority of morphs come from a relatively small set of etymological paths.

Language Sequence	Count
`ces` (Czech)	11,736
`lat` (Latin)	1,776
`ell` (Greek)	638
`deu` (German)	616
`lat,ell` (Latin, Greek)	515
`deu,lat` (German, Latin)	388
`eng` (English)	380
`fra,lat` (French, Latin)	297
`deu,fra,lat` (German, French, Latin)	196
`fra` (French)	195
`gmh` (Middle High German)	166
`deu,lat,ell` (German, Latin, Greek)	138

Table 4.6 Most common etymological origin sequences in the extended training set (∼19,000 morphs)

4.5 Classification Models

For the final classification step—predicting the etymological origin of each morph based on extracted features—we use standard machine learning models from the `scikit-learn` library. Specifically, we experiment with Logistic Regression (LR), Support Vector Machines (SVM), and Multi-Layer Perceptrons (MLP).

4.6 Self-Training

Self-training is a semi-supervised learning approach in which a model is first trained on a smaller annotated dataset and then used to predict labels for a larger unlabelled dataset. The newly labelled examples are added to the training data to further improve the model. This method is particularly useful in scenarios where obtaining additional annotated data is difficult or costly, as is often the case with manually labelled datasets.

In this work, we experimented with self-training by applying the model trained on the manually annotated data to annotate the Prague Dependency Treebank (PDT) (Hajič; Bejček, et al., 2020), part of the Universal Dependencies project (Nivre; Marneffe, et al., 2020). This corpus contains newswire and other articles from the 1990s and consists of 87,907 sentences and around one million words.

First, we normalized the text by lowercasing it and removing punctuation, numbers, and other special characters. The words were then segmented and classified using an automated tool developed by John (2024).

Finally, a few manual corrections were made to fix clear mistakes in the predictions, such as cases where the model incorrectly predicted that a single morph had both Czech and another language as its origin, which contradicts the principle that each morph is either native or borrowed, but not both.

5 Experiments and Results

In this chapter, we present the results of the experiments conducted as part of this thesis. First, we describe the outcomes on the development set, which were used to explore different model configurations and inform key decisions about parameter selection.

In the following section, we evaluate the final chosen models on the test set to measure their performance.

5.1 Experiments on the Development Set

There are many hyperparameters and options for configuring the model to predict morpheme-level etymology. In this section, we present results on the development set, which were used to guide the selection of the best-performing model for final evaluation on the test set. A wide range of hyperparameter combinations was tested using grid search. We summarize selected configurations along with their evaluation results.

For each model, we report several evaluation metrics as discussed in Chapter 3.2. All values are in %:

- **F1** — Standard F1-score averaged over all morph occurrences.
- **RER** — Relative Error Reduction compared to the dummy baseline that always predicts Czech.
- **Native / Borrowed** — F1-scores computed separately for native and borrowed morphs.
- **Unique** — F1-score grouped by morph text, where scores are averaged over unique morphs.

5.1.1 Baseline Results

We first evaluate the baseline solutions to establish lower-bound performance for the task. The results are summarized in Table 5.1.

Model	F1	RER	Native	Borrowed	Unique
Dummy Baseline	90.1	0.0	100.0	0.0	79.4
Word Dictionary	94.0	39.2	98.5	53.0	88.9
Most Frequent Origin	94.2	41.8	99.3	48.0	86.8
Morph Dictionary	94.6	45.4	98.9	55.7	89.5
OpenAI o3	94.7	46.5	99.2	53.7	88.0

Table 5.1 Performance of baseline models on development set

The dummy baseline, which always predicts Czech, achieves a standard F1-score of 90.1 %. This is because it correctly predicts all native morphs (each receiving an individual F1-score of 100 %) and fails on all borrowed ones (scoring 0 %).

The overall score reflects the proportion of Czech morphs in the dataset—90.1 % of all morphs are native. The Unique score is lower since many Czech morphs appear very frequently and therefore have a larger influence on the standard F1 average.

The other baseline models performed within a narrow range, achieving F1-scores between 94.0 % and 94.7 %. The Most Frequent Origin model successfully predicted nearly all native morphs by memorizing frequent native forms and defaulting to a native origin for unknown morphs. It also managed to correctly predict almost half of the borrowed morphs, but only those that appeared in the training data. Although additional morphs were present in both the training and development sets, some appeared with different origins depending on the context, leading to incorrect predictions, since the model only memorizes the most frequent origin for each morph.

The Morph Dictionary baseline slightly outperformed the Word Dictionary. This is likely because the word-level dictionary used in the experiment missed many entries, whereas morph-level coverage was more complete, as there are fewer unique morphs that occur frequently.

The OpenAI o3 reasoning large language model achieved F1-scores comparable to the Morph Dictionary baseline. Its average F1 score over all morph instances was the highest among all baseline methods. It correctly predicted 99.2 % of native morphs and 53.7 % of borrowed ones. These results indicate that the o3 model is capable of understanding the task and can perform it reasonably well using only the information provided in the prompt.

The prompt included not only task instructions but also the training data and hints about the distribution of origins (e.g., the high proportion of native-origin morphs). The full prompt is described in Section 3.4.5.

