

Zsanett Ferenczi

**AUTOMATIC DICTIONARY BUILDING
METHODS FOR FINNO-UGRIC LANGUAGES**

PhD Dissertation

Pázmány Péter Catholic University

Doctoral School of Linguistics

Chair: **Prof. Balázs Surányi DSc**

Language Technology Workshop

Coordinator: **Prof. Gábor Prószéky DSc**

Supervisor

Eszter Simon PhD

Budapest

2023

Ferenczi Zsanett

**AUTOMATIKUS SZÓTÁRÉPÍTÉSI MÓDSZEREK
FINNUGOR NYELVEKRE**

Doktori (PhD) értekezés

**Pázmány Péter Katolikus Egyetem
Bölcsészet- és Társadalomtudományi Kar
Nyelvtudományi Doktori Iskola**

Vezetője: **Dr. Surányi Balázs**

egyetemi tanár, az MTA doktora

Nyelvtechnológia Műhely

Vezetője: **Dr. Prószéky Gábor**

egyetemi tanár, az MTA doktora

Témavezető

Dr. Simon Eszter

Budapest

2023

Contents

Acknowledgements	1
1 Introduction	2
1.1 Motivation	2
1.2 Vocabulary Acquisition and Lexicography	3
1.2.1 Electronic Lexicography	7
1.2.2 Advantages of Electronic Dictionaries	10
1.2.3 Finnish and Hungarian Bilingual Dictionaries	11
1.3 Grammar	15
1.4 Computer-Assisted Language Learning	15
1.5 Research Questions	20
1.6 Roadmap	20
2 Bilingual Dictionary Building Methods	23
2.1 Existing Automatic Methods	24
2.1.1 Parallel Corpora	24
2.1.2 Comparable and Monolingual Corpora	26
2.1.3 Alternative Solutions	29
2.2 Automatic Bilingual Dictionary Building for Finno-Ugric Languages	32
2.3 Resources	33
2.3.1 Wordnets	34
2.3.2 Wiktionary	37
2.3.3 OPUS	40
2.4 Language Processing Tools	41
2.5 Methodology	43
2.5.1 WordNet Connector	44
2.5.2 Wiktionary Parser	46
2.5.3 OPUS Extractor	48

2.5.4	wikt2dict	50
2.6	Evaluation	52
2.6.1	Details of Data Extraction	52
2.6.2	Manual Evaluation in the Dictionary Writing System	55
2.7	Discussion	62
3	Lexicographic Database	64
3.1	Definitions	64
3.2	Data Representation	65
3.3	Data Model	66
3.4	Database Schema	67
3.4.1	Entity	69
3.4.2	Relation	73
3.4.3	Source	75
3.4.4	SourceType	76
3.4.5	Remark	76
3.4.6	Users	77
3.4.7	PartOfSpeech	78
3.4.8	Label	78
3.4.9	Languages	80
3.4.10	Inflection	81
3.5	Populating the Database	83
3.6	Views	86
3.6.1	ViewRelation	86
3.6.2	ViewSourceEntity	87
3.6.3	ViewSourceRelation	88
3.7	Triggers	89
3.8	Querying	89
3.9	Conclusion	93

4	Computer-Assisted Language Learning	95
4.1	What is CALL?	95
4.2	Evolution of CALL	96
4.2.1	CALL in Education	98
4.2.2	NLP-enhanced CALL	99
4.3	Limitations of CALL	100
4.4	Conclusion	102
5	Finno-Ugric Lexical Resources	104
5.1	Introduction	104
5.1.1	User Profile	105
5.2	Dictionary Writing System	106
5.2.1	Off-the-shelf Dictionary Writing Systems	107
5.2.2	Manual Validation and Dictionary Editing	108
5.2.3	Results	116
5.2.4	Future Work	119
5.3	Dictionary	120
5.3.1	Macrostructure	121
5.3.2	Microstructure	122
5.3.3	Automatic Entry Creation and Formatting	126
5.3.4	Future Work	126
5.4	Computer-Assisted Language Learning Application	127
5.4.1	Motivation	128
5.4.2	Extracted Data	130
5.4.3	Accessing the Application	131
5.4.4	Virtual Flashcard Module	131
5.4.5	Cloze Tasks Module	133
5.4.6	Evaluation and Error Analysis	142
5.4.7	Conclusion and Future Work	148
5.5	Conclusion	151

6 Conclusions	153
6.1 Summarized Results	153
6.2 Directions for Further Research	156
References	158
Appendices	174
Appendix A: Entity Relationship Diagram of the Database	174
Appendix B: Resources and tools created during this research	175
Appendix C: Contents of the Inflection Table	177
Appendix D: List of User Roles and Permissions	187
Magyar nyelvű összefoglaló	188
Summary in English	189

List of Tables

1.1	Language learning material availability for Finnish and Hungarian on different learning platforms.	17
2.1	Number of synsets, word senses and words in the Hungarian WordNet grouped by parts of speech.	35
2.2	Number of synsets, word senses and words belonging to each part of speech in the Finnish WordNet.	36
2.3	The list of fields that constitute a word line in the CoNLL-U format.	42
2.4	Details of the resources that facilitated the generation of translation pairs.	44
2.5	Structure of the output file when the <code>synsets</code> option is selected.	46
2.6	Structure of the output file when the <code>definitions</code> option is selected.	46
2.7	Excerpt from the output of OPUS translation candidates before lemmatization.	49
2.8	Results of data collection.	53
2.9	Intermediate data validation results.	56
2.10	Precision and expected number of correct translations and synonyms for each method.	58
2.11	Precision and expected number of correct definitions and example sentences for each method.	60
3.1	Some of the most common data types used in the database.	68
3.2	Structure of the Entity table.	69
3.3	Types of entities with examples.	70
3.4	Number and percentage of lemmata where frequency data is known.	71
3.5	Types of entities and their meaning.	72
3.6	Information about the polysemous Finnish lemma <i>laki</i>	73
3.7	The structure of the Relation table.	74
3.8	The contents of the RelationType table.	75
3.9	The structure of the Source table.	76
3.10	The contents of the SourceType table.	77
3.11	The structure of the Remark table.	77
3.12	The structure of the Users table.	78

3.13	The contents of the PartOfSpeech table.	79
3.14	The contents of the Label table.	80
3.15	The contents of the Languages table.	81
3.16	The structure of the Inflection table.	82
3.17	A subset of the contents of the Inflection table.	82
3.18	Statistics about inflection type and consonant gradation for Finnish lemmata.	84
3.19	Results of SQL query 3.	87
3.20	Result of SQL query 7.	92
3.21	Results of SQL query 8.	92
3.22	Results of SQL query 8 with an additional condition applied to the part of speech of the resulting words.	94
5.1	List of fields that appear when a certain type of entity is edited.	112
5.2	Relations can be assigned to exactly one sense of <i>szél</i>	115
5.3	Detailed results of entity validation.	116
5.4	Evaluation of methods based on the validated relations.	118
5.5	The structure of the TokenAnalysis table.	135
5.6	The structure of the Analysis2Sentence table.	135
5.7	Parameters in the JSON file for Finnish exercises.	138
5.8	Parameters in the JSON file for Hungarian exercises.	142
5.9	Number of validated data that can be utilized as flashcards.	143
5.10	Number of sentences obtained for each exercise type.	143
5.11	Details of manual validation regarding analysis of sentences.	144
5.12	Examples for sentences where the language processing tools made mistakes.	144
5.13	Number of expected sentences where the results of language processing are sup- posedly correct.	145
5.14	Details of the manual validation of automatically generated tasks.	147

List of Figures

1.1	List of translations without sense indicators in DictZone.	14
1.2	Outline of the dissertation.	22
2.1	The structure of the Finnish WordNet.	36
2.2	The structure of a synset in the Hungarian WordNet.	37
2.3	Definitions in Wiktionary when the language of the headword and that of the Wik- tionary edition are the same.	38
2.4	Translation tables from the English Wiktionary entry <i>gel</i>	39
2.5	Finnish entry in the English Wiktionary edition.	40
2.6	Excerpt from the OPUS word alignments.	41
2.7	Triangulation of Finnish and Hungarian words through English as a pivot.	51
2.8	Number of common translation pairs between methods.	54
2.9	Visualization of the evaluation of translation relations.	58
2.10	Precision of methods regarding the definition and example sentence relations.	61
3.1	Simplified data model of the database.	67
4.1	Quiz types in Canvas.	98
5.1	Entity editing form in the Dictionary Writing System.	110
5.2	Entity merging form in the Dictionary Writing System.	114
5.3	Splitting an entity in the entity editing form.	115
5.4	Relation validation page.	116
5.5	Bar chart of the entity validation results.	117
5.6	Bar chart example in the evaluation page.	119
5.7	The search bar of the dictionary.	122
5.8	Microstructure of dictionary entries.	122
5.9	Hungarian inflection information when hovering over the icon.	123
5.10	Microstructure of dictionary entries with more than one sense.	124
5.11	Homograph headwords are displayed as separate entries in the dictionary.	124
5.12	The order of senses is defined by their frequency, while translations within one sense appear in alphabetical order.	126

5.13	Example of a monolingual Finnish flashcard.	132
5.14	Example task to choose the correct case for a Finnish object.	137
5.15	Example task to conjugate the verb in the correct past tense in Finnish.	138
5.16	Example task to practice passive construction in Finnish.	138
5.17	Example of the Hungarian definite and indefinite conjugation task.	139
5.18	Example of the Hungarian preverb task.	141
5.19	Example task for the Hungarian possessive constructions.	142

List of abbreviations

BLI	bilingual lexicon induction
CALL	Computer-Assisted Language Learning
DWS	dictionary writing system
EGIDS	Expanded Graded Intergenerational Disruption Scale
FFL	Finnish as a foreign language
FK	foreign key
FU	Finno-Ugric
FULR	Finno-Ugric Lexical Resources
GIDS	Graded Intergenerational Disruption Scale
HFL	Hungarian as a foreign language
JSON	JavaScript Object Notation
L1	first language
L2	second language
LMS	learning management system
MALL	Mobile-Assisted Language Learning
MWE	multi-word expressions
NLP	Natural Language Processing
PK	primary key
SLA	second language acquisition
SLT	second language teaching
SQL	Structured Query Language
UD	Universal Dependencies
UI	user interface

Acknowledgements

This thesis and the work done towards it would not be possible without the contributions of many people. First and foremost, I would like to thank my opponents Dr. Gábor Prószéky and Dr. Csilla Horváth for their insightful and constructive suggestions. Their comments helped me shape my dissertation into its present form. I would like to express my deepest appreciation to the members of my committee for their invaluable suggestions regarding my manuscript and for posing thought-provoking questions regarding my research work.

I would like to thank my supervisor, Eszter Simon, for giving me the opportunity to work on this topic, and for the support and guidance throughout the years.

I would like to extend my thanks and gratitude to my professors at Pázmány Péter Catholic University from whom I have learned a lot and received lots of help and support.

I am also incredibly thankful to the professors of the Department of Finno-Ugric Studies at Eötvös Loránd Tudományegyetem (including Valéria Simon, Kata Kubínyi and Laura Bábai) who always had their doors open whenever I had any questions or doubts regarding this work. They provided me with many valuable insights and ideas from the very beginning of this PhD research.

I am deeply grateful to the volunteers who helped me validate the data sets. Without their help, this research would not have been possible. I would like to offer my special thanks to Simon Nguyen for checking so many of the relations and giving me constructive feedback many times. Thanks should go to Zsuzsanna Gyallai who not only helped me with validating the data set, but also put endless hours and efforts into the manual creation of Memrise courses with me.

Last but not least, I would like to thank my family and friends for their support, love, and unwavering belief in me throughout the years. I would never have gotten this far without you!

1 Introduction

1.1 Motivation

The present dissertation investigates the different possibilities of automatic dictionary building processes. During this research project, three new bilingual lexicon extraction techniques have been proposed which are applied to two Finno-Ugric (FU) languages with complex morphology. To evaluate their performance, the results of the proposed tools are compared to that of existing solutions applied to the same language pair.

This work also describes the practical outcome of the research, a language-independent lexicographical framework that has been built to store and share the obtained data using these methods. This framework – in its present state – provides a platform for two languages with rich morphology: Finnish and Hungarian, however, it is extensible with other languages as a result of the structure of its components. The framework consists of an online dictionary, a dictionary writing system (DWS), and accommodates a language learning application that uses the same data set that has been extracted with the above-mentioned methods.

Language learning motivation is a well-researched topic in the field of second language acquisition (Gardner and Lambert 1959; Gardner 1960; Clément 1980; Crookes and Schmidt 1991; Dörnyei 2005, 2009). There are many aspects that play an essential role in successfully acquiring a language and various theories and paradigms have been developed to account for students' desire and reasons for engaging in learning.

Some of the main reasons why people might like to learn Finnish according to Siitonen and Wessel (2020) include many personal aspects: the learners' interest in Finnish rock and metal music; their interest in Finnish nature and tourism, and their linguistic interest as Finnish is not an Indo-European language. Apart from these motivations, learning Finnish also plays an important role in the integration process among immigrant women (Ghaffar Ahmed and Mwai 2014).

International students who study in Hungary are often motivated to learn the local language in order to be part of the local community (Zhang 2018; Trujillo et al. 2020). Stamenkovska et al. (2022) found that students are motivated to learn this language because they want to socialize with Hungarians, understand the culture and also for other personal and practical reasons (such as finding

a job, renting a flat or attending events held in Hungarian).

My motivation to learn Finnish was mainly personal: I was always interested in its morphology and found it very different from other languages I had learned before. In Hungary, primary schools and high schools offer a variety of languages to learn as a second language (L2), however, it is very rare to see Finnish or Estonian among the options. There are some high schools where it is possible to choose Finnish (for instance, Leövey Klára Gimnázium in Budapest or Csokonai Vitéz Mihály Gimnázium in Debrecen). Sadly enough, having gone to Szent István Gimnázium in Budapest, I did not have the opportunity to learn either Finnish or Estonian. However, it did not prevent me from trying to learn Finnish autonomously. After high school, I finally had the chance to learn this language at Eötvös Loránd University, and I could not be happier. Until I realized that the resources available (such as online dictionaries, and tools to practice vocabulary) are mostly for speakers of English if they even exist. The group that I was part of while learning Finnish tried to make a difference, and we uploaded bilingual word lists in one of the online vocabulary learning platforms for later generations, and of course, for ourselves.

During my Master's program, I learned about Natural Language Processing (NLP) tools and techniques and saw how all the manual work we put into those bilingual word lists could have been saved and done in minutes, if not seconds by the computer. Since there still did not exist a high quality, large-scale, freely available online Finnish–Hungarian and Hungarian–Finnish dictionary, I applied for a PhD program to try to create one with automatic dictionary building methods, and at the same time, provide a detailed analysis of tools and techniques applied to these two languages with complex morphological system.

The framework presented in this thesis aims to serve as a useful lexical resource for future generations who speak any of the FU languages as a native language and attempt to learn another one of these languages. As will be presented in this work, the proposed framework also improves the interoperability of the already existing systems and resources for Finnish and Hungarian.

1.2 Vocabulary Acquisition and Lexicography

In order to understand and communicate in a foreign language, one must pay attention to and improve the two main components of a language: grammar and vocabulary. Vocabulary, or lexis, can be improved in many ways using many different resources, however, the credibility of the source

is crucial. The most effective way of acquiring new vocabulary items and how exactly the learners should encounter them in a language class are well-researched areas of second language acquisition (SLA) and second language teaching (SLT).

Sinclair and Renouf emphasized the importance of vocabulary learning and argued that “it is almost impossible to teach grammar without in passing teaching some vocabulary” (Sinclair and Renouf 1988: 143). Nevertheless, they also noted that it is extremely difficult to teach both lexis and grammar at the same time in an organized manner.

One way to learn new lexical items is through reading. As Waring and Nation (2004) suggested, reading has a great impact on foreign language vocabulary acquisition. They observed the correlation between reading in a foreign language and the growth of the lexicon that is gained with it. However, it is noteworthy that without sufficient vocabulary knowledge, reading can be tiresome, dull, and discouraging. A number of researchers have claimed that vocabulary knowledge strongly relates to reading proficiency, which implies that the more lexical items a person knows, the better he or she will comprehend foreign language text (Thorndike 1973; Anderson and Freebody 1981; Beck et al. 1982; Qian 2002). The number of vocabulary items that is necessary for good comprehension is about 3,000 to 5,000 of the highest frequency word families in a foreign language according to Nation and Waring (1997). This number is estimated to cover up to 84%-88.7% of the Brown corpus (Francis and Kucera 1982). Word family is defined as a group of words that – from the point of view of reading – “consists of a base word and all its derived and inflected forms that can be understood by a learner without having to learn each form separately” (Bauer and Nation 1993: 253). Nation and Waring suggested that, when the number of frequently occurring word families known by the reader reaches this approximation (3,000-5,000 word families), the unassisted comprehension of written texts can be expected.