5.1.2 Learning Model Results

This section presents the results achieved by the trained learning models. The extracted features and model configurations are described in Chapter 4.

Features Impact on Performance

To better understand how different features affect model performance, we tested combinations of three feature types: **Vowel** (vowel start/end), **Position** (position of the morph: prefix, suffix, root, interfix), and **Type** (morph type: derivational affix, inflectional affix, root).

We always use the n-grams feature as it is the main input for the model.

Table 5.2 presents the results for models using an MLP classifier with a single hidden layer of size 30. Table 5.3 shows the results for models using an SVM classifier with an RBF kernel. Each row corresponds to a different combination of input features used in the model.

MLP Features	F1	RER	Native	Borrowed	Unique
Vowel-Position-Type	95.5	54.5	99.2	62.0	90.3
Position-Type	95.2	52.0	98.8	63.0	89.3
Vowel-Type	94.9	48.9	98.8	60.0	88.7
Vowel-Position	95.3	52.7	98.9	62.3	89.5
Type	95.0	49.9	98.8	60.9	89.4
Position	95.2	51.8	99.0	60.8	89.5
Vowel	95.0	49.2	98.7	60.9	89.6
Just n-grams	94.9	49.0	98.9	59.3	89.1

Table 5.2 MLP model performance with different feature combinations.

The results in Table 5.2 show that each added feature improves performance slightly. When all features are combined, the relative error reduction increases by 5.5 %. The vowel feature alone has minimal effect on the F1 score, and when combined with the type feature, the results are the same or slightly worse than using just character n-grams. However, adding positional information leads to a noticeable improvement. Although the combination of vowel and type features does not outperform the base setup, adding positional information to this combination (`vowel-type-position`) results in better performance than using position alone or position combined with type.

SVM Features	F1	RER	Native	Borrowed	Unique
Vowel-Position-Type	94.7	46.5	99.5	50.7	88.1
Position-Type	94.6	45.4	99.5	49.5	87.8
Vowel-Type	94.6	45.8	99.5	50.7	88.1
Vowel-Position	94.8	47.9	99.5	52.7	88.3
Type	94.1	40.3	99.3	46.5	87.2
Position	94.5	44.5	99.5	48.6	87.5
Vowel	94.3	42.9	99.5	47.7	87.8
Just n-grams	94.3	42.3	99.5	46.4	87.7

Table 5.3 SVM model performance with different feature combinations.

The results in Table 5.3 show that the best performance was achieved using the `vowel-position` combination, without the type feature. This setup improved the standard F1 score averaged over all morphs by 5.6 %. Using the type feature together with character n-grams alone actually led to worse performance. However, both the `position-type` and `vowel-type` combinations performed better than using only `vowel` or only `position`. In these configurations, the type feature improved the result slightly.

Embeddings

We experimented with different ways of incorporating embeddings into the model. Specifically, we tested setups using only morph embeddings, only word embeddings, and a combination of both. We also tested different dimensions of the embedding vectors.

These configurations were evaluated using both SVM and MLP classifiers, with and without extended training data, and using both multi-label and whole-sequence prediction.

Table 5.4 presents results for the model using an MLP classifier with a single hidden layer of 30 neurons. Table 5.5 shows results for the SVM model with an RBF kernel. Finally, Table 5.6 contains results for the MLP model with a hidden layer of 150 neurons, trained using the extended dataset with additional morphs extracted from CzEtyL and evaluated in a multi-label setup.

Embedding Config	F1	RER	Native	Borrowed	Unique
None	95.5	54.5	99.2	62.0	90.3
Word	95.0	49.5	98.7	61.2	89.2
Morph	95.1	50.8	98.9	60.4	89.4
Word + Morph	95.3	52.5	98.7	64.2	89.4

Table 5.4 Effect of morph and word embeddings (dimension 300) on MLP model (30 neurons, single-label classification).

Embedding Config	F1	RER	Native	Borrowed	Unique
None	94.7	46.5	99.5	50.7	88.1
Word	94.6	45.8	99.3	52.0	88.1
Morph	94.6	46.0	99.5	50.1	87.9
Word + Morph	94.6	45.5	99.5	50.3	87.8

Table 5.5 Effect of morph and word embeddings (dimension 300) on SVM model (single-label classification).

Embedding Config	F1	RER	Native	Borrowed	Unique
None	95.8	58.2	98.8	69.4	91.1
Word	95.2	52.1	98.6	65.0	89.8
Morph	95.7	57.2	98.7	69.0	91.1
Word + Morph	95.2	51.8	98.4	66.3	89.4

Table 5.6 Effect of morph and word embeddings (dimension 300) on MLP model (150 neurons, multi-label classification, with extended training data, filtering on minimal 3 occurrences of languages sequence).

A wide range of configurations was tested using grid search, including combinations of different feature sets, classifier types, embedding types and dimensions, and training data variants. The tables 5.4, 5.5, and 5.6 illustrate a selected subset of these configurations.

Ultimately, the use of embeddings did not result in any consistent improvement. In nearly all tested setups, the performance was comparable to or slightly worse than models trained without embeddings.

Learning models base results

Table 5.7 shows the results on the development set using different final classifiers. In all cases, single-label prediction was used, treating the entire etymology sequence as a single class.

Model	F1	RER	Native	Borrowed	Unique
Logistic Regression	94.0	39.7	98.7	51.4	86.2
SVM	95.2	51.5	99.2	59.1	89.2
MLP-30	95.5	54.5	99.2	62.0	90.3
MLP-100	95.5	54.8	99.2	62.4	90.2
MLP-300	95.5	55.1	99.2	62.7	90.3

Table 5.7 Evaluation results of the learning models with default settings on the development set.

Logistic regression with different settings was tested, but it consistently underperformed compared to both SVM and MLP models.

The MLP refers to a model using a multilayer perceptron with a single hidden layer of a specified size. Various regularization settings were explored during development. For the MLP models in this section, the `alpha` parameter for L2 regularization was set to 1×10^{-4}. For SVM models, regularization strength was controlled by the `C` parameter, set to 3 (which is inversely proportional to the regularization strength).

Increasing the number of neurons in the hidden layer did not result in significant improvements across the tested configurations. One notable exception is the setup with an extended training set that included root and affix entries extracted from CzEtyL. In this case, the number of training morphs nearly tripled, and the additional capacity of the larger model proved beneficial.

MLP Size	F1	RER	Native	Borrowed	Unique
30	95.2	52.0	98.6	65.1	89.0
100	95.4	54.1	98.3	69.9	90.2
300	95.8	58.1	98.5	71.9	90.8
500	95.8	57.8	98.5	71.6	91.1

Table 5.8 MLP performance with extended training data and different hidden layer sizes.

Extended and Multi-Label

Table 5.9 shows the results for three models using an MLP classifier with different hidden layer sizes in a multi-label setup, where each language is predicted independently rather than predicting the entire sequence as a single class.

MLP Size	F1	RER	Native	Borrowed	Unique
30	95.2	52.2	98.8	63.4	89.5
100	95.1	50.7	98.7	62.7	89.0
300	95.4	53.3	99.1	61.6	90.0

Table 5.9 Multi-label MLP performance with different hidden layer sizes (just regular training data).

In Table 5.10, the results are shown for the multi-label configuration with an extended training set, where roots and affixes extracted from CzEtyL were added to the training data.

MLP Size	F1	RER	Native	Borrowed	Unique
30	96.1	60.4	99.0	69.8	91.5
100	96.0	59.4	98.7	71.3	91.1
300	95.8	57.5	98.9	67.6	90.9

Table 5.10 Multi-label MLP performance with extended training data.

Using a multi-label target strategy—where one binary classifier is trained for each individual language—does not improve performance when using only the original training data. However, when the training set is extended with additional morphs from the etymological dictionary, the results improve.

It also appears that increasing the number of neurons in the hidden layer does not lead to further improvements under this strategy. This may suggest that the model already has sufficient capacity. With this multi-label configuration, a separate classifier is trained for each language, which effectively increases the overall capacity of the model.

The results for the models using SVM are shown in Table 5.11.

SVM Variant	F1	RER	Native	Borrowed	Unique
multi-label + extended	86.0	-41.4	89.8	51.3	76.0
extended	95.0	49.6	99.1	57.9	89.3
multi-label	94.9	48.7	98.9	58.4	88.3
base	95.2	51.5	99.2	59.1	89.2

Table 5.11 Performance of SVM model variants under different configurations.

In the configuration that combined both multi-label classification and the extended training set, the SVM model performed significantly worse than in other settings, even falling below the dummy baseline. This suggests that training a separate SVM classifier for each language led to overfitting on the extended data. Notably, this setup performed much worse on native morphs, achieving an F1 score below 90 %, compared to 99.2 % in the base SVM configuration. All other SVM variants—including those using either multi-label classification or extended training alone—achieved comparable but slightly worse results than the base model.

Best Achieved Result on Development Set

The best-performing model on the development set, selected from hundreds of tested parameter combinations, was a multi-label classifier using a multilayer perceptron (MLP) with one hidden layer of 30 neurons. This model achieved the results shown in Table 5.12.

Model	F1	RER	Native	Borrowed	Unique
Best Configuration	96.2	61.2	98.8	72.5	91.6

Table 5.12 Best result on the development set across all model configurations

The model was trained in a multi-label setup, where a separate binary classifier was used for each language. It used the ReLU activation function and applied L2 regularization with a strength (`alpha`) of 1×10^{-4}. The optimizer was Adam with default parameters from the `scikit-learn` implementation: a learning rate of 1×10^{-3} and a batch size of 200. The training process ran for 200 epochs. Additionally, the training data was extended using root and affix dictionaries extracted from CzEtyL, and morphs that appeared fewer than two times in the training set were filtered out.

Self-Training Results

The best-performing model was used to predict labels on the Prague Dependency Treebank (PDT) (Hajič; Bejček, et al., 2020), a corpus consisting mainly of news articles from the 1990s, consisting of almost one million words. A new model with the same configuration was then trained on this self-labelled data and evaluated on the development set. The results are presented in Table 5.13.