Contrary to the findings of Nation and Waring, other research found that this threshold is higher. Nation (2006), for example, suggested 8,000–9,000 as the range of the optimal number of word families acquired, which would lead to an ideal text coverage of 98%. Furthermore, Laufer and Ravenhorst-Kalovski (2010) provided a minimal (4,000 to 5,000 word families) and an optimal (8,000 word families) number. The minimal threshold proves to result in the understanding of 95% of the running tokens of a text. On the other hand, knowing 8,000 word families yields a coverage of 98% of most written texts, in line with Nation (2006). It is worth noting that there can be huge

differences between languages. Hazenberg and Hulstun (1996) investigated how many base words a non-native speaker of Dutch needs to know in order to be able to comprehend university material. They came to the conclusion that the threshold is higher for Dutch: 10,000 words would lead to only 70% text coverage. However, they based their numbers on a dictionary which contains lemmata, instead of word families. Because of this difference, their findings must be considered with caution.

While lexical knowledge appears to enhance reading competence, and reading facilitates the learning of L2 vocabulary items, other research suggests that deliberate vocabulary study techniques (such as word cards, also called paired-associate learning) are also efficient (Elgort 2011). Word cards can be combined with spaced learning, which is an effective technique that promotes the repeated revision of vocabulary items over a given interval of time, and flashcards allow this revision to take place anytime and anywhere, due to their portability.

Nation (1980) presented vocabulary development strategies. Using word lists is efficient and it allows the acquisition of many words in a short period of time. He described the exact technique that the learners should apply in order to learn new vocabulary items and tips that can further boost learning. For instance, the use of a translation in the mother tongue proves to be more efficient, than the use of a synonym or definition in the target language. He also highlighted that word cards can be re-arranged, unlike word lists (written on a sheet of paper), which can help remember the word because of its position in the list.

Although Hungarian and Finnish are related languages, the links between the vocabularies of these languages are somewhat more obscure than, for instance, between the vocabularies of Estonian and Finnish (Jones 1995). It is therefore difficult to enhance lexical competence in Hungarian or Finnish, even if the native tongue of the learner is Finnish or Hungarian (respectively) since their lexis is quite different from one another. After observing the difficulties of learners of Finnish, Branch (s.a.), a lecturer in Finnish, emphasizes the following:

The greatest obstacle is the vocabulary, which requires memory; and the teacher cannot memorise for you.

Laakso (2015) highlights the same fact – vocabulary being the obstacle for learning Hungarian, instead of grammar, which can be learned in terms of interdependent rules – based on her own experience and that of many teachers.

In vocabulary acquisition, dictionaries and lexicons play an important role and help learners understand unknown lexical items. As will be shown in Section 1.2.3, there exists no large-scale, high-quality, online bilingual dictionary for the Finnish–Hungarian language pair which would constantly revise the meaning of words and keep the content up-to-date.

Constructing dictionaries is a complex process, and editors follow different guidelines when creating these resources. Lexicography is the study of lexicons, it is the art and craft of compiling and editing dictionaries and other reference works. Lexicography is hence closely related to dictionary creation, as Burkhanov (1998: 64) describes:

The term “dictionary” in a broad sense denotes the central category of lexicography, i.e. any reference work intended to provide linguistic information: pronunciation, spelling, lexical meaning, etymology, various aspects of usage, including valence, etc.

A dictionary cannot be a list of linguistic information without structure, without a system. However, the more carefully considered the design of a system is, the more useful and helpful the outcome can be for the end users. And as in many fields, the prospective users must be clearly identified and kept in mind while planning the structure of the product, since they are the benefiteres of the lexicographic processes.

Another important factor that must be mentioned is that languages change dynamically and the vocabulary also transforms continuously. It is therefore almost impossible to create a perfectly illustrative sample of it, which is why Atkins and Rundell (2008: 2) emphasized that “[a]ll dictionaries are incomplete, and come under the heading ‘work in progress’”. Hence, there is a constant need for newer editions of dictionaries, and none of the lexicons can ever be declared fully complete.

Bilingual dictionaries are undoubtedly one of the most valuable resources for language learners and professional translators. These reference works are mostly used when looking for a phrase or word in a foreign language to be able to express one’s thoughts or when looking up an unknown word to understand others.

The contents and size of a dictionary depend on several factors, such as the type of the dictionary (general or specialized, bilingual or monolingual dictionaries), the intended use (full-size or learner’s dictionaries), or the format (printed, online, etc.). In the case of a bilingual dictionary (otherwise known as a translation dictionary), further categorization is possible: it can be *monodi-*

rectional (only translating from one source language into the target language) and *bidirectional* (translating from language ‘A’ into language ‘B’ and from language ‘B’ into language ‘A’). Another approach to classify dictionaries may be based on the first language (L1) of the user: when the L1 of the dictionary user is the source language (left-hand side) of the dictionary, it is called an *encoding* dictionary, and when it is the target language (right-hand side), it is called a *decoding* dictionary.

There is an important decision that has to be made when building a dictionary: the order in which the entries are written. The editors of many dictionary projects decide to create the entries in alphabetical order, starting with the first letter of the alphabet. There is a major problem with this approach according to Mosel (2011): a project can run out of time and come to an end before the last entries could be finished. This can result in an unfinished dictionary covering, for instance, only the letters A, E, and H (as in the case of the Finnish–Hungarian–German dictionary started by József Budenz in the 1860s). It is less useful (even more in the case of endangered languages, as Mosel (2011) warns the reader, but also in general) than a dictionary in which the order of the entries to be written is defined using another approach. One way to avoid this issue is to compose the headword list in a thematic manner, starting with an arbitrary topic, and creating the entries for the headwords in that topic. This way, the writing of entries can be done in a shorter time, the collected data will cover a whole topic, and the resulting dictionary can be considered finished (although still incomplete, as mentioned above).

For many decades, all lexicographic work was done manually, however, with the shift from mechanical to digital technologies and the advent of computers, lexicography also underwent some major changes.

1.2.1 Electronic Lexicography

At the end of the 1950s and at the beginning of the 1960s, a new kind of lexicography emerged. After producing paper-based dictionaries and constructing their content manually for centuries, building electronic dictionaries became increasingly popular.

At the end of the 1990s, dictionaries started to appear in the online space, and each appearance was an attempt to adapt even more to this new context. Initially, it did not hinder the existence of printed dictionaries, however, with time, online dictionaries have become mainstream, causing a

reduced interest in paper-based lexicons and reference works. As handheld devices (such as tablets, smartphones, and laptops) and the Internet have become more widely used and accessible, digital, online lexical databases became also more recognized and widespread. According to the survey of Gaál (2016), dictionary users prefer online dictionaries, and the number of users of only paper-based dictionaries keeps decreasing. In the last 60 years, electronic dictionary building went through some major changes. In the 1960s, lexicographers and dictionary users had no contact with computers. These tools were initially used to organize the list of headwords in alphabetical order and it was the task of computer specialists to handle the data given to them by lexicographers and construct the dictionary from that data. Since then, it has changed a lot: graphical user interfaces make it easier for anyone who does not have strong technical skills to access and query data. Nowadays, a DWS is a fundamental part of all (electronic) lexicographic work and they are designed to be user-friendly, with the intention to enable lexicographers to use them without difficulty, and without advanced computer literacy skills.

Machine-readable dictionaries are indispensable when developing and fine-tuning certain NLP tools, for example, in the case of statistical machine translation systems or to help to map two sets of monolingual word embedding (numeric representation of words) spaces into a shared space. Since it is time-consuming and labor-intensive to manually compile a list of bilingual translation pairs from scratch to include in the dictionary to be published, and write the entries for each headword, there has been many attempts to automate this process and create both human- and machine-readable lexicons with the help of automatic methods.

One way to fasten this process is the attempt to collect data from the community. Wiki technology enabled the creation of crowd-sourced, community-based dictionaries, letting anyone interested in dictionaries and languages contribute to a dictionary project. These wiki-projects, however, are “of little scientific value” according to de Schryver (2003: 160), at least, without some validation methodology.

Another way to create dictionaries fast and without much human effort is by taking advantage of automatic, language processing methods and algorithms. Nevertheless, depending on the applied method, the outcome of these procedures (so-called proto-dictionaries, which consist of word pairs that did not undergo manual validation) might contain some erroneous translations, and their precision may vary greatly, never reaching 100%. These dictionaries might be considered unreliable

and useless for end users unless manual post-editing is carried out on their content.

The contents of a dictionary must be thoughtfully planned and composed. Whether a word forms part of a language and should be included in the dictionary must be decided by the lexicographers. To determine what is lexicographically relevant, dictionary editors used to rely on their intuitions about language. Linguistic evidence, which is an important part of lexicon building, was provided by citation slips until about 40 years ago, however, some lexicography projects may still use them today. Citation slips contain extracts from a text which can prove the existence of a word or phrase in a given language. During the creation of a dictionary and dictionary entries, these slips were used as the main data source. Since huge amounts of texts and language data are available now in digital form, corpora, which can serve as objective linguistic evidence, have replaced citation slips in many cases. One of the benefits of corpus-based methods in lexicography was highlighted in Mikhailov and Cooper (2016: 150):

A corpus-based dictionary would not give those artificial or invented equivalents that are so often found in 'old-fashioned' dictionaries.

In the compilation process of electronic dictionaries, three main strategies can be applied. It is possible to write a dictionary the traditional way, with the help of professional lexicographers who write the entries one by one and add all the information that they consider important and relevant about a specific headword to the corresponding entry. If this work is done using computers (and perhaps even a dictionary writing system) at any degree, it can be considered one of the typical examples of electronic dictionary building. Exploiting automatic lexicon extraction algorithms is a common practice nowadays. This method allows dictionary creators to extract information for the entries from different resources and display them in a certain way, which is determined by some guidelines. The third way is to collect data with the help of many volunteering contributors online, utilizing the power of communities and creating a crowd-sourced dictionary. Of course, there are many other ways to compile a lexicon, and combining two or more methods within the same project is also a regular solution.

Apart from the large amounts of (digital) texts and corpora that can be used as objective linguistic evidence in the dictionary creating process, several advantages can be attributed to electronic dictionaries.

1.2.2 Advantages of Electronic Dictionaries

From the early design stages of e-dictionaries, until the publication of the final product, several different processes can be employed to support and facilitate the work of lexicographers. In this section, a non-exhaustive list of the benefits of electronic lexicography is provided.

One of many advantages of an electronic dictionary (or e-dictionary) over a printed dictionary is that there are no space restrictions regarding the size or number of entries that the dictionary contains. The extent of the dictionary, however, must be considered and limited when building a paper-based dictionary. Space limitation is the reason why printed dictionaries abbreviate specific terms and frequently used phrases (e.g. part of speech tags, grammatical gender), or why they repeat the headword with a tilde (~) symbol. These space-saving strategies can easily lead to unwanted effects and misunderstanding. The usage of this symbol is not straightforward in the case of languages like Hungarian, where stem-internal vowel length can alternate depending on the suffix. How can we appropriately abbreviate the headword and indicate the alternation of the root-final vowel between word forms like *anya~anyák* ('mother'~'mothers')? Is it clear for speakers (or more importantly, for learners of the language in case of a learner's dictionary) that the vowel at the end of the root is changed if we substitute the plural form of *anya* with *~k*? Lexicographers do not necessarily have to resolve these kinds of issues, because space restriction is not one of the biggest concerns of electronic dictionaries.

Searching in digital dictionaries is faster and easier, facilitated by many additional functions and features. Being able to search in the main body of the entry, and the option to use regular expressions in order to find headwords matching a given pattern are definitely major benefits of e-dictionaries.

Another advantage is that an e-dictionary can be constantly updated and extended without having to reprint it over and over, the contents of the database can be re-used and another dictionary can be built on top of the same database (for example, containing only a part of the vocabulary present in the database to create a specialized dictionary).

As mentioned before, it is not an exhaustive list of all the benefits and features that electronic dictionaries have over paper-based dictionaries. There can be other additional features that make the use of the e-dictionary easier and the interface more user-friendly (such as customization, and hyperlinks between entries for faster navigation). For more advantages, see Granger and Paquot

(2012). However, it is imperative that the above-mentioned features be implemented and put into effect since these differentiate electronic dictionaries from paper dictionaries. Weschler and Pitts (2000) noticed that these advantages were not always exploited: “At present, electronic dictionaries are still fundamentally paper dictionaries on a microchip”. This observation still holds for many resources found online, as will be shown in the following section.

1.2.3 Finnish and Hungarian Bilingual Dictionaries

Despite being a less popular language pair, there exist several bilingual dictionaries in both directions for Finnish and Hungarian (Finnish–Hungarian and Hungarian–Finnish). According to Maticsák (2017), it can be explained by the relatedness of these languages, and the fact that Hungarian and Finnish have been taught at universities in Finland and Hungary, respectively, since the late 19th, early 20th century. From the beginning of the 1960s, the majority of Hungarian lecturers in Finland contributed in some way to the creation of textbooks, dictionaries, and other language learning materials.

Maticsák and Laihonen (2011b) analyzed seven Finnish–Hungarian and Hungarian–Finnish printed bilingual dictionaries in their critique. They argued that the quality of these dictionaries is surprisingly good. However, Maticsák and Laihonen (2011a) emphasized that updating and re-publishing such resources and dictionaries are necessary and crucial, since the vocabulary of a language is constantly changing, and words can have different meanings at different points in time. Additionally, the linguistic kinship of these languages does not help Finnish as a foreign language (FFL) and Hungarian as a foreign language (HFL) learners who have Hungarian or Finnish (respectively) as a mother tongue, since they are mutually unintelligible. Furthermore, learning these languages is an integral part of Finno-Ugric studies in both Hungary and Finland, hence, it is extremely important that these learners have easy access to high-quality, up-to-date bidirectional, bilingual dictionaries.

The first Finnish–Hungarian monodirectional lexicon was written by József Szinnyei, who composed a dictionary with approximately 15,000 headwords in 1884 (Szinnyei 1884). The aim of this reference work was to help Hungarian linguists read Finnish folklore, modern fiction, and newspapers. Following this work, István Papp published a modern Finnish–Hungarian dictionary with 48,000 entries (Papp 1962), which was updated, revised, and republished several times since its

first edition.

The most recent Finnish–Hungarian dictionary was published in Finland in 2015 (Forsberg et al. 2015). This is the first dictionary that was published in Finland, and that is a collaboration between Finnish and Hungarian lexicographers as the editors of the dictionary. The rest of the dictionaries ever compiled for this language pair were exclusively edited and published by Hungarian lexicographers and linguists. The number of headwords is more than 41,500. This is almost three times as many as the dictionary composed by Szinnyei. However, it took ten years of work to publish this most recent bilingual lexicon, so a question may arise: is it possible to facilitate this arduous work with the help of automatic extraction of translation pairs and help lexicographers with the creation of such resources?

Even though there exist numerous online dictionaries for this language pair, none of them provides both accurate and abundant data at the same time. Some of them are built automatically without any post-editing and validation, resulting in many erroneous translation pairs or in a dictionary that does not provide sufficient information about certain parts of the entry. Therefore, these are not always helpful for learners of Finnish and Hungarian. Other online dictionaries are edited by professionals, but – since the manual creation is costly and time-consuming – they only contain less than 20,000 entries, or the dictionary editing process lasts for several years or decades.

One of the biggest online resources for Finnish and Hungarian is Wiktionary, which contains data for not only the most widely spoken languages, but for several endangered (Livonian, Veps, Inari Saami, etc.) as well as artificial or constructed languages (Esperanto, Klingon, Dothraki, etc.). This resource is a crowd-sourced project, utilizing the power of the community. The data found in Wiktionary (either exclusively or partially) serves as the basis of many other online Finnish–Hungarian–Finnish dictionary projects, such as FinnHun¹, Sanakirja.org², and Ilmainen-Sanakirja.fi³.

Some dictionary projects combine the data found in several different resources on the Internet and use automatic methods to extract an initial vocabulary for many language pairs. After the extraction, they allow the dictionary users to be active contributors regarding the contents of the

¹ retrieved October 28, 2022 from <https://www.finnhun.com>

² retrieved October 28, 2022 from <https://www.sanakirja.org>

³ retrieved October 28, 2022 from <https://ilmainensanakirja.fi>

dictionary, and add new, or modify existing translations, example sentences and record the pronunciation of specific words. Two online dictionaries that utilize this kind of hybrid approach for Finnish and Hungarian – among many other language pairs – are Glosbe⁴ and DictZone⁵. These dictionaries do not provide an approval workflow that is supervised by professional lexicographers concerning the initially extracted vocabulary, or the modifications and new entries written by the community, nor does it use automated ways (so-called bots) to filter out unwanted content and obvious mistakes. These two dictionaries were created by only a few people (DictZone by one person located in Hungary, Glosbe by two friends located in Poland) with a strong programming background.

Another free online dictionary that has been manually constructed by lexicographers is dict.com⁶. There are several language pairs available in it, however, the manual compilation of dictionaries is indeed time-consuming: the development of these dictionaries started about 20 years ago, and the number of headwords is also quite limited: the Finnish–Hungarian direction contains 16,300, while the Hungarian–Finnish dictionary contains 17,400 headwords.