Model	F1	RER	Native	Borrowed	Unique
Self-training	96.2	61.1	98.7	73.8	91.7

Table 5.13 Results of self-training on PDT corpus evaluated on the development set

The results are very similar to the model that labeled the training data for this experiment. The model trained on the automatically labeled large corpus achieved a slightly better score on borrowed morphs than the original model. However, overall, self-training in this setup did not yield the improvements we had hoped for.

5.2 Final Results on Test Set

Having explored a wide range of models and parameter settings on the development set, we now move on to evaluating the best configurations on the test set. In this section, we assess the performance of the final selected models, alongside the baseline methods, to provide a direct comparison and confirm how well the findings from the development experiments generalize to unseen data.

5.2.1 Baseline Results

We first evaluate the baseline solutions on the test set. The results are summarized in Table 5.14.

Model	F1	RER	Native	Borrowed	Unique
Dummy Baseline	90.0	0.0	100.0	0.0	77.4
Word Dictionary	94.8	48.1	98.6	60.9	90.2
Most Frequent Origin	94.4	43.4	99.2	50.9	86.1
Morph Dictionary	94.6	46.0	98.6	58.8	88.9
OpenAI o3	94.4	43.5	99.3	50.9	86.2

Table 5.14 Performance of baseline models on test set

The Word Dictionary model performed best on the test set, whereas on the development set, the Morph Dictionary model achieved better results. The difference may be due to a higher match rate between words in the test set and entries in the Word Dictionary.

5.2.2 Learning Models Results

The performance of the learning models on the test set is shown in Table 5.15.

Model	F1	RER	Native	Borrowed	Unique
Logistic Regression	94.7	47.3	99.3	53.4	86.7
SVM	95.0	49.5	99.7	52.2	87.1
MLP-30	95.3	53.0	98.9	63.2	88.0
MLP-100	95.1	50.4	98.4	64.6	87.8
MLP-300	95.1	51.1	98.6	64.0	88.0

Table 5.15 Evaluation results of the learning models with default settings on the test set.

The results of both the baseline models and the learning models on the test set are consistent with their performance on the development set. The differences are small, with most models showing a deviation of less than 0.5 % in the standard F1 score. An exception is the logistic regression model, which achieved a 0.7 % higher F1 score on the test set compared to the development set. Additionally, while the best-performing model on the development set was MLP-300, on the test set it was MLP-30. However, in both cases, the differences between MLP models with different hidden layer sizes were minimal.

Training on Train + Development

Table 5.16 shows the results of various classifiers trained on the training set, which includes both the training and development data. All models were evaluated using single-label prediction.

Model	F1	RER	Native	Borrowed	Unique
Most Frequent Origin	94.7	46.6	99.2	54.0	87.0
Logistic Regression	95.0	49.5	99.3	56.3	87.4
SVM	95.0	50.0	99.3	56.1	87.9
MLP-30	96.0	59.7	98.6	72.5	90.2
MLP-100	95.5	55.3	98.4	69.5	89.2
MLP-300	95.9	58.4	98.6	71.2	89.7

Table 5.16 Evaluation of various classifiers trained on the extended training set (train + dev)

When comparing the results of the learning models trained only on the training set (Table 5.15) with those trained on the combined training and development sets (Table 5.16), we observe a significant improvement. In particular, the MLP models show an increase of 5–7 % in relative error reduction. For the MLP-30 model specifically, the F1 score on borrowed morphs improves by nearly 10 %.

This shows that increasing the size of the training dataset is beneficial, as the model encounters more morphs with correct labels to learn from. The improvement is not simply due to the model memorizing new morphs from the development set that also appear in the test set, as the performance of the most frequent origin baseline does not improve as much when trained on the combined training and development sets compared to training on the original training set alone.

The Logistic Regression model and the SVM model did not improve as much. The SVM model in particular showed almost no overall improvement. Although there was an increase of nearly 4 % in the F1 score for borrowed morphs, the score for native morphs slightly decreased.

5.2.3 Best Model Results

The best-performing configuration from the development experiments—multi-label setup, extended training set, and an MLP with 30 neurons—was trained both on the original training set and on the combined training and development sets. The evaluation results on the test set are presented in Table 5.17.

Model	F1	RER	Native	Borrowed	Unique
Just train	96.1	61.0	98.4	75.4	90.7
Train + dev	96.8	67.9	98.9	77.8	91.8

Table 5.17 Results of the best model on test set

6 Implementation Details

This chapter describes the structure of the attached repository with the code and data used in this thesis and provides additional implementation details.

All code and data used in this thesis are also publicly available on GitHub: `https://github.com/ampapacek/MorphemeOrigin`

The repository includes all scripts, models, and annotated data necessary to reproduce the experiments presented in this work. It is implemented in `Python`, and uses the `scikit-learn` library for training machine learning models.

A brief overview and usage instructions can be found in the `README.md` file. The following sections describe the structure of the repository and the implementation of the code in more detail.

6.1 Repository Structure

```
.
├── Makefile
├── README.md
├── requirements.txt
├── data
│   ├── etymological_data
│   │   ├── affixes_etymology.tsv
│   │   ├── czetyl_max.tsv
│   │   └── roots_etymology.tsv
│   └── annotations
│       ├── ces.sentence.dev.tsv
│       ├── ces.sentence.train.tsv
│       ├── dev.tsv
│       ├── dev_annotator2.tsv
│       ├── test.tsv
│       └── train.tsv
└── src
    ├── baselines.py
    ├── data_sentece.py
    ├── data_transformers.py
    ├── inter_annotator.py
    ├── main.py
    ├── model.py
    ├── morph_classifier.py
    ├── prepare_for_annotation.py
    └── utils.py
```

The `Makefile` handles the creation of a virtual environment and installs dependencies listed in the `requirements.txt` file. It provides a simple way to

run all baseline models and the learning model with default settings on the development set.

Additionally, the Makefile includes commands to measure inter-annotator agreement and to clean up output files generated by the scripts.

6.1.1 Data

The directory `data/` contains two subdirectories: `etymological_data/` and `annotations/`. The first contains extracted files from CzEtyL (Rejzek et al., 2025), which are used mainly by the baseline models and optionally as an extension of the training data for the learning model.

The `annotations/` directory contains the original data from the SIGMORPHON 2022 shared task, which were used to create the training, development, and test sets by adding morph-level etymological annotations. These annotated files are stored here, along with the file `dev_annotator2.tsv`, which includes development set annotations provided by the second annotator.

6.1.2 Source Code

For custom configurations and advanced options, the main entry point is the script `src/main.py`. The remaining source files in the `src/` directory are:

- `baselines.py` – Implements baseline models such as the dummy predictor, dictionary-based models, and most frequent origin predictor.
- `data_sentece.py` – Defines the `DataSentence` class, which stores a list of words, each containing morphs annotated with features such as type, etymology, and position.
- `data_transformers.py` – Contains transformers that generate additional features from the sentence data. Includes the embedding transformer and vowel-start/end transformer.
- `inter_annotator.py` – Computes inter-annotator agreement statistics, including exact match rate and Cohen's kappa. Also writes differences between annotations to a file.
- `model.py` – Defines the abstract model interface shared by all model types.
- `morph_classifier.py` – Implements the main learning model pipeline using the `scikit-learn` library.
- `prepare_for_annotation.py` – Generates annotation files in the expected format, such as `data/annotations/ces.sentence.train.tsv`.
- `utils.py` – Contains utility functions used across the project, including file I/O and evaluation routines.

6.2 Main

The main entry point of the repository is the script `src/main.py`. It integrates all modules and components needed to run experiments with various settings, which can be specified using command-line arguments processed by an argument parser.

Through these arguments, it is possible to configure which model(s) to run, what training and testing data to use, and various settings for the learning model (`MorphClassifier`). Features such as morph type, position, vowel patterns, character n-grams, and embeddings can be individually controlled.

The script can also output useful statistics, such as the frequency of specific language sequences in the data or the frequency of individual morphs. Furthermore, it can log model prediction errors to a file, allowing inspection of incorrectly classified morphs.

Models can be saved after training and later loaded to avoid retraining from scratch. Model saving and loading is handled using `pickle`. The results are printed to both the console and a `.tsv` file by default.

For a more detailed description of available arguments, see the `README.md` file or use the `-h` or `--help` options when running the main script.

6.3 Morph Classifier

The `MorphClassifier` class implements the `Model` interface by extending the abstract `Model` class. It provides the `fit` and `predict` methods. The `fit` method trains the model on a list of `DataSentence` objects, adjusting its internal parameters accordingly. The `predict` method returns a new list of sentences, each with predicted etymology labels filled in for every morph. Additionally, the `MorphClassifier` includes `save` and `load` methods, allowing the trained model to be saved and later reused.

6.3.1 DataSentence Class

The `DataSentence` class stores a list of `Word` objects. It provides an iterator to iterate directly over all morphs in the sentence and a property that returns the total number of morphs. The class also includes a constructor for creating a deep copy from an existing `DataSentence`, as well as customized `repr` and `str` methods for debugging and printing.

Word Class

The `Word` class is a lightweight wrapper around a list of `Morph` objects. It provides properties to retrieve the text representation of the word, the number of morphs it contains, and includes `repr` and `str` methods for debugging and display.

Morph Class

The `Morph` class represents an individual morph. Each instance stores the morph's text, its etymology (a list of language codes), its morph type (root,

derivational affix, or inflectional affix), and its morph position (prefix, root, interfix, or suffix). Both type and position are defined using enumeration types. The class also implements `str` and `repr` methods.

6.3.2 Fitting

First, the list of `DataSentence` objects is converted into a single `pandas DataFrame`. Morphs that occur less frequently than a specified minimum frequency parameter are filtered out.

Next, the preprocessing pipeline is constructed. It includes a set of transformers depending on the configuration. These may consist of one-hot encoders for morph type and morph position, a character n-gram count vectorizer, custom embedding transformers for morphs and words, and a transformer for generating vowel structure features.

After preprocessing is set up, the final classifier is initialized. This can be an MLP (either standalone or as part of an ensemble), an SVM, or a logistic regression model, each configured with specific hyperparameters.