A big shortcoming of most of these dictionaries is that they cannot properly handle homonymy and polysemy. They do not provide definitions for the headwords that have more than one sense, nor do they use sense indicators that can help to find the correct word sense and the translation equivalent the user is looking for. Many times, an entry only contains a list of translations of the headword into the target language, which confuses language learners as to which translations they shall use. Figure 1.1 shows an example of the polysemous Hungarian word *toll* ‘pen, feather’, that appears with a list of possible Finnish translations in DictZone (*höyhen* ‘feather’, *kynä* ‘pencil, pen, quill’, *mustekynä* ‘ink pen’, *sulka* ‘flight feather’).

To make the usage and meaning of the headwords more clear and understandable, example sentences shall be listed in the entries, which is also a very rare phenomenon in the case of these freely available dictionaries. Some of them do have example sentences extracted from parallel corpora, but these appear on the bottom of the website without any distinction between the senses of the headword. This leads to an overcrowded, hardly readable or processable website, in which case

⁴ retrieved September 2, 2022 from <https://glosbe.com/fi/hu>

⁵ retrieved November 2, 2022 from <https://dictzone.com/>

⁶ retrieved September 2, 2022 from <https://dict.com/>

Magyar	Finn
toll	höyhen
	kynä
	mustekynä
	sulka

Figure 1.1: List of translations without sense indicators in DictZone.

the dictionary fails to meet one of the essential requirements of online dictionaries as established by Gaál (2012).

The attitude of users to different lexicon building methods is also a particularly important factor to consider when creating a resource for the community. Gaál (2016) observed that dictionary users consider dictionaries that take advantage of crowd-sourcing to be less reliable than those that are edited and published by different (language) institutes or dictionary publishers. On the other hand, free content is the preferred option among online dictionary users, even if it means that the entries are interrupted and split in half by numerous advertisements on the dictionary interface.

After analyzing the freely available bilingual and multilingual dictionaries on the Internet, it is clear that there is room for a Finnish–Hungarian–Finnish dictionary that takes into account the needs and expectations of dictionary users, while providing high-quality content, and up-to-date lexical information.

The present research attempts to fill this gap and provides a detailed comparison of multiple automatic bilingual lexicon building methods while producing a freely available, online Finnish–Hungarian–Finnish learner’s dictionary. Several factors of the dictionary writing process need to be optimized: the amount of human effort, time spent to create this resource, and the quality of the entries that are automatically generated. The main goal is to combine the speed of automatic dictionary building methods and the precision of manual editing and validation. To achieve this, a hybrid method has been implemented which starts with the automatic extraction of a bilingual list of prospective headwords, some fundamental information about these words, as well as additional entry components (such as example sentences and definitions). After collecting data automatically, manual validation ensures the high quality of the contents of the dictionary.

1.3 Grammar

As mentioned earlier, bilingual dictionaries are invaluable resources for language learning and professional translation processes. Dictionaries are repositories of words, which are commonly used to understand and use new lexical items, however, lexis is not the only area that needs attention when learning a foreign language. To be able to connect words and express a multitude of grammatical relations and functions within a sentence, one must know the different rules that exist in a given language. Carter and McCarthy (2013: 42) cite Wilkins' (1972: 111) book, in which he states that "[w]ithout grammar very little can be conveyed, without vocabulary nothing can be conveyed". As a consequence, learners have to improve their grammar skills as well as acquire a sufficiently large lexical knowledge, if they want to efficiently communicate in the target language.

Most FU languages, particularly Finnish and Hungarian, are considered to be on the synthetic, agglutinative end of the scale in terms of morphological types (Bergmann et al. 2007; Körtvélyessy 2017; Al A'amiri and Jameel 2019). This means that the word formation process depends on affixation, and it results in extensive case systems and rich morphological paradigms in these languages. Learning the correct usage of each grammatical case and the correct inflection or declension regarding each and every root can pose a big challenge for language learners. Providing an abundant amount of grammar exercises during foreign language classes might be helpful for learners, however, it would imply that teachers need to prepare and conduct these tasks manually, which requires tremendous amount of time and effort. Similarly to lexicon building automatization, it is worth examining if this manual work can be replaced with automatic methods and how efficient the outcome of such procedures would be regarding the precision and suitability of such tasks. Solutions to this issue are somewhat available within the framework of Computer-Assisted Language Learning (CALL), which are often supported by NLP methods to automate the processes. CALL will be shortly presented in the next section, and Chapter 4 provides a more detailed insight into the intersection of computers and language learning.

1.4 Computer-Assisted Language Learning

CALL is a term referring to the use of computer technology to support any kind of language learning. CALL is thus an interdisciplinary field that aims to provide exercises and systems for language

learners to improve different language skills at their own pace. Automatizing the generation of exercises is made possible by NLP-enhanced CALL.

There are several applications that are freely available and offer games, courses, and digital flashcards to engage learners in language activities daily in order to perfect their communicative skills and lexis. In what follows, some examples of these applications will be presented.

Nowadays, different tools and materials that can be used when learning a foreign language are more accessible than ever. The reason for this is not only the advancement of technology but also the fact that portable devices like smartphones and tablets constantly surround us. There are a great number of resources available at our fingertips. Online dictionaries, textbooks, machine translation systems, online courses, short videos explaining a certain peculiarity of a language, language learning applications, and websites that facilitate vocabulary building - everything we may need to practice a language is at our disposal.

The aim of language learning applications is to improve one's language skills, for instance, by teaching new lexical items, reviewing certain aspects of grammar, or improving listening skills. Many platforms offer online courses and flashcards to learn languages and vocabulary either for free or with a paid subscription.

There are flashcard software and applications like Memrise⁷, Quizlet⁸ and AnkiApp⁹, which facilitate the learning of new vocabulary items with a technique called spaced repetition. Since words are learned in an incremental way, and not in an "acquired/not acquired" manner (Schmitt 1998), repetition of the same lexical items is essential. Spaced repetition is a technique that makes learners repeatedly encounter the same parts of the learning material over time. The interval between two reviews of the same item depends on whether the learner remembers the item or not. When trying to improve one's lexical competence, it means that every vocabulary item is revised at systematic intervals, several times, until the learner can recall it without any difficulty.

Most of these applications, however, offer courses and materials only for a limited number of languages that are generally widely spoken (such as English, Spanish or Chinese). Languages that are less frequently chosen as foreign languages have fewer resources available. It is even harder

⁷ retrieved September 5, 2022 from <https://www.memrise.com/>

⁸ retrieved September 5, 2022 from <https://quizlet.com/>

⁹ retrieved September 5, 2022 from <https://www.ankiapp.com/>

to find appropriate and sufficient materials of the desired quality and level when not only the target language but also the source language (the native language of the learner) is one of the less commonly taught languages. There is a limited amount of materials available for Finnish and Hungarian as a target language, as indicated in Table 1.1, and it is mostly generated by the community, only a handful of applications provide high-quality materials created by professionals for these FU languages.

Application	Type	Finnish	Hungarian
AnkiApp	flashcards	C	C
Babbel	vocabulary, grammar lessons, listening exercises, speech recognition	-	-
Busuu	grammar lessons	-	-
Duolingo	flashcards	P	P
Memrise	flashcards	C	C
Quizlet	flashcards	C	C

Table 1.1: Language learning material availability for Finnish and Hungarian on different learning platforms. P stands for professional, and C stands for community-provided material availability.

AnkiApp allows users to create flashcards for any topic, or use the publicly shared cards of other users. The creators of the application do not provide verified, professional flashcards, only those in the publicly available cards can be used immediately after logging in. There are some language learning flashcards available for English, Chinese, French, Spanish, Japanese, and flashcards based on specialized terms and other topics (such as geography and medicine). At the time of writing, Finnish and Hungarian flashcards are not available in this application. The learners of these languages would need to create their own cards with the vocabulary items they want or need to learn.

Babbel¹⁰ is a subscription-based application that offers multiple-choice and listening exercises, while it has an additional feature that helps to improve pronunciation, which is implemented with an automatic speech recognition system.

Busuu¹¹ uses the spaced repetition technique, which means that if the learners remember the

¹⁰ retrieved October 12, 2022 from <https://www.babbel.com/>

¹¹ retrieved October 12, 2022 from <https://www.busuu.com/>

previously acquired item, the interval between two tests will become longer, while if they do not recall the vocabulary item, the system will display it again after a shorter period of time. It offers vocabulary and grammar lessons with audio recordings and writing exercises, which are corrected by native speakers. Nevertheless, some of these features are only available for learners who have a paid membership.

Neither Babbel nor Busuu offers courses or flashcards with which one can learn Hungarian or Finnish, and there is no opportunity to create custom flashcards and upload one's own material in a missing language on these platforms.

Duolingo¹² is a large-scale, commercial platform that attempts to improve listening skills, and offers virtual flashcards which contain words, phrases, or sometimes even whole sentences. When starting a new course, Duolingo demands the user to set a daily goal for studying, which is usually between 5 and 20 minutes a day. It has courses for both Finnish and Hungarian, although one must choose English as one's native language and learn it with the help of English translation equivalents. If Hungarian is chosen as one's native language, Finnish is not among the options, only English and German courses appear in the application. Although both the Finnish and the Hungarian courses include audio files to listen to the phrases to be learned, these are automatically generated and they are of low quality.

Memrise facilitates vocabulary acquisition by engaging learners with entertaining flashcards and uses spaced repetition to help the learners review and revisit already acquired words and phrases by displaying them and testing the learners' knowledge after a certain amount of time. Memrise also offers listening exercises, and a feature called "Learn With Locals", which provides videos of speakers saying the phrase at the natural speech rate, and demonstrating its meaning by mimicking it at the same time. On the online platform of Memrise, users can create courses and add their own flashcards (in the form of translation pairs, or lexemes and their definitions), which can be accessed and practiced by other users as well. These flashcards can be supplemented with additional information and attributes. Because of this feature, there are some rudimentary courses consisting of phrase and word pairs for Finnish and Hungarian, however, there are no additional materials or professional videos available for these languages provided by Memrise.

¹² retrieved September 5, 2022 from <https://www.duolingo.com/>

Quizlet provides a free platform for everyone who wants to create flashcards and study sets. There are several kinds of users, such as students, teachers, Quizlet Plus users, and verified creators. When looking for a specific topic among the study sets, it is possible to filter the results according to the kind of creator, which may affect the quality of the flashcards either positively or negatively. Finnish and Hungarian flashcards can be found on this platform that were created by the community members of Quizlet.

There is still much room for improvement, mainly regarding the quality and amount of courses and flashcards for Finnish and Hungarian, as was demonstrated with the platforms described above. Even though materials exist for Finnish and Hungarian as target languages, the source language in most cases is English. Therefore, the applications fail to provide resources for a huge number of potential users, who cannot speak English. Further projects and applications shall be implemented in order to fill this gap.

There is a recent tendency in the NLP community to help revitalize and reinvigorate languages at risk of disappearing and close to digital language extinction (Kornai 2013). Several systems have been created throughout the years that aim to categorize languages according to their degree of danger, for instance, the ranking of UNESCO (2003) is based on 9 factors, and the Expanded Graded Intergenerational Disruption Scale (EGIDS) (Lewis and Simons 2010) expands the scale initially proposed by Fishman (1991) (Graded Intergenerational Disruption Scale (GIDS)) and offers an evaluative system with 13 levels. The vast majority of the Uralic languages can be classified as (definitely, severely or critically) endangered or extinct on the UNESCO scale (Moseley 2010). The three FU languages that have national language status (the standard versions of Estonian, Finnish and Hungarian) are not considered endangered, there are numerous initiatives and projects whose goal is to create and develop NLP tools and resources for these languages (Tkachenko et al. 2013; Orasmaa et al. 2016; Haverinen et al. 2014; Oravec et al. 2014). Finnish and Hungarian (as well as English and Russian) are also often used when building tools for FU minority languages, since they are of high importance for the FU community (Moshagen et al. 2013; Simon et al. 2015; Vincze et al. 2015).

1.5 Research Questions

As a result of what was described in the previous sections, it is clear that there is a need for a high-quality, up-to-date bilingual dictionary for the Finnish–Hungarian language pair. The most recent bilingual paper-based dictionary was compiled more than 5 years ago, and the existing online dictionaries do not have a quality control system in place. Language learning applications either ignore these languages or provide some materials that can be produced with minimum effort.

Therefore, the research questions (RQ) that I am trying to investigate and answer in the present dissertation are the following:

- RQ 1. Which dictionary building method works best for languages like Finnish and Hungarian, i.e. agglutinative languages with rich morphology?
- RQ 2. Does language processing techniques, such as lemmatization, affect the results and help reach a better precision?
- RQ 3. Is it possible to create a language-independent database that stores the proto-dictionaries obtained by automatic methods, which can be the basis for a bilingual, automatically reversible online dictionary?
- RQ 4. Is it possible to create language learning exercises with automatic methods and predefined rules in order to help learners practice the most difficult aspects of these languages?
- RQ 5. Can a large number of exercises be generated with the help of different rules that would substitute the manual creation of such exercises done by foreign language instructors?
- RQ 6. How accurate are the examples of such an application? Are the applied rules general enough to cover multiple examples, but at the same time specific enough to only include suitable examples?

1.6 Roadmap

The present thesis is structured as follows:

Chapter 2 presents different solutions and methods to automatic bilingual dictionary generation, describes the resources that have been used to collect data in order to obtain an initial list of bilingual

translation pairs as the foundation of an online Finnish–Hungarian–Finnish bilingual dictionary, and reports the results of the manual evaluation of each method comparing them to the precision of an already existing tool, `wikt2dict` and its two functionalities.

After comparing some of the numerous possibilities of data storage technologies and database systems in Chapter 3, the data model and database schema are presented. The database is expected to store the extracted data in an efficient, non-redundant way. This chapter describes how the collected data set and carefully designed database can be queried in order to display valuable, meaningful information for evaluation as well as educational purposes.

Chapter 4 is devoted to the field of Computer-Assisted Language Learning (CALL). It gives a historical overview of the development of this field and examines the position and degree of integration of computer-assisted applications in language education. The chapter then presents the subfield of NLP-enhanced CALL, and demonstrates the limitations and shortcomings of the field.

The framework and web application that have been created in this doctoral research are presented in Chapter 5. Furthermore, it introduces the contents and the details of the online Finnish–Hungarian–Finnish dictionary interface and the CALL application. The chapter also describes the dictionary writing process, as well as the evaluation process regarding the examples of the CALL application, along with the results.

Chapter 6 summarizes the results of the thesis and provides some directions for further research regarding the dictionary, the dictionary writing system, and the language learning application.

The structure of the dissertation is depicted in Figure 1.2.

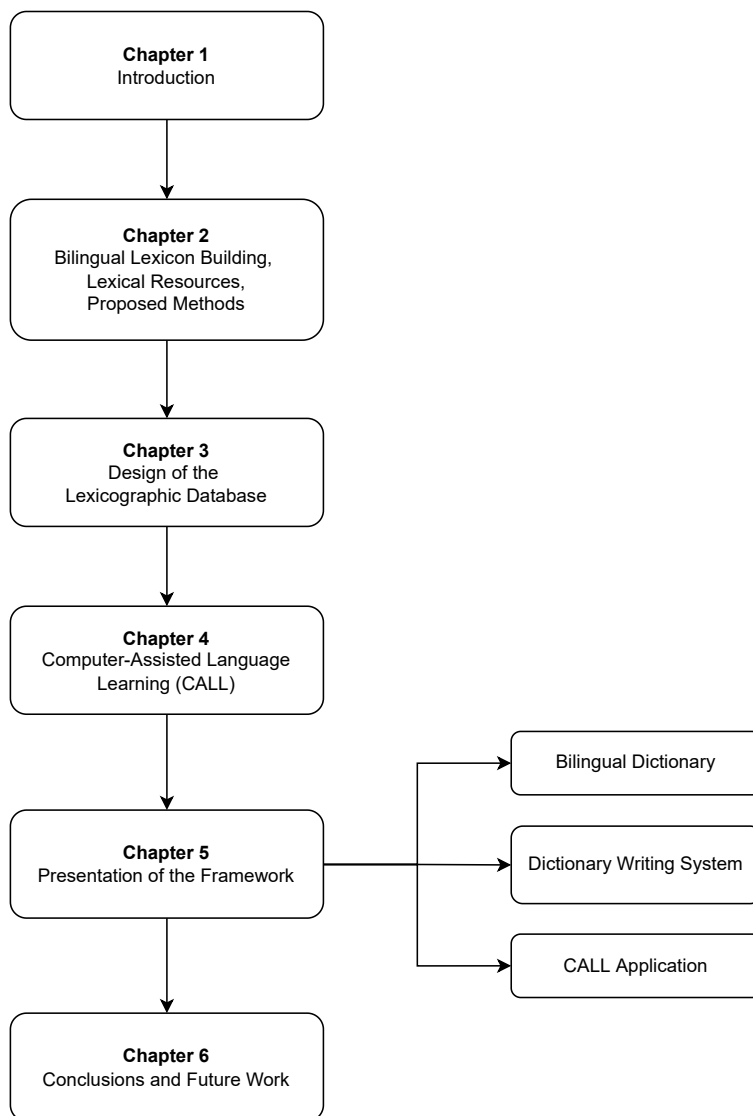


Figure 1.2: Outline of the dissertation.