The target values are then prepared: they are converted either into a one-hot encoded form (for the single-label setup) or processed using a multi-label binarizer (for the multi-label setup). Finally, the classifier is trained on the resulting transformed feature matrix.

6.3.3 Prediction

When predicting the etymology of morphs in test sentences, we first construct a `pandas DataFrame` from all input `DataSentence` objects. The pipeline's preprocessor transforms this DataFrame into a feature representation, which is then passed to the trained classifier to generate predictions.

Depending on whether the model is configured for single-label or multi-label classification, the predicted outputs are inverse-transformed into lists of language labels. These predicted etymologies are subsequently assigned to the corresponding morphs in the original sentence objects. In the multi-label setup, it is possible for a morph to receive an empty prediction. In such cases, we default to assigning the label `ces`, indicating native origin.

6.3.4 Save and Load

The `MorphClassifier` class includes functionality for saving and loading trained models using the `pickle` module. This allows the trained model to be saved to disk and reloaded later without the need for retraining. The `save` method serializes the full model pipeline, including the classifier, feature transformers, label encoder, and relevant configuration parameters—such as the model name and flags indicating the use of multi-label classification or specific feature types.

6.4 Reproducibility

Model performance can be influenced by random initialization, particularly in the case of MLP classifiers. To ensure reproducibility, we fixed the random seed

for all experiments. The seed value was set to `34867991`, (the author's university identification number). With this seed, all results presented in this thesis can be reproduced.

In addition, we tested several other arbitrarily chosen seed values and observed only minor variability in the results. These variations did not affect the overall trends or conclusions.

Conclusion

In this thesis, we addressed the problem of automatic identification of morpheme origins. For each morpheme in morphologically segmented text, we aimed to determine whether it is native or borrowed, and if borrowed, from which language it originated and through which languages it entered Czech.

Although linguists possess this knowledge, most available resources focus on the word level, and it is difficult to extract information for individual morphs. There is currently no dataset that provides etymological annotation at the morph level for segmented words.

To tackle this, 300 Czech sentences from SIGMORPHON (2022), already segmented into individual morphemes, were manually annotated with etymological information at the morpheme level. The resulting dataset was then divided into three parts: training, development, and test sets.

Another annotator also manually annotated part of the dataset to measure inter-annotator agreement. This serves to evaluate how clearly the task is defined, how consistently humans agree on the annotation, and to help improve the quality of the dataset by revealing possible mistakes. The inter-annotator agreement can also serve as an estimate of the upper bound for model performance.

Several baseline solutions were developed, such as always predicting a native origin, remembering the most frequent origin for a given morph from the training data, using a dictionary of origins for roots and affixes extracted from CzEtyL (Rejzek et al., 2025), or applying a morphological analyser to retrieve lemmas and match them with an etymological dictionary. Predictions generated by OpenAI's latest reasoning model, o3, were also used as an additional baseline.

For the main learning model, we extracted various features, with the most significant being character n-grams, positional type of the morph (prefix/interfix/-suffix/root), lexical type of the morph (root/derivational affix/inflectional affix), and whether the morph starts or ends with a vowel.

Based on these features, a classifier was trained to predict the correct origin of each morph. We tested a single-label setup, where the full sequence of origins for a single morph was treated as one class, and a multi-label setup, where each language was predicted separately. The multi-label setup achieved better results, especially when combined with an extended training set using roots and affixes extracted from CzEtyL. For final classification, we experimented with MLP, SVM, and logistic regression models, with the MLP achieving the best performance. The best results were obtained using an MLP with a single hidden layer containing 30 neurons. Increasing the number of neurons or adding additional hidden layers did not lead to any significant improvement.

We also experimented with embeddings, but they did not improve the model's performance. Additionally, we tried self-training as a form of semi-supervised learning: the model trained on manually annotated data was used to predict labels for a large corpus (PDT (Hajič; Bejček, et al., 2020), over one million running text tokens), and a new model was trained on this automatically labelled data. However, this approach did not lead to any noticeable improvement in performance.

Main Achievements

I created a dataset consisting of 300 sentences, 3,982 words, and 9,961 morphs, where I manually annotated the origin of each morph, including the sequence of languages through which the morph entered the Czech language. This dataset can be used independently by other researchers exploring different approaches to this task.[1]

I defined an evaluation metric for this task and tested a range of baseline solutions, as well as several learning models. Various parameter configurations were explored to assess their impact on model performance.

I trained a model that outperformed all baseline solutions and reduced the error of the simple baseline (always predicting Czech) by nearly 70 %. On the test set, the best configuration achieved an F1-score of 98.9 % on native morphs and 77.8 % on borrowed morphs, with an overall F1-score of 96.8 % averaged across all morphs.

Future Work

One direction for future work is to expand the manually annotated dataset, as a larger training set would likely lead to better-performing models. Additionally, the annotation could be made more fine-grained. For example, by distinguishing between different historical stages of Latin, German, or Greek, and specifying whether native morphs are of broader Slavic origin or specific to Czech.

Another promising direction is to explore the use of pretrained large language models and fine-tune them specifically for the task of morph-level etymology prediction.

It would also be beneficial to further explore alternative strategies for self-training, with the goal of minimizing the need for supervision and manual annotation. More complex model architectures might yield better results and could be more effective at learning from self-labelled data. Additionally, models that naturally handle sequence prediction may outperform those based on simple classification.

Instead of relying on a single classifier and configuration, future work could also consider ensemble approaches. Combining different types of classifiers may capture complementary strengths, improving overall performance.

Finally, although this work focused on the Czech language, the proposed approach is not language-specific and could be extended and tested on other languages as well.

[1]The dataset is publicly available at `https://github.com/ampapacek/MorphemeOrigin/tree/main/data/annotations`

Bibliography

Aronoff, M.; Fudeman, K., 2011. *What is Morphology?* Wiley. Fundamentals of Linguistics. isbn 9781444351767. Available also from: https://books.google.cz/books?id=bolGMMyZVjMC.

Batsuren, K.; Bella, G.; Giunchiglia, F., 2021. MorphyNet: A Large Multilingual Database of Derivational and Inflectional Morphology. In: *Proceedings of the 18th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pp. 39–48.

Bojanowski, P.; Grave, E.; Joulin, A.; Mikolov, T., 2017. Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics*. Vol. 5, pp. 135–146. issn 2307-387X. Available from doi: 10.1162/tacl_a_00051.

Booij, G., 2007. *The Grammar of Words: An Introduction to Linguistic Morphology*. Oxford University Press. isbn 9780199226245. Available from doi: 10.1093/acprof:oso/9780199226245.001.0001.

Cohen, J., 1960. A coefficient of agreement for nominal scales. *Educational and psychological measurement*. Vol. 20, no. 1, pp. 37–46.

Eberhard, D. M.; Simons, G. F.; Fennig, C. D. (eds.), 2025a. *Ethnologue: Languages of the World*. 28th ed. Dallas, Texas: SIL International. Available also from: http://www.ethnologue.com.

Eberhard, D. M.; Simons, G. F.; Fennig, C. D., 2025b. *Ethnologue: Languages of the World*. 28th. SIL International. Available also from: https://www.ethnologue.com/insights/largest-families/. Online version, accessed on 21.2. 2025.

Gamkrelidze, T. V.; Ivanov, V. V., 1995. Chapter Four: The Structure of the Indo-European Root. In: *Indo-European and the Indo-Europeans: A Reconstruction and Historical Analysis of a Proto-Language and Proto-Culture*. Berlin, Boston: De Gruyter Mouton, pp. 185–187. Available from doi: 10.1515/9783110815030-111.

Goethe, J. von, 1817. *Zur Morphologie*. J.G. Cotta. Goethes Werke, no. sv. 1. Available also from: https://books.google.cz/books?id=jk5StAEACAAJ.

Grave, E.; Bojanowski, P.; Gupta, P.; Joulin, A.; Mikolov, T., 2018. Learning Word Vectors for 157 Languages. In: *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*.

Hajič, J.; Bejček, E.; Hlaváčová, J.; Mikulová, M.; Straka, M.; Štěpánek, J.; Štěpánková, B., 2020. Prague Dependency Treebank – Consolidated 1.0. In: *Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020)*. Marseille, France, pp. 5208–5218. Available also from: https://aclanthology.org/2020.lrec-1.641.pdf.

Hajič, J.; Hlaváčová, J.; Mikulová, M.; Straka, M.; Štěpánková, B., 2024. *MorfFlex CZ 2.1*. Available also from: https://hdl.handle.net/11234/1-5833. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Haspelmath, M.; Sims, A., 2013. *Understanding Morphology*. Taylor & Francis. Understanding Language. ISBN 9781444117110. Available also from: https://books.google.cz/books?id=gWnYAQAAQBAJ.

Haspelmath, M., 2020. *The morph as a minimal linguistic form*. De Gruyter Mouton.

Holub, J.; Kopečný, F., 1952. *Etymologický slovník jazyka českého*. 3. přepracované vydání. Praha: Státní nakladatelství učebnic.

John, V., 2024. *Morph Classifier*. MA thesis. Charles University.

Kosek, P., 2017. Periodizace vývoje češtiny. In: Karlík, P.; Nekula, M.; Pleskalová, J. (eds.). *CzechEncy - Nový encyklopedický slovník češtiny*. Masarykova univerzita. Available also from: https://www.czechency.org/slovnik/PERIODIZACE%20V%C3%9DVOJE%20%C4%8CE%C5%AOTINY. Last accessed: 23. 2. 2025.

Kümmel, M.; Zehnder, T.; Lipp, R.; Schirmer, B., 2001. *LIV: Lexikon der indogermanischen Verben*. Wiesbaden: Dr. Ludwig Reichert Verlag.

Machek, V., 1968. *Etymologický slovník jazyka českého*. Praha: Academia.

McCarthy, A. D. et al., 2020. UniMorph 3.0: Universal Morphology. In: *Proceedings of the 12th Language Resources and Evaluation Conference (LREC 2020)*. European Language Resources Association (ELRA).

Mikolov, T.; Chen, K.; Corrado, G.; Dean, J., 2013. Efficient Estimation of Word Representations in Vector Space. In: *Proceedings of the International Conference on Learning Representations (ICLR)*. Available also from: https://arxiv.org/abs/1301.3781.