2 Bilingual Dictionary Building Methods

Collecting the contents of a dictionary is one of the most important tasks of lexicon building. Language data is normally assembled, filtered and organized by lexicographers, whose job is to review it and write consistent dictionary entries with its help. As mentioned earlier, lexicographers used to take advantage mainly of citation slips as this was a usual form of empirical language data until the 1980s (Atkins and Rundell 2008: 48). With the emergence of million word corpora (which were developed for both Finnish and Hungarian in the beginning of the 21st century), citation slips were slowly replaced by concordance tools and big corpora as linguistic evidence. Introspection has always played a huge role in deciding what exactly to include in a dictionary and how to present language data the way it is used in a language. Relying on one’s intuition is still supported and preferred, even after the appearance of big corpora according to Sinclair (1991: 4):

Students of linguistics over many years have been urged to rely heavily on their intuitions and to prefer their intuitions to actual text where there was some discrepancy.

These two factors (i.e. large collections of texts and introspection) along with previous dictionaries and documents describing the language in question are the main sources of lexicographic evidence, as reported by Sinclair (1985).

The performance of NLP tools have been improved and perfected throughout the years, and billion word corpora are readily available for more and more languages. The automatic bilingual dictionary building methods – that consist of these two main components and combine the benefits of big data sets and language processing procedures – yield more precise results. In many cases, these results are used as the data sets that form the backbone of e-dictionaries (either with or without manual validation). The term “proto-dictionary” can refer to the collection of data that has not yet been manually corrected. Proto-dictionaries are defined as “automatically generated bilingual resources” by Héja (2015: 5). In this work, it is used similarly, referring to the collection of bilingual translation candidates, which has not yet been subject to manual post-editing.

The rest of this chapter is structured as follows. The next section presents some of the already existing NLP methods for dictionary creation. Section 2.2 mentions the biggest challenges regarding lexicon extraction for Finno-Ugric languages. Then the resources that were used for bilingual

lexicon creation for the Finnish–Hungarian language pair are introduced in Section 2.3, which is followed by the description of the language processing tools that were used during this research in Section 2.4. The detailed presentation of the proposed methods can be found in Section 2.5. In Section 2.6, the evaluation and the results of the proto-dictionaries are reported.

2.1 Existing Automatic Methods

There exist many approaches to how a list of bilingual translation candidates can be generated from different sources. Hence, the categorization of such methods can be based on the type of resource they exploit in order to create pairs of words and phrases.

The main resource categories which can provide a good basis for lexicon building methods regarding content are the following:

- parallel corpora,
- comparable corpora,
- monolingual corpora,
- existing dictionaries.

In the next sections, each resource will be presented in detail, and examples will be provided to how these kinds of data sets can be utilized and turned into bilingual translation pairs along with their benefits and shortcomings.

2.1.1 Parallel Corpora

A parallel corpus (also known as translation corpus) is a collection of bilingual (or multilingual) texts, which contain the translation(s) of an original text (as well as the original text itself), or which are originally written in two or more languages and are translated into further languages (Teubert 1996). Parallel corpora are normally aligned at sentence level, even though there might be some discrepancies, some deviations from the original text with respect to sentence boundaries.

Atkins and Rundell (2008) use the phrase “parallel corpus” as an umbrella term, which includes both translation corpora (similarly defined as above) and comparable corpora. A translation corpus, according to them, is a set of documents which consist of an original text written in one language which is then translated into another one (or several others).

It is perhaps noteworthy that two texts that are both translations of a third, original text (through a ‘hub’ language) are considered to compose a so-called “pseudo-parallel corpus” (Mikhailov and Cooper 2016), which decreases the number and size of texts that can be considered truly parallel corpora for a given language pair.

There are many well-known, large-scale parallel corpora that must be mentioned. The Bible which has been digitalized, translated into, and sentence aligned in more than 100 languages (Christodouloupoulos and Steedman 2015) can be considered one of the largest projects that aims to build a parallel corpus. Many parallel corpora (DCEP, JRC-Acquis, DGT-Acquis, etc.) and other language resources have been made available by European Union institutions. It is possible to access them on the website of the European Commission¹³.

One of the goals of CLARIN (Common Language Resources and Technology Infrastructure) is to collect the already available corpora, in order to be re-used within the frameworks of new research. CLARIN is a European Research Infrastructure offering data and NLP tools in order to improve Europe’s multilingual competence. In this infrastructure¹⁴, there exist 87 parallel corpora mostly for European languages, however, languages like Vietnamese, Tagalog and Hindi can also be found among the resources.

There are several corpus-based, statistical methods to translate words from one language into another, which exploit parallel texts.

Wu and Xia (1994) collect a parallel corpus for English and Chinese, align the sentences, and use a statistical method to extract English–Chinese bilingual lexicon from the corpus. They reach 91.2% precision for the single most probable translations.

Héja (2010) extracts core dictionaries to facilitate the creation of a Hungarian–Lithuanian dictionary for human use. The word pairs are obtained with the help of word alignments which are based on lemmatized parallel corpora. The evaluation of translation candidates is conducted on three probability bins (i.e. the data set is divided into three ranges according to the probability assigned to the translation candidates), and the best performance is observed for the range [0.7, 1), in which 97.2% of the translation candidates is lexicographically useful. The data bin, where the translation pairs have the probability of 1, on the other hand, only reaches 51%.

¹³ retrieved November 6, 2022 from https://joint-research-centre.ec.europa.eu/language-technology-resources_en

¹⁴ retrieved November 6, 2022 from <https://www.clarin.eu/resource-families/parallel-corpora>

Another approach for Chinese–English translation equivalent extraction is presented in Chang et al. (2002). They evaluate the methods which are based on 4 different statistical means, and give the accuracy of each method. The best performing χ^2 score achieves 81% accuracy for the Chinese–English translation equivalents.

There are many approaches that one can use in order to extract bilingual lexicons from parallel corpora and state-of-the-art methods can achieve higher and higher precision if given large enough data sets.

However, finding sufficiently large parallel corpora for a specific language pair is not a trivial task, since truly parallel texts are scarce. According to Rapp (1995: 320), finding a large-scale parallel corpus for a given language pair, in a specific field “will always be the exception, not the rule”. Most of the time, they are not readily available for some less-frequent language pairs (such as Finnish and Hungarian), while more common language pairs (like English–French) have available, large enough parallel corpora to be used for any purpose. If there are no available options for the desired language pair, the creation of a parallel corpus can be considered. However, compiling such a resource is costly and time-consuming. In order to create a learner’s dictionary based on parallel corpora, there is another design criterion that must be considered: they should contain only contemporary, general-language texts, as opposed to historical and domain-specific texts. Besides these drawbacks regarding parallel corpora, another issue must be pointed out here. Translated texts differ from their untranslated counterparts, meaning that a text that is the translation of another can hardly be regarded as a balanced, representative sample of the given language, which is disadvantageous when attempting to prepare the core vocabulary of a dictionary. To illustrate this point more in detail: McEnery and Xiao (1999) analyze and compare a translated Chinese corpus to authentic, L1 Chinese texts, and find that there is a statistically significant difference between the usage of perfective markers in these types of corpora.

2.1.2 Comparable and Monolingual Corpora

A monolingual corpus refers to a collection of texts in one language. A parallel corpus is in fact two or more monolingual corpora which are translations of each other, aligned at sentence level.

Comparable corpora, on the other hand, are composed of monolingual corpora which are not direct translations of each other, but are somewhat related, for instance, regarding their subject,

register or domain. As Teubert (1996) also underlines, the documents constituting a comparable corpus must be of equal size as well as equal composition. What we call monolingual corpora here are entirely unrelated documents in different languages.

To compile comparable corpora, large amounts of texts can be obtained by downloading data in multiple languages from the Web. It is of utmost importance that the sampling frame of the texts in the chosen languages is similar. Oftentimes the different language editions of Wikipedia are used as comparable corpora, as well as texts collected by web-crawling methods (such as the WaCky corpora (Baroni et al. 2009), and Aranea (Benko 2014)) or different newspaper articles from the same year, same domain, etc.

Bilingual lexicon induction (BLI) is a term which describes methods that identify word translations with the help of comparable or two or more unrelated monolingual corpora (Irvine and Callison-Burch 2017). Obtaining cross-lingual equivalents from parallel corpora is more of an alignment task. Distributional similarity metrics are used in order to induce pairs of translations and select the most probable translation equivalents. Supervised BLI methods generally make use of a small seed dictionary (a general bilingual dictionary), too.

Extracting bilingual translation equivalents from comparable corpora cannot be based on the positional co-occurrence of a word and its translation (Fung 1995), as it is the case for methods processing parallel corpora, since sentences are not directly aligned in the corresponding documents.

Many of these methods are based on lexical context analysis, i.e. examining the context of a target word and comparing it to the context of words in the other language. Rapp (1995) uses unrelated texts and co-occurrence matrices to determine translation equivalents. In order to extract word pairs, the order of words in one matrix is permuted until the similarity of the two matrices reaches a maximum. Then, the resulting word order in one matrix is supposedly identical to that of the other matrix, leading to translation pairs. Fung and Yee (1998) propose a method based on the hypothesis that a word is closely associated with some other words in its context and this holds for all languages. Hence, if two words in different languages share many context words, then they are probably translation equivalents. They apply this method to a Chinese–English non-parallel text, making use of the TF/IDF measure and a list of predefined word pairs that is used to determine the amount of shared context words.

A different approach is proposed by Otero (2007) to extract English–Spanish bilingual transla-

tion pairs from comparable corpora, which avoids the use of an external bilingual seed dictionary. Instead, pairs of bilingual lexico-syntactic templates are used which are obtained from small samples of parallel corpus. Considering only the best translation candidates for words, the precision of the method is 79%, while if we consider the top 10 candidates, it increases to 90%.

It can be seen that nearly identical results can be reached when exploiting comparable corpora to induce bilingual lexicons. Non-parallel, comparable texts are, however, more prevalent and accessible than parallel corpora. Creating comparable texts (by for instance crawling the web) is also more straightforward and less expensive than getting large amounts of texts translated by professional translators. For under-resourced languages and less common language pairs, the availability of large comparable corpora is greater than that of parallel texts.

One of the shortcomings of comparable corpora is mentioned by Morin and Prochasson (2011: 27), who state that “for technical domains, for which large corpora are not available, the results obtained, even though encouraging, are not completely satisfactory yet”. The bilingual lexicon building techniques based on non-parallel, specialized texts yield less precise results, and they presuppose the existence of an initial seed dictionary for the given language pair, which enables the comparison of the context words shared between any randomly chosen target word pair. Seed dictionaries can be avoided if parallel corpora exist for the languages in question, in order to obtain lexico-syntactic templates from these (sentence-aligned) resources. Parallel corpora or bilingual seed dictionaries are, however, not always available for any arbitrarily selected language pair.

Furthermore, without preprocessing the corpora before lexicon extraction, these methods would result in low quality output. Using large amounts of texts collected by Web crawling methods that are not cleaned and processed would distort the conclusions one can make from the data set, because repeated material as well as noise and mixed language content might be present in the corpora. The methods for comparable texts would also achieve less precision without lemmatizing the corpus first, especially in the case of languages that have complex morphology, which is also true for parallel corpora (see Héja (2010)).

Word Embeddings Word embeddings have attracted a lot of attention recently, and are useful in many NLP tasks. Each embedding represents a word with the help of a numeric vector. The numeric vectors are obtained using large collections of texts based on various learning methods. Word

embedding models have been applied to named entity recognition and chunking (Turian et al. 2010), sentiment analysis (Rezaeinia et al. 2019), and speech recognition (Bengio and Heigold 2014) to mention a few. Multilingual word embeddings and shared embedding spaces are also able to address cross-lingual tasks such as machine translation (Lample et al. 2018) and multilingual dependency parsing (Ammar et al. 2016). Although it might seem attractive at first to use this method to build a model which can provide translation candidates, in Søgaard et al. (2018) it can be seen that even for two closely related languages like English and German, the nearest neighbor graphs are not isomorphic. This means that the mapping between two word embedding spaces is almost impossible. The authors conduct multiple experiments, which include not only isolating and fusional languages with hardly any grammatical cases (like English, Spanish and German), but also agglutinative languages (like Estonian, Finnish, Hungarian and Turkish) which have at least 6 grammatical cases each. The cross-lingual embedding models underperform on the languages with complex morphology. The baseline method reaches 82.62% on the BLI task for the English–Spanish language pair, while English–Finnish and English–Estonian barely achieve 28.01% and 31.45%. The method for the Estonian–Finnish pair fails completely (24.35%). It appears that word embedding models underperform for the BLI task in the case of morphologically complex languages. Since Finnish and Hungarian both have inflectional morphology, it is expected that embedding models trained on the data of these languages would perform poorly, similarly to the Finnish–Estonian case (Søgaard et al. 2018). Furthermore, the training and fine-tuning of such models are computationally costly, data intensive, and the estimated power consumption is high. For these reasons, it was determined that using word embedding algorithms is beyond the scope of this research work. Nevertheless, it is one of the many avenues for future research on BLI for FU languages.

2.1.3 Alternative Solutions

There exist several resources for many languages and language pairs (including Finnish and Hungarian) which in general are not interoperable. Improving the infrastructure between these resources and creating a common interface would make it easier for language learners to consult only one online dictionary and find the necessary information easily in a short period of time. Instead of building an infrastructure that prioritizes the needs of dictionary users, the existing resources duplicate the material found in other sources and create different structures for the same content without

complementing it or adding new information to it, as previously mentioned (see Section 1.2.3).

It is possible to create new data sets using an alternative dictionary building (pivot language based, also called triangulation) method. As the name suggests, these methods use a third, intermediated language that is most commonly English between two, less represented languages. When two languages do not have sufficient data to compile comparable or parallel corpora for them, but (parallel or non-parallel) collections of texts exist for both languages paired with a third language, it is viable to apply pivot language based methods. The generation of bilingual lexicons is possible in two steps: translating from source to pivot, and then from pivot to target language words. This, however, is an oversimplification of the problem, since it does not consider word senses. When the pivot word has more than one senses, in the second translation step (from pivot to target language) it would translate into multiple target language words, which would lead to incorrect source–target translation pairs.

Tanaka and Umemura (1994) construct a Japanese–French dictionary with the help of English as an intermediate language. They attempt to tackle the issue of ambiguity of the pivot language words by a method called inverse consultation (IC). Since looking up words in only one direction through the pivot language might result in many faulty equivalence candidates, checking the result set again using the inverse dictionary may help to reduce the number of candidates by measuring the nearness of the senses of words in different languages.

Saralegi et al. (2011) analyze the inverse consultation method in detail (the solution of Tanaka and Umemura (1994) to mitigate the ambiguity problem). They show that IC can solve only a very small subset (17%) of ambiguous entries when applied to the task of compiling a Basque–Spanish dictionary with English as pivot.

There are several other projects that aim to build bilingual lexicons with the help of a pivot language, such as Shirai and Yamamoto (2001), Sjöbergh (2005) and Varga and Yokoyama (2009).

Varga and Yokoyama (2009) compile an English pivoted Japanese–Hungarian dictionary while trying to achieve high precision (ratio of the number of relevant elements identified by the model to the total number of elements identified by the model) with high recall (ratio of the number of relevant elements identified by the model to the total number of relevant elements). To reach high recall, they introduce bidirectional selection which makes sure that every source word, as well as every target word, has at least one translation available. In order to increase precision, they implement

semantic information extraction using the English WordNet (Miller 1995).

WordNet is a large database which organizes nouns, verbs, adjectives and adverbs into synonym sets, which represent different concepts, instead of organizing entries alphabetically like a traditional dictionary. (More details about WordNet can be found in Section 2.3.1.) In the translation selection and scoring phase, Varga and Yokoyama utilize the semantic information extracted from WordNet in order to determine the probability of a translation candidate. Evaluation shows that using WordNet as an alternative to strictly lexical overlap improves the precision of the method. However, multiple issues are mentioned by the authors that must be paid attention to in connection with this method. Certain translation pairs are not extracted, because they are missing from the entries of one of the dictionaries. WordNet is also not a complete lexical database of the English language and all its words, hence, synonyms and antonyms, as well as hypernyms and hyponyms of a given word may be absent.

Wikipedia and Wiktionary are both crowd-sourced, wiki-based resources, edited by a community of volunteers. As mentioned earlier, dictionary users have a dispreference for crowd-sourced projects. However, these resources appear to be of surprisingly high quality, and many language technology projects show interest in the use of these resources in relation to several NLP tasks (e.g. question answering (Ahn et al. 2004), multilingual named entity recognition (Richman and Schone 2008), word sense disambiguation (Mihalcea 2007) and idiom identification (Muzny and Zettlemoyer 2013)).

Simon et al. (2015) uses Wikipedia and Wiktionary to create bilingual dictionaries for language pairs containing one of six small Finno-Ugric languages (Komi-Zyrian, Komi-Permyak, Udmurt, Meadow and Hill Mari and Northern Saami) paired up with one of four major languages (English, Russian, Hungarian, Finnish). They compile parallel and comparable corpora for these 24 language pairs in total, and conduct many alternative dictionary building methods. Using the interwiki links, part of the proto-dictionary is obtained from Wikipedia title pairs. They also exploit Wiktionary by applying the method proposed in Ács et al. (2013), which extracts translation candidates from translation tables of the entries. The evaluation of the proto-dictionaries for the 4 language pairs including Northern Saami and the four major languages can be found in Simon and Mittelholcz (2017). Another kind of dictionary building method is applied to three of these language pairs (excluding the Northern Saami–Russian pair). This method is based on the dictionaries available

in the OPUS corpus¹⁵. The resource was generated from a parallel corpus of KDE4 localizations. The evaluation shows that the lowest performing procedure uses the KDE4 localization parallel corpora (with 29.23% precision), probably because they were generated from running text full of inflected and derived words, and non-dictionary forms. The best performance, on the other hand, was reached by the method collecting Wikipedia title pairs.

2.2 Automatic Bilingual Dictionary Building for Finno-Ugric Languages

The previously presented methods are generally based on well-resourced language pairs. When at least one of the languages to which bilingual translation candidates are generated is a lower resourced or morphologically complex language, the quality of the applied method decreases.

This decrease in quality can be caused by the morphological complexity of languages. To verify this observation, previous research investigated the effects of lemmatization. Cotterell et al. (2018) show that there is a correlation between morphological richness and language model performance. When inflectional morphology is stripped away by lemmatizing the initial corpora, this correlation disappears, which can be the proof that using lemmatized data results in better performance. This result was achieved on a language modeling task, therefore, it still remains a question whether the same observation can be made in relation to BLI methods. Hence, one of the research questions (RQ 2.) that will be investigated in the present chapter is whether or not lemmatization has a positive effect on the precision of the obtained proto-dictionaries.

In Section 2.1, many research projects were mentioned that aimed to extract word pairs for language pairs including either Hungarian or Finnish (Varga and Yokoyama 2009; Héja 2015; Simon et al. 2015). Héja (2015) generates Hungarian–Lithuanian and Hungarian–Slovenian dictionaries using parallel corpora. Varga and Yokoyama (2009) use pivot language based methods and tackle the issue of ambiguity with the help of WordNet. Simon et al. (2015) extract data from Wiktionary and Wikipedia with alternative methods and also apply a triangulation method through a pivot language using Wiktionary translation tables. In Simon and Mittelholcz (2017), they extend the used resources with the parallel OPUS corpus, and show that the precision of the word pairs extracted from this resource is quite discouraging for the morphologically rich Northern Saami.

¹⁵ retrieved 13 October, 2022 from <https://opus.nlpl.eu/>

After the examination of the applied methods for Finnish and Hungarian, there seems to be some unexplored areas of BLI that utilize resources that were either not available before, or not yet discovered as potential sources of lexicon induction. To the best of the author’s knowledge, there does not exist any method that would obtain translation equivalents by

- connecting the Finnish and Hungarian wordnets,
- extracting word pairs from the Finnish and Hungarian versions of Wiktionary, and
- generate translation candidates using the Finnish–Hungarian word alignment files in the OPUS corpus.

This work aims to provide solutions to these alternative lexicon creation possibilities for Finnish and Hungarian and examine their performance. These methods shall automatically produce large amounts of translation candidates which can be obtained from the above mentioned, existing resources, while being computationally inexpensive and replicable. The resources and freely available dictionaries that will serve as the source of these bilingual lexicon extraction methods will be presented in the next section. After obtaining the proto-dictionaries, the manual validation of a representative sample will make it possible to compare the performance of the proposed solutions to that of other, already existing techniques.

2.3 Resources

There are many different resources which facilitate automatic dictionary building. These resources can be divided into two groups according to the medium they appear in (i.e. printed or electronic materials). A resource can also be described by the number of languages it contains: monolingual, bilingual or multilingual. It is possible to create newer resources for a language pair that is already well-resourced, but improving the interoperability of already existing sources and materials by creating a framework that incorporates them and comparing their quality can also be of great value.

There are several monolingual and bilingual online dictionaries that contain lexical information for Finnish and Hungarian. These can be used in order to generate a list of Finnish–Hungarian translation candidates by applying different rules depending on the structure of the resource.

Since knowing the exact structure of such materials is a key factor when attempting to create proto-dictionaries from them, their construction is described in detail in the next sections.

2.3.1 Wordnets

A wordnet is a large lexical database whose primary aim is to provide a machine-readable semantic network which can improve the performance of NLP applications. The basic building blocks of wordnets are called synonym sets (or *synsets* for short). A synset is a set of cognitive synonyms, a group of words describing a certain concept, having the same meaning. Wordnets provide a unique identifier for each synset (called a *synset offset*) which helps to identify the exact sense of a given word. It is especially useful when a language has lots of homonyms and polysemes, i.e. when a certain (written) word form has more than one senses. Depending on the wordnet edition, glosses may be provided which describe the meaning of synsets and example sentences that illustrate the usage of the words.

The very first wordnet (called Princeton WordNet) was created for English (Miller 1995), and over the years several other language editions have been created. In the case of the Hungarian edition, also called HuWN (Miháلتz et al. 2008), some machine-translation heuristics were used. The output of this automated translation was then manually examined by professionals, who corrected or completed the automatically translated synsets to create a high-quality resource. Professional translators translated manually the Princeton WordNet 3.0 into Finnish (Lindén and Carlson 2010) and in the newer, 2.0 version of the Finnish WordNet (FinnWordNet), translators also extended the initial list of synsets by adding new hyponyms linking them to the existing concepts.

The execution of manual validation and the help of professional translators guarantee the high quality of these lexical databases.

The synset offsets offer an opportunity to connect two different language editions and construct bilingual lexicons by finding the intersection of the two synsets. It is perhaps worth noting that the same synset offset can appear in connection with more than one part of speech categories, so it is necessary to keep the part of speech information along with the identifier of the synset when trying to link the words of two wordnets.

It is also important that synset offsets must be based on the same set of identifiers in order to create the correct connection between two language editions. With each version change of the

Princeton WordNet, new synset offsets are generated for the concept sets. Wordnets in other languages, hence, can use mismatching identifiers, because they may be the translations of different Princeton WordNet versions.

This applies to the Hungarian and Finnish wordnets, too. The contents of HuWN were based on Princeton WordNet version 2.0 and the synsets from BalkaNet Concept Set. These were extended with several synsets to include the most important Hungarian concepts. The initial synset offsets present in the Hungarian database were later mapped to match the v3.0 offsets, including both identifiers in relation to most of the concepts. The FinnWordNet was translated from the Princeton WordNet version 3.0, using its synset offsets. If the Hungarian offsets had not been mapped to the 3.0 identifiers, an additional step would have been necessary to tackle this issue.

The proposed method (described in Section 2.5.1) is based on the HuWN git (version control system) commit b9641132 (November 10, 2018) which can be found on GitHub¹⁶. Synsets and additional data are saved in XML format. This file contains 42,288 synsets, 60,463 word senses and 50,238 words. The distribution of these numbers among the 4 parts of speech can be seen in Table 2.1.

Part of Speech	# of Synsets	# of Word Senses	# of Words
Nouns	33,530	45,508	39,079
Verbs	3,607	6,947	4,905
Adjectives	4,112	6,215	4,900
Adverbs	1,039	1,793	1,354
Total	42,288	60,463	50,238

Table 2.1: Number of synsets, word senses and words in the Hungarian WordNet grouped by parts of speech.

Regarding the Finnish WordNet, its latest version (v2.0) was downloaded and used¹⁷. The ZIP archive contains several files. The number of synsets and other data in the FinnWordNet can be found in Table 2.2. According to these numbers, and the ones found in Table 2.1, it is clear that the FinnWordNet contains almost 3 times more synsets and words than its Hungarian counterpart.

By definition, a synset describing a certain concept can contain more than one words or expressions. It can be observed in Tables 2.1 and 2.2 that the number of synsets is smaller than the number

¹⁶ retrieved June 4, 2020 from <https://github.com/mmihaltz/huwn>

¹⁷ retrieved June 9, 2020 from <https://korp.csc.fi/download/FinnWordNet/v2.0/>

Part of Speech	# of Synsets	# of Word Senses	# of Words
Nouns	84,905	142,763	108,557
Verbs	13,767	26,252	11,108
Adjectives	18,156	33,246	18,435
Adverbs	3,621	6,384	3,978
Total	120,449	208,645	142,078

Table 2.2: Number of synsets, word senses and words belonging to each part of speech in the Finnish WordNet.

of words in both wordnets. 56% of all Finnish synsets contain more than one word, while 26% of all Hungarian synsets consist of at least two lexical units. The average number of words in a synset is 1.732 in the Finnish edition, and 1.430 in the Hungarian.

The structure of the `rels/fiwn-transls.tsv` file that is used by the linking algorithm (described below in Section 2.5.1), is shown in Figure 2.1. TSV (or tab-separated values) is a text-based file format, that (as the name suggests) contains values separated by tab characters. The values of the file that the algorithm uses consist of the synset offset prefixed with the part of speech abbreviation, the word that is part of the synset, the Princeton WordNet 3.0 synset offset, an English word from the original synset and the relation type between the Finnish and English words (which can be `synonym`, `near_synonym`, `hypernym`, or `hyponym`). In this figure it can be seen that the first three columns are repeated wherever the English synset contains more than one word.

```

fi:a01329413    homogenoitu    en-3.0:a01329413    homogenised    synonym
fi:n07710616    peruna    en-3.0:n07710616    Irish potato    synonym
fi:n07710616    peruna    en-3.0:n07710616    potato    synonym
fi:n07710616    peruna    en-3.0:n07710616    white potato    synonym
fi:n12130937    järviruoko    en-3.0:n12130937    carrizo    synonym
fi:n12130937    järviruoko    en-3.0:n12130937    ditch reed    synonym

```

Figure 2.1: The structure of the Finnish WordNet.

The Hungarian WordNet file, on the other hand, uses the XML structure, which is illustrated with a synset in Figure 2.2. Besides the Princeton WordNet 2.0 and 3.0 synset offsets (`<ID>` and `<ID3>` nodes), the part of speech tag (in the `<POS>` node) and several other data, such as a gloss (`<DEF>`) and an example sentence (`<USAGE>`) that demonstrates the usage of the Hungarian word can be found. The words that form part of the synset are contained within the `<SYNONYM>` node, each word in separate `<LITERAL>` nodes.

```

<SYNSET>
  <ID>ENG20-07262056-n</ID>
  <ID3>ENG30-07734017-n</ID3>
  <POS>n</POS>
  <SYNONYM><LITERAL>paradicsom<SENSE>2</SENSE></LITERAL></SYNONYM>
  <ILR>ENG20-07235951-n<TYPE>hypernym</TYPE></ILR>
  <ILR>ENG20-12156165-n<TYPE>holo_part</TYPE></ILR>
  <DEF>Piros, húsos, leves, bogyótermésű konyhakerti növény.</DEF>
  <USAGE>A paradicsom az olasz konyhaművészet egyik alapvető anyaga.</USAGE>
  <STAMP>nkfp 2006/07/26</STAMP>
  <DOMAIN>gastronomy</DOMAIN>
  <SUMO>FruitOrVegetable<TYPE>+</TYPE></SUMO>
  <EKSZ>paradicsom_2_1<TYPE>=</TYPE></EKSZ>
  <EKSZ>paradicsom_2_2<TYPE>=</TYPE></EKSZ>
  <EKSZ>paradicsom_2_3<TYPE>=</TYPE></EKSZ>
</SYNSET>

```

Figure 2.2: The structure of a synset in the Hungarian WordNet.

2.3.2 Wiktionary

Wiktionary is an online multilingual dictionary project which aims to include as many lexical data as possible for as many languages as possible. As the sister project of Wikipedia, Wiktionary also uses crowd-sourcing to populate its database with different language data.

The results of bilingual dictionary building methods that exploit the power of crowd-sourcing can be more efficient and robust than that of parallel and comparable corpora-based methods. Ács et al. (2013: 57) report that building dictionaries from parallel and comparable corpora cannot add many new translations with high precision when compared to crowd-sourced methods and the former methods are not as powerful as the latter.

The reason is simply the lack of sufficiently large parallel and near-parallel data sets, even among the most commonly taught languages.

There are several Wiktionary editions. The difference between these editions is their target language, which defines the language of the translations, the descriptions, and the user interface. As of November 2022, there are 162 active Wiktionary editions. The source language in each edition can be any language, for example, one can look up English, Finnish or German words in the Hungarian Wiktionary¹⁸, Estonian or Latin words in the Finnish edition¹⁹, or Chinese or Turkish

¹⁸ retrieved October 7, 2022 from <https://hu.wiktionary.org/>

¹⁹ retrieved October 7, 2022 from <https://fi.wiktionary.org>

words in the English language Wiktionary²⁰. In case of each Wiktionary edition, statistics about the number of edited pages, active editors and many others can be found on the Wikimedia website²¹.

A Wiktionary entry has strict rules that must be followed when a volunteer edits an existing page or creates a new one. There are obligatory sections that must be included by all means, and there are optional sections, which can be left out. These vary from edition to edition, but in general, the headword, the language of the headword, its part of speech, as well as one or more translations in the language of the Wiktionary edition must be present. There is a special case, when the translation of the headword is in fact not a translation, but a full definition in the language of the Wiktionary edition — namely when the language of the Wiktionary edition coincides with that of the headword. Definitions explain the meaning of particular senses in detail. It is illustrated in Figure 2.3, where the Finnish headword (*poika* ‘boy, son’) is looked up in the Finnish Wiktionary.

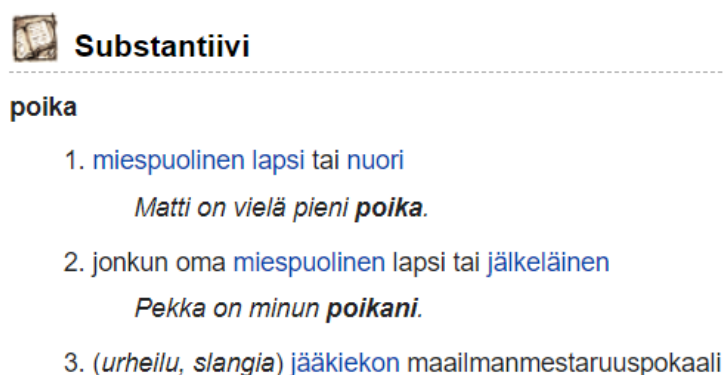


Figure 2.3: Definitions in Wiktionary when the language of the headword and that of the Wiktionary edition are the same.

Another important section included in many entries is the translation table. It appears only when definitions do: when the language of the headword is identical to the language of the Wiktionary edition. This table contains equivalents of the word in other languages, and the information is separated into different tables in case the headword has many senses. Since this section is not obligatory, translation tables appear only in a fraction of the entries. An example translation table (along with a collapsed, hidden table above) can be found in Figure 2.4.

Additional information may be added to entries, such as the etymology of the headword, its

²⁰ retrieved September 23, 2022 from <https://en.wiktionary.org/>

²¹ retrieved June 8, 2022 from <https://stats.wikimedia.org>

Translations [edit]

±suspension of solid in liquid [show ▼]	
±any gel for a particular cosmetic use [hide ▲]	
Select targeted languages	
<ul style="list-style-type: none"> • Catalan: <i>gel</i> ^(ca) <i>m</i> • Chinese: <ul style="list-style-type: none"> Mandarin: 凝膠 ^(zh), 凝膠 ^(zh) (níngjiāo) • Czech: <i>gel</i> ^(cs) <i>m</i> • Dutch: <i>gel</i> ^(nl) <i>m</i> or <i>n</i> • Finnish: <i>geeli</i> ^(fi) • French: <i>gel</i> ^(fr) <i>m</i> • Galician: <i>xel</i> <i>m</i> • German: <i>Gel</i> ^(de) <i>n</i> • Greek: ζελέ ^(el) <i>n</i> (zelé), τζελά ^(el) <i>n</i> (tzel) 	<ul style="list-style-type: none"> • Hebrew: ג'ל ^(he) <i>m</i> (djel), תקריש ^(he) <i>m</i> (takrísh) • Hungarian: <i>gél</i>, <i>zselé</i> ^(hu) • Japanese: ジェル ^(jeru), ジェル ^(jeru) • Maori: <i>pia</i> • Persian: ژل ^(fa) (žel) • Polish: <i>żel</i> ^(pl) <i>m</i> • Portuguese: <i>gel</i> ^(pt) <i>m</i> • Russian: гeль ^(ru) <i>m</i> (gel') • Tagalog: <i>pamada</i> • Thai: เจล ^(th) (jeel) • Turkish: <i>jel</i> ^(tr), <i>jöle</i> ^(tr)

Figure 2.4: Translation tables from the English Wiktionary entry *gel*.

pronunciation, alternative forms, usage notes, derived terms, and many more.