Nivre, J.; Agić, Ž., et al., 2017. *Universal Dependencies 2.1*. Available also from: https://universaldependencies.org/. Accessed: 2025-04-10.

Nivre, J.; Marneffe, M.-C. de; Ginter, F.; Hajič, J.; Manning, C.; Pyysalo, S.; Schuster, S.; Tyers, F.; Zeman, D., 2020. Universal Dependencies v2: An Evergrowing Multilingual Treebank Collection. In: *Proceedings of the 12th International Conference on Language Resources and Evaluation (LREC 2020)*. Marseille, France, pp. 4034–4043.

Olander, T., 2022. *The Indo-European Language Family*. Cambridge University Press.

Olbrich, M.; Brezinová, V.; Dohnalová, Š.; John, V.; Kyjánek, L.; Papáček, A.; Svoboda, E.; Ševčíková, M.; Vidra, J.; Žabokrtský, Z., 2025. *DeriNet 2.3*. Available also from: http://hdl.handle.net/11234/1-5846. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

OpenAI, 2025. *Introducing OpenAI o3 and o4-mini*. Available also from: https://openai.com/index/introducing-o3-and-o4-mini/. Accessed: 2025-05-01.

Rejzek, J., 2019. *Český etymologický slovník*. LEDA. No. 3.th edition. ISBN 978-80-7335-393-3. Available also from: https://leda.cz/Titul-detailni-info.php?i=623.

Rejzek, J.; Papáček, A.; Brezinová, V.; Žabokrtský, Z., 2025. *Czech Etymological Lexicon 1.0*. Available also from: http://hdl.handle.net/11234/1-5845. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Schachter, P.; Otanes, F., 1983. *Tagalog Reference Grammar*. University of California Press. California library reprint series. ISBN 9780520049437. Available also from: https://books.google.cz/books?id=E8tApLUNy94C.

SIGMORPHON, 2022. *SIGMORPHON 2022 Shared Task on Morpheme Segmentation* [https://github.com/sigmorphon/2022SegmentationST]. Accessed: 2025-04-10.

Straková, J.; Straka, M.; Hajič, J., 2014. Open-Source Tools for Morphology, Lemmatization, POS Tagging and Named Entity Recognition. In: *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. Baltimore, Maryland: Association for Computational Linguistics, pp. 13–18. Available also from: http://www.aclweb.org/anthology/P/P14/P14-5003.pdf.

Thomason, S., 2001. *Language Contact: An Introduction*. Edinburgh University Press. ISBN 9781474473125. Available also from: https://books.google.cz/books?id=XLZJzgEACAAJ.

Wiktionary, 2025. *Wiktionary: The free dictionary* [https://www.wiktionary.org/]. Accessed: 2025-04-10.

List of Figures

2.1 Example of the Deconstructor web interface output
Source: `https://deconstructor.ayush.digital` 22
2.2 Example of DeriNet tree for the word *učit* ("to teach"). Source:
`https://quest.ms.mff.cuni.cz/derisearch2/v2/databases/Czech-DeriNet-2.1/dcql` . . 23
2.3 Example of partially expanded DeriNet tree for the word *učit*
("to teach"). Source: `https://quest.ms.mff.cuni.cz/derisearch2/v2/databases/Czech-DeriNet-2.1/dcql` 24

List of Tables

3.1 Size of the annotated dataset used for training, development, and testing . . . 28

4.1 Most common root morphs . . . 41
4.2 Most common affixes . . . 41
4.3 Most common multi-character affixes . . . 42
4.4 Top 10 most frequent etymological sequences in the training data 43
4.5 Number of unique language sequences and languages in the training data based on minimum frequency thresholds. . . . 43
4.6 Most common etymological origin sequences in the extended training set (∼19,000 morphs) . . . 44

5.1 Performance of baseline models on development set . . . 46
5.2 MLP model performance with different feature combinations. . . . 48
5.3 SVM model performance with different feature combinations. . . . 48
5.4 Effect of morph and word embeddings (dimension 300) on MLP model (30 neurons, single-label classification). . . . 49
5.5 Effect of morph and word embeddings (dimension 300) on SVM model (single-label classification). . . . 49
5.6 Effect of morph and word embeddings (dimension 300) on MLP model (150 neurons, multi-label classification, with extended training data, filtering on minimal 3 occurrences of languages sequence). 49
5.7 Evaluation results of the learning models with default settings on the development set. . . . 50
5.8 MLP performance with extended training data and different hidden layer sizes. . . . 50
5.9 Multi-label MLP performance with different hidden layer sizes (just regular training data). . . . 51
5.10 Multi-label MLP performance with extended training data. . . . 51
5.11 Performance of SVM model variants under different configurations. 51
5.12 Best result on the development set across all model configurations 52
5.13 Results of self-training on PDT corpus evaluated on the development set . . . 52
5.14 Performance of baseline models on test set . . . 53
5.15 Evaluation results of the learning models with default settings on the test set. . . . 53
5.16 Evaluation of various classifiers trained on the extended training set (train + dev) . . . 54
5.17 Results of the best model on test set . . . 54