As an example, the entry for the headword *kiitos* ‘thank you’ can be seen in Figure 2.5. It is taken from the English Wiktionary edition. Fundamental, necessary information about the headword (such as its part of speech, and one translation equivalent), as well as additional, optional sections (such as etymology and pronunciation) can be found in this entry.

There exists several methods and tools to extract bilingual dictionaries from Wiktionary data (such as *wikt2dict* proposed in Ács et al. (2013) and *wikokit* by Krizhanovsky and Smirnov (2013)). However, *wikokit* was developed to parse the structure of the English and Russian Wiktionary editions, while *wikt2dict* is language-independent and can be used on many more Wiktionaries. *Wikt2dict* parses the translation tables that are only found in a subset of the existing entries.

To the best of the author’s knowledge, there does not exist any tool that obtains Finnish–Hungarian translation candidates using the headwords of the entries in the Hungarian and Finnish Wiktionary editions and the translation sections. In Section 2.5.2, a tool called *Wiktionary Parser*, which was created for that very purpose, will be described.

Finnish [\[edit \]](#)

Etymology [\[edit \]](#)

kiittää + *-os*

Pronunciation [\[edit \]](#)

- IPA^(key): /'ki:tos/, ['ki:tos]
- Audio: 
- Rhymes: -i:tos
- Syllabification: kii·tos

Interjection [\[edit \]](#)

kiitos

1. [thank you](#)

Figure 2.5: Finnish entry in the English Wiktionary edition.

2.3.3 OPUS

The OPUS corpus (Tiedemann and Nygaard 2004) is a collection of publicly available data collected and aligned at sentence level with automatic methods. The corpus contains free data such as subtitles, documents of the European Commission, software localization. These are compiled and pre-processed automatically, however, no manual validation is carried out by the creators in order to provide only clean data sets. The OPUS corpus contains resources for several languages and language pairs, and provides different types of resources, including parallel texts, bilingual word alignments, and phrase tables, among others. There exists a few resources for the Finnish–Hungarian language pair in OPUS, which can be utilized to get bilingual translation candidates. A small subset of these lists is shown in Figure 2.6, where the columns contain – besides the word alignments in the third and fourth fields – the number of co-occurrences (first column), Dice similarity coefficient (second column), and other scores retrieved from the phrase tables.

In the next section, the language processing tools will be presented which were used in order to analyze different, non-canonical word forms in Finnish and Hungarian, and bring them to the most common word form by means of lemmatization. With the help of this text-processing procedure, it will be possible to answer one of the research questions (RQ 2.) and determine if lemmatization

11	0.107843137254902	peruna	burgonyából	0.0728477	0.0839695
12	0.151898734177215	perunaa	burgonyát	0.10084	0.103704
3	0.4	perunaerän	burgonyatétel	0.375	0.428571
5	0.0641025641025641	perunajalosteista	burgonyából	0.0331126	0.0381679
2	0.6666666666666667	perunakasveja	burgonyanövényektől	0.666667	1
2	0.4444444444444444	perunakasvien	burgonyanövények	0.333333	0.222222
2	0.4	perunakasvit	burgonyanövények	0.333333	0.222222
2	0.571428571428571	perunalajikkeen	burgonyafajta	0.666667	0.4
2	0.2666666666666667	perunalajikkeiden	burgonyafajták	0.166667	0.285714
2	0.285714285714286	perunalajikkeita	burgonyának	0.2	0.2
2	0.285714285714286	perunalastuja	burgonyaszírom	0.2	0.125

Figure 2.6: Excerpt from the OPUS word alignments.

affects the precision of translation pairs in a positive way. The NLP tools presented below are able to generate part of speech tags, which are applied in this case for words where this information is not known. The bilingual lexicon extraction methods will be described in Section 2.5: three methods that are proposed in this project in order to collect translation candidates from the resources presented in this section, and one already existing method proposed in Ács et al. (2013), that is used to properly compare the precision of the newly created methods to a baseline which extracts translation candidates from Wiktionary, similarly to one of the proposed techniques.

2.4 Language Processing Tools

In the previous sections, the resources were presented. In some cases, these resources do not exclusively contain the canonical form of words (their lemma), but also other inflected word forms, since the data have not been lemmatized.

In order to extract a list of potential headwords from these sources, and to make the translation candidates as precise and lexicographically useful as possible, the non-canonical word forms must be identified and replaced with the lemma of the word. To achieve this goal, lemmatization needs to be conducted on a subset of the obtained data. Wiktionary and WordNet mostly contain the lemma of words, while OPUS word alignments are extracted from running texts (subtitles, documentation, etc.) without pre-processing them first. It leads to a list of translation candidates containing many inflected word forms of the same root (which can be seen in Figure 2.6). Lemmatization must be hence carried out on the output of the algorithm that extracts translation candidates from the OPUS corpus (presented in Section 2.5.3 below).

Moreover, morphological analysis is also valuable, since it can provide part of speech tags for

words where this information is unavailable.

`omorfi` (Pirinen 2015) is an open-source rule-based morphological analyzer for Finnish. It is an easy-to-install and easy-to-use tool, which involves a specific script that is able to yield output compatible with the Universal Dependencies (UD) CoNLL-U format.

UD is a framework that provides consistent grammar annotation across different languages (de Marneffe et al. 2021), combining several tagsets and guidelines. CoNLL-U is a standard file format used by UD in which only three types of lines can be present:

- blank lines (which mark sentence boundaries),
- comment lines (starting with hash marks (#)), and
- word lines (which contain 10 fields separated by tab characters).

The field names along with an explanation of their usage can be seen in Table 2.3. Out of these ten fields, the `omorfi` script that produces output in CoNLL-U format (`omorfi-conllu.bash`) can fill in the LEMMA, UPOS, and FEATS fields.

Field Name	Usage
ID	Word index starting at 1 for each new sentence.
FORM	The word form.
LEMMA	Lemma of the word form.
UPOS	Universal part of speech tag.
XPOS	Language-specific part of speech tag.
FEATS	Morphological features list.
HEAD	The ID of the head of the current word.
DEPREL	Dependency relation to the head.
DEPS	Enhanced dependency graph.
MISC	Any other annotation.

Table 2.3: The list of fields that constitute a word line in the CoNLL-U format.

There are several reasons why `omorfi` was chosen as a morphological analyzer for Finnish, besides that it is a lightweight tool, which is easy to use, and provides UD compatible output format.

First of all, at the time of implementing the new methods and lemmatizing the output, other, more precise and efficient tools had not yet been developed (e.g. `trankit` was developed in 2021 (Nguyen et al. 2021), and the Stanza Python package in 2020 (Qi et al. 2020)).

The performance of these analyzers is somewhat better according to an accuracy evaluation²² of multiple open-source part of speech tagger and lemmatizer algorithms for Finnish, but in fact, the improvement of newer methods is only a couple of percentages (about 4% in case of lemmatization), while the time needed to analyze the same amount of tokens and the computational cost are higher compared to `omorfi`.

Since `omorfi` is a morphological analyzer, and dependency parsing of the obtained sentences was also necessary for the language learning application, another tool was selected to parse sentences. `UralicNLP` (Hämäläinen 2019) is a free and easy-to-use Python library, that provides dependency parsing according to the UD annotation guidelines. This package supports 13 languages in total, e.g. Skolt Saami, Votic, Erzya, Udmurt, and Finnish.

For Hungarian, the same tasks needed to be conducted. Hence, a morphological analyzer and a dependency parser were necessary, which are both parts of the `emtsv` (Indig et al. 2019) language processing pipeline. It is a free, easy-to-use text processing system, which provides a Python library API. `EmtsV` is an attempt to integrate existing tools and unify them in a common, `xtsv` framework for Hungarian. It offers a module (`emCoNLL`) that converts the output into UD-compatible, CoNLL-U format. One of the strengths of `emtsv` is its modularity and compactness.

To summarize, there are many reasons why these tools were chosen to analyze the data sets that were obtained using the methods described in the next section. First of all, all the above-mentioned tools are free and open-source. They are easy to install, and all of them provide sufficient documentation to illustrate the usage of the toolkit. Last but not least, these tools were readily available at the time of conducting lemmatization, morphological analysis, and dependency parsing.

2.5 Methodology

In the next sections, the proposed methods (also described in Ferenczi (2021)) are presented to extract bilingual translation candidates for Finnish and Hungarian.

The main aspects and requirements that were considered before the creation of and in relation to these methods included the following:

1. The results shall be replicable and easy to use,

²²retrieved 26 September, 2022 from <https://github.com/aajanki/finnish-pos-accuracy>

2. the methods shall be open-source, and available for the NLP community,
3. the manually validated translation pairs shall be publicly available, and
4. low computational cost is preferred.

The detailed information of the resources that the following methods were applied to can be found in Table 2.4.

Resource Type	Edition	Version
WordNet	Finnish WordNet	2.0
	Hungarian WordNet	commit b9641132
Wiktionary	Finnish	dump of 21/03/2021
	Hungarian	dump of 21/03/2021
	English	dump of 31/03/2021
OPUS corpus	Finnish–Hungarian word alignments	downloaded 28/03/2021

Table 2.4: Details of the resources that facilitated the generation of translation pairs.

2.5.1 WordNet Connector

The Finnish and Hungarian WordNets are both based on the English WordNet, as mentioned before. In these databases, each synset is identified by an 8-digit long unique number (the synset offset). These identifiers make it possible to find synsets in both language editions which correspond to the same concept. Hence, synset offsets create a bridge between the different language editions of WordNet and make it possible to compile translation candidates. As Varga and Yokoyama (2009: 863) state in their paper: “The internal structure of the multilingual WordNets itself can be a good starting point for bilingual dictionary generation”. Back in 2009, the Hungarian WordNet was still under development, and the Finnish WordNet project only started in 2010.

To link the Finnish and Hungarian WordNets and extract bilingual lexicons, the `WordNet Connector` algorithm has been developed. This script first parses both the Finnish and the Hungarian data sets, then creates a list of translation pairs by combining each element from the two synsets that share a unique identifier based on the Princeton WordNet v3.0 offsets.

Regarding the Finnish data, the `rels/fiwn-transls.tsv` file was used from the freely downloadable ZIP file. This file contains the Finnish equivalents of the English concepts. The tab-separated file contains the relation type (such as `synonym`, `hypernym`, `hyponym`) of the relation

between the English and the Finnish words, but in this algorithm, only synonyms of the English words were used.

To demonstrate the basic idea behind this method, let us consider the two synsets with the unique identifier 07710616 among nouns and match the words in the synsets. In the FinnWordNet there are three lemmata in the synset identified by the offset 07710616 as shown in example (1a). Its Hungarian counterpart consists of two lemmata (see example (1b)). All five words mean ‘potato’: the Finnish word *potaatti* is more informal than *peruna*, and *pottu* is colloquial, in Hungarian, *krumpli* is a vernacular word, while *burgonya* is more formal.

- (1) a. *peruna*, *potaatti*, *pottu*
b. *burgonya*, *krumpli*

These five lemmata then (two from the Hungarian and three from the Finnish synset) are combined by the `WordNet Connector` algorithm (when the `translations` option is provided to the script) and result in the Cartesian product of them, i.e. six translation candidates in total (2x3), as shown in example (2).

- (2) *peruna* - *burgonya* *peruna* - *krumpli*
pottu - *burgonya* *pottu* - *krumpli*
potaatti - *burgonya* *potaatti* - *krumpli*

Both wordnet editions contain multi-word expressions (MWE). These are handled the same way as single words.

The algorithm can also extract the list of Finnish and Hungarian synsets when the `synsets` option is provided to it. This will facilitate the identification of synonyms in both languages. This function outputs the structure that can be seen in Table 2.5. The TSV file contains the part of speech (POS) tag from the UD tagset, the synset offset, the language code (either `fi` or `hu`), and the elements of the synset separated by tab characters.

The `WordNet Connector` method also obtains a list of Hungarian example sentences (with the `examples` option), and a list of definitions (with the `definitions` option), since the Hungarian WordNet contains these kinds of data, as well. The construction of the file that these options produce can be seen in Table 2.6. As it is clear from this sample, the lemma field contains one

POS tag	Synset offset	Language code	Elements of the synset		
NOUN	07710616	fi	peruna	potaatti	pottu
VERB	00069879	hu	megsért	megsebesít	megsebez
ADJ	02071301	fi	erilainen		
NOUN	08556491	hu	földbirtok	birtok	

Table 2.5: Structure of the output file when the `synsets` option is selected.

element of the synset, and the line is repeated (except the content of the lemma field) if there are more than one lemmata in the synset. The same structure is used in the case of example sentences.

POS tag	Synset offset	Lemma	Definition
VERB	00069879	megsért	Sebet ejt, sérülést okoz valakinek.
VERB	00069879	megsebesít	Sebet ejt, sérülést okoz valakinek.
VERB	00069879	megsebez	Sebet ejt, sérülést okoz valakinek.

Table 2.6: Structure of the output file when the `definitions` option is selected.

This algorithm is freely available and downloadable from GitHub²³ and licensed under the GNU AGPL v3.0 License.

2.5.2 Wiktionary Parser

Wiktionary can be used both as a monolingual resource and as a bilingual dictionary. When the language of the headword is different from that of the Wiktionary edition, it works as a bilingual dictionary. Whenever the headword is an element of the target language (the language of the Wiktionary edition), the term is described with a full definition, instead of giving translation equivalents for it. The translations (or definitions) of the headword, as well as the language of the headword, appear in the main body of the entry. For example, if the Finnish word *kiitos* is looked up in the English Wiktionary, the English translation ‘thank you’ is given in this entry under the Finnish header, see Figure 2.5 above.

An algorithm called `Wiktionary Parser` has been developed that iterates over the latest available Finnish and Hungarian Wiktionary dumps that are downloaded automatically. It compiles bilingual translation pairs (including MWE), as well as monolingual lemma–definition and

²³ https://github.com/ferenczizsani/connect_wordnets

lemma–example sentence pairs. Hungarian headwords are extracted from the Finnish, and Finnish headwords from the Hungarian Wiktionary edition. As Wiktionary includes part of speech for most of the headwords, the algorithm also gathers this additional information from the entries and saves it with the translation pairs.

The Finnish Wiktionary edition that was used in this work (the Wiktionary dump of 21/03/2021) contains 416,295 articles (which equals to the number of headwords this Wiktionary incorporates), while the Hungarian one (dump of 21/03/2021 as well) contains 369,292.

The Wiktionary templates and the layout of the entries vary depending on the edition. Hence, the Finnish and Hungarian dumps need to be parsed differently, and two separate functions were created for this purpose (`extract_fi_dict()` and `extract_hu_dict()`). The elements of the part of speech tagset had to be closely examined and collected for both languages because they do not use a unified, standard set, but tags that are written in the language of the Wiktionary (such as *erisnimi* ‘proper noun’ in Finnish, or *ige* ‘verb’ in Hungarian). The `wikpos2ud()` function normalizes this information and produces the part of speech tag compatible with the UD standard. Further normalization and cleaning procedures had to be conducted in order to parse and extract the desired information from the Wiktionary dumps. Removing unnecessary characters and HTML tags present in the fields is done by the `clean_line()` function.

The output is a TSV file, which contains the Finnish and Hungarian lemmata (translation candidates), and the part of speech tag (using the standard UD tagset) separated by a tab character in case of translation extraction. When obtaining definitions and example sentences, the produced output contains the part of speech tag of the headword, the headword, and the sentence (definition or example sentence). The two kinds of relations (lemma–example sentence and lemma–definition relations) are saved in separate files.

The dump versions that were used to obtain the translation candidates and additional data are given in Table 2.4.

`Wiktionary Parser` is a free, open-source tool²⁴ that is licensed under the GNU AGPL v3.0 License.

²⁴https://github.com/ferenczizsani/wiktionary_parser

2.5.3 OPUS Extractor

The Finnish–Hungarian word alignments available in OPUS corpus serve as the base of the `OPUS Extractor` tool, the third method developed in this project. This script downloads and extracts translation candidates from the lists containing bilingual word alignments. After manually validating and analyzing a small subset of the resulting translations, it was observed that the output contained many erroneous translation candidates. The validation of a randomly selected sample, 400 translation candidates in total, showed that the precision of the raw word alignments is very low. When the “correct translations” were defined as two lemmata being perfect translation equivalents of each other, the precision was below 10% (6.25%). Including the named entities (proper nouns) which are not necessarily part of these languages did not achieve better results, the precision still barely reached 25%. It was found that these faults could be easily identified and filtered out. In order to improve the quality of the translation pairs, some heuristics and conditions had been defined and applied, which are described below.

A constraint that is applied to the bilingual word list is that translation pairs are removed if any of the participating words or phrases contain any characters that are not included in the alphabet of that language.

The `OPUS Extractor` algorithm downloads the available word alignments from OPUS, filters the lines according to the constraint that was previously mentioned, and outputs the resulting translation candidates. The result of the algorithm is a TSV file that contains the Finnish and Hungarian single words. There are no MWE in this data set since the word alignment method is applied to single word units only. The values of the resulting file are separated by tab characters, and the data is either in alphabetical order or arranged in descending order by the number of co-occurrences in the OPUS dictionaries. For this latter to happen, a second parameter has to be passed to the Bash script (after the output folder), and its value is either `true` or `1`.

After further examination of the translation candidates, there were still many erroneous pairs because the word alignments were generated from running texts. As also observed by Simon and Mittelholz (2017); Ferenczi et al. (2018), the parallel corpus does not provide high-quality translation pairs in the case of morphologically complex languages. To illustrate this issue, Table 2.7 contains an excerpt of the translation candidates before any text-processing method was applied.

Out of these 8 – sometimes incorrectly aligned – word pairs, none is in fact the lemmata of the expected two translation pairs, consisting of the nominative singular forms of the Finnish lemma (*poika* ‘boy, son’) and the Hungarian lemmata (*fiú, fia* ‘boy, son’), which would be included in the dictionary as headwords and translation equivalents of each other. To attain higher quality output, lemmatization and morphological analysis must have been done on the Finnish words with the help of the `omorfi` tool, and on the Hungarian words with `emtsv`. More details about these NLP algorithms can be found in Section 2.4.

	Finnish	Hungarian
1	Poika ‘Son-NOM.SG, Boy-NOM.SG’	A ‘The’
2	Poikaa ‘Son-PTV.SG’, ‘Boy-PTV.SG’	a ‘the’
3	Poikaa	Fia ‘Son-NOM.SG’
4	poikaa ‘son-PTV.SG’, ‘boy-PTV.SG’	fia ‘son-NOM.SG’
5	poikaa	fiat ‘son-ACC.SG’
6	poikaa	fiát ‘his.son-ACC.SG’
7	poikaa	fiú ‘boy-NOM.SG’
8	poikaa	fiút ‘boy-ACC.SG’

Table 2.7: Excerpt from the output of OPUS translation candidates before lemmatization.

In this example, it is also possible to observe that the OPUS word alignments contain upper-case and lower-case words as distinct entries, capitalized when the word appears as the first word in the sentence. This results in duplicate entries where the only difference is whether the word is capitalized or not (see rows 3 and 4 in Table 2.7). The lemmatizers tackle this issue since the root of words are lower-cased whenever the part of speech tag is other than a proper noun. There were many words to which the lemmatizers and morphological analyzers assigned more than one lemmata (sometimes belonging to different part of speech categories) due to, for example, homonymy. In these cases, all information is kept at text processing, and when creating the processed list that

contains the lemmatized elements of the original translation candidates, only those lemmata and part of speech information are paired together which have a part of speech category in common.

Before lemmatization, there were more than one million unique translation pairs (1,022,945). Unique, in this case, means that each Finnish–Hungarian translation candidate appears only once in the list of translations, hence, there are no duplicate entries when observed on the character level. Applying lemmatization showed a 73.17% decrease, resulting in 274,430 translation candidates, which shows that the data included many forms of the same root in both languages.

The script that extracts translation candidates from the OPUS word alignments, and applies conditional filtering in order to clean the data, is freely available on GitHub²⁵ and licensed under the GNU AGPL v3.0 License.

2.5.4 wikt2dict

The performance of the methods developed in this research shall be put into perspective. To do that, the `wikt2dict` tool was used to compile bilingual dictionaries for the same language pair and its results have been also manually validated following the same instructions. This tool was developed by Ács et al. (2013).

Although this parser was not specifically built for the Finnish–Hungarian language pair, it is still possible to collect translation candidates with the help of this algorithm. It does not use headwords and their translations from the main body of the Wiktionary entry to extract translations, but rather the information found in the translation tables that are included in only a subset of the Wiktionary entries. It has two different functionalities.

When given a source and a target language code, the first function (which the developers call `extract`) searches for target language words in the translation tables in the source language Wiktionary dump, making a possible translation pair from the headword and its equivalent in the other language found in the table. To extract Finnish and Hungarian bilingual translation candidates, this function was applied to the Finnish and Hungarian Wiktionary editions.

The other function (which is called `triangulate`) requires three different language codes and uses the third language as a pivot to create connections between source and target language words.

²⁵https://github.com/ferenczizsani/opus_extractor

This assumption is based on the fact that translation is a transitive relation between words. If A is a translation equivalent of B and B is a translation equivalent of C, then supposedly, A is also a translation equivalent of C. However, this assumption must be treated with caution: when the pivot word (here B) is polysemous or there is another word which is homonymous with the pivot word, triangulating can result in erroneous translation pairs, as mentioned in Section 2.1.3. Figure 2.7 shows two translation tables (collapsed) from the English Wiktionary, where the headword (‘letter’) has two meanings. Wiktionary handles polysemy and homonymy in such a way, that each sense appears in a separate translation table, and in case these rules are respected, the correct translation pairs can be extracted. The list of correct translations from these translation tables are shown in example (3).

Translations [edit]	
±	a symbol in an alphabet - Finnish: kirjain ^(fi) , aakkonen ^(fi) ; Hungarian: betű ^(hu)
±	written message - Finnish: kirje ^(fi) ; Hungarian: levél ^(hu)

Figure 2.7: Triangulation of Finnish and Hungarian words through English as a pivot. Translation tables from the English Wiktionary entry ‘letter’.

- (3) a. *kirjain* = *betű*
 b. *aakkonen* = *betű*
 c. *kirje* = *levél*

However, the number of potential translation pairs that the triangulating function of `wikt2dict` extracts is greater, and, as will be shown in the following section, is less precise. The translation candidates that are extracted by this method can be seen in example (4), resulting in 50% precision (examples (4c), (4d), and (4e) are incorrectly induced).

- (4) a. *kirjain* = *betű*
 b. *aakkonen* = *betű*
 c. *kirje* = *betű*
 d. *kirjain* = *levél*

- e. *aakkonen = levél*
- f. *kirje = levél*

Since the English language edition of Wiktionary contains the largest number of entries, it was chosen as the pivot language to extract additional translation pairs for Finnish and Hungarian.

2.6 Evaluation

In this section, the intermediate results are described. First, the details of data extraction are provided along with some major observations about data (Section 2.6.1). Then, the results and lessons learned during validation are presented in Section 2.6.2.

Due to time constraints and the large amount of data compiled from different resources, the complete data set could not be analyzed and evaluated manually to date. Therefore, a representative sample of the data set has been validated by speakers of Finnish and Hungarian (which also include the author).

2.6.1 Details of Data Extraction

The results of data collection with the above-mentioned methods can be seen in Table 2.8. Along with the method name, the type of data is indicated that was collected with the help of that method, and the number of instances (translation candidates, synonym pairs, lemma–definition and lemma–example sentence pairs) it extracted in total. The synonym pairs are combined from the output produced by the `synsets` option of the `WordNet Connector` method. To create synonym pairs, a list of unordered subsets of 2 elements from the synset is generated. This is illustrated in example (5) with the synset which contains the Finnish lemmata *peruna*, *potaatti*, *pottu* ‘potato’.

- (5)
- a. *peruna - potaatti*
 - b. *peruna - pottu*
 - c. *potaatti - pottu*

The biggest number of translation pairs were extracted by the `wikt2dict triangulate` method, followed by the translation pairs obtained from the OPUS corpus word alignments. The

Method	Type of Data	# of Instances
WordNet Connector	translation pairs	98,883
Wiktionary Parser	translation pairs	9,544
OPUS Extractor	translation pairs	274,430
wikt2dict extract	translation pairs	12,731
wikt2dict triangulate	translation pairs	294,757
WordNet Connector	synonym pairs (Hungarian)	28,196
WordNet Connector	synonym pairs (Finnish)	54,534
WordNet Connector	definitions (Hungarian)	42,365
Wiktionary Parser	definitions (Hungarian)	30,423
Wiktionary Parser	definitions (Finnish)	111,555
WordNet Connector	example sentences (Hungarian)	36,484
Wiktionary Parser	example sentences (Hungarian)	1,157
Wiktionary Parser	example sentences (Finnish)	29,221

Table 2.8: Results of data collection.

least amount of bilingual word pairs was gathered by the `Wiktionary Parser` algorithm, which collected Hungarian words from the Finnish, and Finnish words from the Hungarian Wiktionary.

Although there is a very big difference between these numbers (the least number of translation pairs is 30 times smaller than the biggest collection), it is important to note that their quality might not be directly proportional to the number of extracted word pairs. This big difference can be explained by the size of the resource that is utilized by the method and the applied technique. Pivot language based methods connect data sets with English as a pivot, and English data sets are always greater in size. On the other hand, the word alignments are automatically generated in OPUS using big amounts of parallel texts, and this results in inflected word forms repeated several times throughout the extracted data, as shown in Table 2.7, containing lots of incorrect translation candidates.

The difference between the synonym pairs from HuWN and the FinnWordNet is anticipated since the Finnish data set contains almost three times more words and synsets (see Tables 2.1 and 2.2).

Another interesting observation that can be made regarding the results of data collection is that the Finnish Wiktionary edition seems to contain way more entries with definitions and example sentences than its Hungarian counterpart.

To determine how many correct translation pairs and other types of relations each method results in, their precision is estimated by human evaluation. To facilitate this work, a dictionary writing system (DWS) was created, which is described in more detail in Section 2.6.2.

Since the same language pair was used when applying each lexicon building method, there exists some expected overlap between the vocabulary they cover. The number of common translation candidates in the intersection of each pair of methods is presented in Figure 2.8.

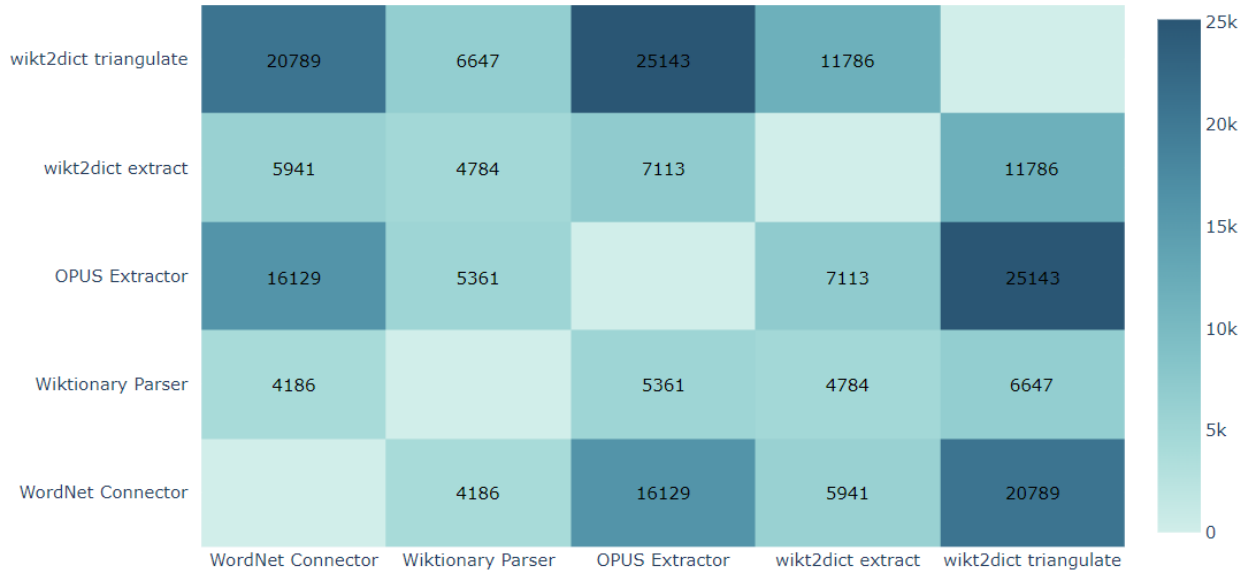


Figure 2.8: Number of common translation pairs between methods.

For the sake of simplicity, the values in the diagonal of the matrix are hidden, since these cells would reflect the total number of translation candidates obtained by each method. The missing values can be found in Table 2.8, where the number of extracted instances was presented.

Although two methods (`OPUS Extractor` and `wikt2dict triangulate`) resulted in numerous translation candidates (more than 250,000 in both cases), the overlap between these methods and the rest of the (smaller) methods is not enormous. It can be due to the nature and precision of the methods, as mentioned earlier. They might not coincide with the word pairs obtained by other methods, caused by their quality. By contrast, the overlap between `Wiktionary Parser`, and `wikt2dict extract` reaches more than half of the extracted word pairs from `Wiktionary Parser`.

The next step is to validate the translation candidates, synonym pairs, definitions, and example sentences. This evaluation will make it possible to examine which method provides the most pre-

cise results. The manual validation of word pairs was done by a small number of volunteers who are speakers of both Finnish and Hungarian, including the author. It was done on an online user interface created for this purpose. Detailed instructions are provided on this website about how the translation, synonym, example sentence and definition relations should be evaluated, when they should be deleted, and when they should be marked as correct.

2.6.2 Manual Evaluation in the Dictionary Writing System

To manually validate the automatically extracted data, an online user interface (UI) was developed that also functions as a DWS for the Finnish–Hungarian–Finnish online bilingual dictionary. This dictionary is also part of the present research work and will be described in Section 5.3 in more detail.

The extracted data was imported into a MariaDB-based relational database that stores lexical information in a non-redundant way, using a language-independent model which will be described in Chapter 3. Since the structure of this underlying database is quite complex, and it is not expected from the human evaluators to be able to work with SQL statements, a user-friendly platform was developed. Furthermore, this interface enables secure communication between users and the database, while avoiding the majority of security threats and keeping the contents of the database consistent.

This DWS allows authorized users to mark translation and synonym candidates, as well as lemma–definition and lemma–example sentence pairs as correct or incorrect. A guideline containing detailed instructions (which is available from the platform) helps the evaluators to validate the automatically attained data in accordance with the predetermined requirements.

A translation is considered correct when the two words are perfect translations of each other (Des Tombe 1992). This ensures the automatic reversibility of the dictionary. To mark a (translation or other) relation as correct, the first step is to make sure that both of the entities in the relation are existing elements of the respective languages (e.g. both lemmata participating in the translation pair). During the validation of lemmata and other entities, it was found that the type of data greatly impacts precision in the case of both languages. As will be shown in Section 3.4.1, the extraction did not only lead to lemmata and sentences, but also to word forms, affixes, and MWE. For the sake of simplicity, all data types – except sentences – henceforth are collectively referred to as

“non-sentences”. It is clear from the intermediate results of the first step of data validation that the precision of sentences and that of non-sentences differ. As it was observed, only a very few entities belonging to the non-sentence group were rejected and marked as incorrect, while the extraction of sentences led to more incorrect entities in the database, as can be seen in Table 2.9. If only non-sentences are considered in the evaluation (lemmata, word forms, etc.), the reached precision is higher than 99% for Hungarian as well as for Finnish.

Language	Type of Data	# of Validated Data	Precision
Hungarian	non-sentences	613	99.837%
Hungarian	sentences	854	91.452%
Hungarian	all	1467	94.956%
Finnish	non-sentences	520	99.808%
Finnish	sentences	425	92.941%
Finnish	all	945	96.719%

Table 2.9: Intermediate data validation results.

It effectively means that in the validated data set, only a very few number of non-sentences were incorrect, i.e. not part of the Hungarian or Finnish vocabulary. Examples of incorrect lemmata include *the* as a Hungarian and *whether* as a Finnish lemma. It is perhaps important to note that every non-sentence that was deleted had been extracted by the `OPUS Extractor` algorithm.

Sentences were marked as incorrect due to many reasons. However, there were some observable patterns and a couple of examples can be seen in example (6) to demonstrate the prototypical problems. Many times, Wiktionary contains mathematical formulae which use a subset of `TeX` markup (see example (6a)).

The rest of the examples (example (6b), (6c), and (6d)) typically occur when the headword is a country name; they give more information about the country (the country code, its abbreviation code used by the International Olympic Committee, and its vehicle registration code). These are not necessarily the definitions of the headword, so the validators were asked to delete sentences that might resemble this pattern. This kind of data also originates from Wiktionary, they were extracted by the `Wiktionary Parser` algorithm that parses the definitions from the main body of the entries.

- (6) a. $\lambda \cdot \underline{a} \in H$ (which is rendered as $\lambda \cdot \underline{a} \in H$)
- b. Telefon előhívó szám: 1787
- c. Olimpiai rövidítés: PAN
- d. Gépkocsi felségjel: ROU

Following the validation of lemmata and sentences, the relations between them need to be checked and corrected if necessary. In case two lemmata, or a lemma and a sentence are connected incorrectly, the relation shall be deleted.

Due to the large amounts of data collected from different resources (as can be seen in Table 2.8), and the given time constraints, the evaluation has been conducted on a representative sample of the gathered data. The relations which compose the subset were selected using a random sampling technique, including at least 200 relations for each method and type of relation (translation pairs, synonyms, definitions, or example sentences).

The precision of the methods can be determined by the number of correct relations divided by the total number of evaluated relations compiled by that method. For each method, the exact number of evaluated relations (whether translation or synonym pairs), and the precision associated with them are shown in Table 2.10.

Although the initial expectation was to validate a minimum of 200 relations per method, the common, overlapping translation candidates between algorithms resulted in more validated candidates than the predefined, randomly selected subset. In other words, when a translation candidate is evaluated in relation to a certain method, but it is also among the candidates of the output of another method, this candidate helps determine the precision of both methods. For example, there are 881 translation candidates that were validated in the case of the `WordNet Connector` algorithm, and 408 in the case of the `OPUS Extractor`.

The results of the translation candidate evaluation are visualized with the help of a bar chart which can be seen in Figure 2.9. To answer one of the research questions outlined in Section 1.5 (RQ 1.), it can be observed in both Table 2.10 and Figure 2.9 that the highest performing method is `Wiktionary Parser`, which was created during this research work. The method that reaches the second highest precision was a function of a tool developed by Ács et al. (2013) (`wikt2dict extract`). Both of these methods extract data from Wiktionary, which is a dictionary project

Method	# of Validated Relations	Precision	Expected
WordNet Connector	881	72.645%	71,834
Wiktionary Parser	261	98.851%	9,434
OPUS Extractor	408	93.137%	255,596
wikt2dict extract	258	98.062%	12,484
wikt2dict triangulate	617	72.609%	214,020
WordNet Connector (Finnish synonyms)	279	72.043%	40,164
WordNet Connector (Hungarian synonyms)	307	69.707%	19,903

Table 2.10: Precision and expected number of correct translations and synonyms for each method. In column ‘Expected’ the number of relations appears that is expected to be correct according to the manual validation and the total number of obtained data.

with many volunteering contributors. According to the results, Wiktionary proves to be an online dictionary with reliable data.

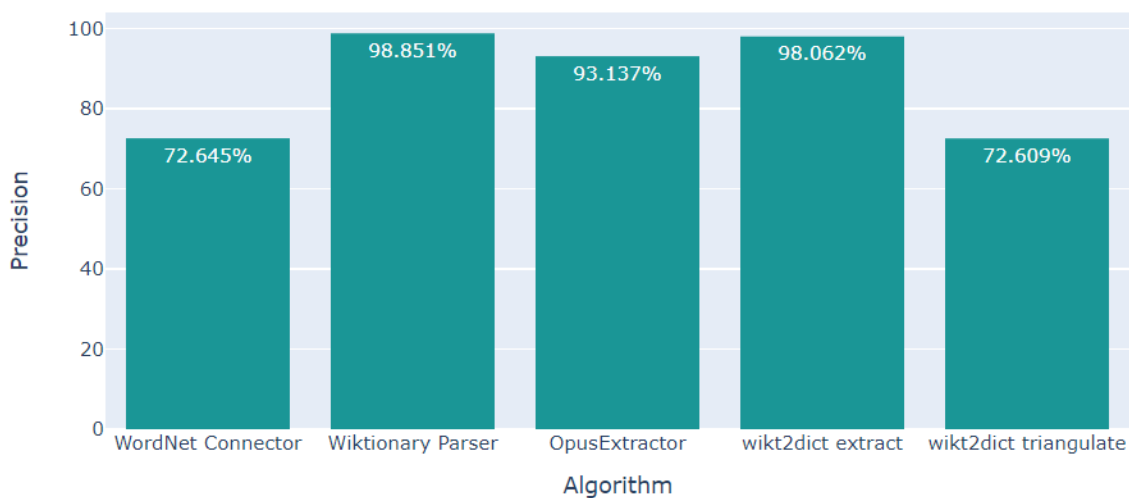


Figure 2.9: Visualization of the evaluation of translation relations.

In the case of the third algorithm that is based on Wiktionary (`wikt2dict triangulate`), it was found that the erroneous translation pairs must come from the way the parsing tool processes Wiktionary translation tables. When the details of the source code of this tool were manually verified, it was encountered that the different word senses belonging to a certain headword were not considered to have separate meanings. If a headword has more than one sense, its Wiktionary entry contains several translation tables (as already presented in Figure 2.7), one for each meaning, and

these tables are not handled separately by the algorithm. In this case, words with completely different senses will be incorrectly marked as translation candidates (cf. examples (3) and (4) above). The triangulating algorithm should respect the different translation tables, and handle the distinct senses separately. This (perhaps intentional) flaw in the method results in the lowest quality regarding the precision of the translation candidates.

Among the newly proposed methods, the least accurate translation pairs were obtained by the `WordNet Connector` algorithm. Although the utilized wordnets were manually constructed and evaluated by professional translators, the assumption to create equivalents by the Cartesian product of the two synsets (the Finnish and its Hungarian equivalent connected by their offsets) does not seem to always result in correct translation pairs. For example, in the case of synsets which refer to some real-world entities or individuals, comparing the words in the two synsets and only generating pairs that are literal equivalents might improve the precision of the results. This issue can be demonstrated by the following example: the English synset `<James Joyce, Joyce, James Augustine Aloysius Joyce>` is unmodified in Finnish and translated into Hungarian as `<James Joyce, Joyce>`, removing the last element of the original synset. These synsets are connected by their offset and the combination of all possible pairs is extracted with the current version of `WordNet Connector` (in this case, 6 pairs in total). Out of these pairs, only two are valid according to the guidelines of the validation process (the ones that are literal equivalents), leading to a 33.33% precision. Synset pairs like this one worsen the overall precision of the method.

The translation candidates extracted from the OPUS corpus have lower precision than the two best-performing methods, however, a more than 20% gain is obtained with its help when compared to `WordNet Connector` and `wikt2dict triangulate`. Furthermore, the quality of the output of this method after lemmatization greatly exceeds that of the initial evaluation, where 400 translation pairs were validated and less than 25% of the translations were correct (see Section 2.5.3). This result provides a direct answer to one of the research questions (RQ 2.) formulated in Section 1.5, and confirms that lemmatization can improve the precision of Finnish–Hungarian translation pairs that are to be included in the contents of a dictionary.

When examining the precision of the extracted synonym pairs in Finnish as well as in Hungarian, it can be observed that the quality of these relations are very similar to that of the translation

candidates obtained from the same resource (see Table 2.10). As a consequence, the reason for the poor quality of the results of this method seems to lie in the flexibility of how synonyms are defined in WordNet. A synonym set in WordNet may be a wider and less strict concept than what a synonym or translation relation is considered to be during the validation process of this work. Many times, synsets contain several words, as shown in Section 2.3.1. One of the synsets that contains numerous lemmata is the one with offset 01806505. In the English WordNet, there are 13 synonyms listed that are closely related to this concept, defined as “attract; cause to be enamored”. In the Finnish version, 11 verbs, while in the Hungarian, 8 verbs are present in the synset with the same offset. This leads to 88 translation pairs, and 55 as well as 28 synonym pairs (Finnish and Hungarian, respectively). Among these, there are many pairs that are not perfect translations or synonyms of each other, leading to a decrease in precision.

Apart from translation candidates and synonyms, a subset of definitions and example sentences was also evaluated. The number of validated relations, as well as the expected number of correct relations, are shown in Table 2.11 for each method. The precision can be seen in Figure 2.10. Only two methods could obtain this kind of additional information since in OPUS corpus, there are no definitions or example sentences, while the `wikt2dict` algorithm was not designed to collect them from Wiktionary.

Method	# of Validated Relations	Expected
WordNet Connector (Hungarian definitions)	213	40,973
WordNet Connector (Hungarian examples)	241	20,437
Wiktionary Parser (Hungarian definitions)	234	24,442
Wiktionary Parser (Hungarian examples)	200	856
Wiktionary Parser (Finnish definitions)	211	98,926
Wiktionary Parser (Finnish examples)	211	25,620

Table 2.11: Precision and expected number of correct definitions and example sentences for each method. In column ‘Expected’ the number of relations appears that is expected to be correct according to the manual validation and the total number of obtained data.

`WordNet Connector` collected definitions and example sentences from the Hungarian WordNet, as the Finnish resource does not contain such information. The precision of the definitions proves to be of way higher quality than that of example sentences. The reason for this poor performance seems to be due to a minor misconception in the extracting algorithm and the con-

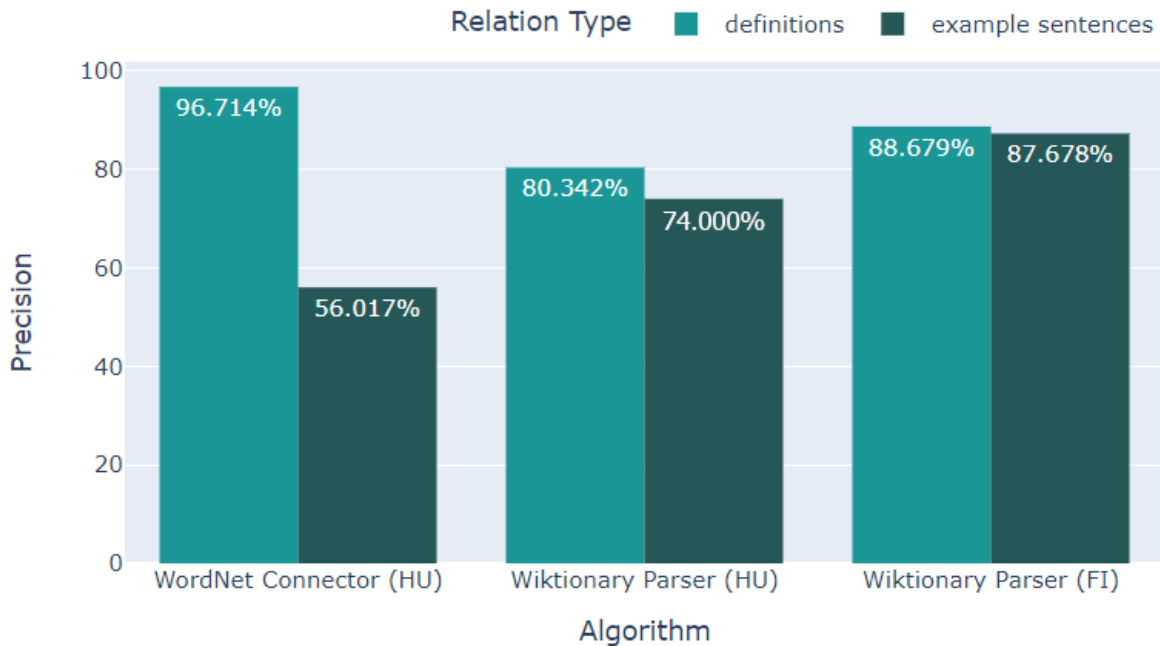


Figure 2.10: Precision of methods regarding the definition and example sentence relations.

struction of the Hungarian WordNet. Most of the synsets have one and only one definition and example sentence describing their meaning and illustrating their usage. However, when a synonym set contains more than one lemmata, the example sentence, that is supposed to illustrate the usage of the synset, contains only one of the lemmata found in that synset. The example sentence extraction connects this single example sentence to all elements of the synset, which leads to at most one correct relation and hence, a decreased precision.

Wiktionary Parser obtained definitions and example sentences for both languages. Comparing the results of the two languages, it can be noted that the Hungarian data is somewhat less precise than the results obtained by the Finnish Wiktionary, and definitions in general tend to have better quality than example sentences. However, there is no significant difference between the precision of the different result sets of this algorithm, which would be reminiscent of the difference produced by the WordNet Connector tool.

The manually validated data sets are freely available on GitHub²⁶ under the Creative Commons Attribution-ShareAlike 4.0 International License.

²⁶ https://github.com/ferenczizsani/fin_hun_resources

2.7 Discussion

The algorithms that have been used to collect bilingual translation pairs and additional lexical information (definitions, example sentences, and synonyms) have been evaluated with the help of human validators, and volunteers who speak both Finnish and Hungarian, including the author on an interface which facilitates communication with the database.

The DWS that has been developed in this project served as the interface for the manual validation of the extracted data. It is also possible to manipulate and edit the entries of the online dictionary on this same platform (for more details, see Section 5.2).

The two methods with the highest precision according to the validated data are `Wiktionary Parser` and `wikt2dict extract`. These methods parse the manually edited, crowd-sourced Wiktionary. The best-performing methods prove that Wiktionary provides high-quality translations when they are obtained from headwords and translation tables directly. This observation is in contrary to the common belief (confirmed by the survey in Gaál (2016)) that crowd-sourced dictionary projects are unreliable. This finding responds to research question 1, which aimed to determine which dictionary building method performs best for Finnish and Hungarian. Nevertheless, it is worth mentioning that the highest precision was reached by methods that generated the least amount of translation candidates. It can be explained by the small amount of Finnish–Hungarian bilingual data present in the different Wiktionary editions.

The translation candidates of the `OPUS Extractor` method reached the third highest precision. Apart from the previously mentioned methods and resources, Simon et al. (2015) also used OPUS to extract word pairs for under-resourced (Finno-Ugric) languages. During the evaluation phase of the project (which is described in Ferenczi et al. (2018)), the precision of the word alignments taken from the OPUS corpus was very low (27.57%). The improvement presented in this chapter may be therefore attributed to the lemmatization of the word alignments, since the languages examined by Ferenczi et al. are also morphologically complex languages, and lemmatization was not conducted on the word pairs obtained from OPUS before validating them. This finding answers one of the research questions (RQ 2., repeated below), and confirms that lemmatization and natural language processing approaches can improve the precision of dictionary building methods.

RQ 2. Does language processing techniques, such as lemmatization, affect the results and help

reach a better precision?

The manual correction of the rest of the automatically extracted translation candidates, synonym pairs, definitions and example sentences needs to be continued, before including them in the public interface of the Finnish–Hungarian–Finnish online dictionary. The manual post-editing and validation provide a high-quality, reliable resource that can be used by language learners.

The requirements mentioned in Section 2.5 are repeated here for further discussion.

1. The results shall be replicable and easy to use,
2. the methods shall be open-source, and available for the NLP community,
3. the manually validated translation pairs shall be publicly available, and
4. low computational cost is preferred.

The open-source code is well-documented and available on GitHub (for a summary, see Appendix B), and the precise version of the data sources is mentioned in the description of each algorithm throughout this chapter. Therefore, the results can be replicated.

The validated translation pairs, as well as lists containing other data, can be downloaded from the GitHub repository²⁷, apart from being accessible on the online dictionary interface (described in Section 5.3).

The execution of these methods did not take longer than a couple of minutes, hence, the low computational cost is also fulfilled. There was no need for long hours of training, nor GPU for matrix multiplications since the methods applied in this work use alternative solutions to obtain data.

The lexicographic database and data model where the obtained translation pairs and additional data can be stored will be presented in the next chapter.

²⁷ https://github.com/ferenczizsani/fin_hun_resources

3 Lexicographic Database

In this chapter, a more practical, technical outcome of the present research will be emphasized and presented. The decisions and considerations that needed to be made when designing this database were, however, based on theoretical and conceptual principles. As Jackson (2002: 161) highlights it: “No dictionary can begin to be compiled without considerable forethought and planning.”

The chapter investigates and attempts to answer one of the research questions of this work (RQ 3., see Section 1.5). The first section clarifies the definitions which usually appear within the field of electronic lexicography and which lexicographers tend to confuse the most. Then, Section 3.2 explains why a relational database has been chosen over all the traditional ways dictionaries are created. The main idea behind the data model is presented. Section 3.3 summarizes the structure of the database (the database schema) that has been created in order to contain all the extracted data. In Section 3.4, the details of the database tables are given, along with some example records and the contents of some prepopulated tables. Section 3.5 demonstrates how the extracted data have been used to populate the database tables, and Section 3.6 and 3.7 present views and triggers that have been created to facilitate the work with the database. Section 3.8 provides examples to illustrate the process of how to obtain information from the different tables.

3.1 Definitions

Bergenholtz and Nielsen (2013) explains that many lexicographical discussions are misleading and often use the terms ‘*dictionary*’ and ‘*database*’ interchangeably. It is important to note that there is a clear difference between these concepts. Fuertes-Olivera and Tarp (2014: 64) describe the distinction between a dictionary and a database as:

Electronic dictionaries are not databases, but consultation tools based upon databases from which they take in the data required to meet their users’ information needs.

The aim of the database is to determine how the data (which forms the material of the dictionary) will be structured and to store the data in such way. It is a “structured collection of values”, as Bergenholtz and Nielsen (2013) define it. The structure of the database is usually referred to as ‘*database schema*’. It determines the way data can be stored in the database, which pieces of

information are mandatory, which are optional and what data type is allowed in certain fields. A dictionary, on the other hand, is an outcome, a product, resulting from a predetermined way of selecting data from the database. The database is hence independent of the dictionary (or dictionaries) that is created from it, and also independent of the user interface (UI) that the user interacts with. There can be more than one UI based on the same database: one for lexicographers to access and edit the entries, and multiple UI which can serve as the connection point between the dictionary (with a given use case and purpose) and its end users.

3.2 Data Representation

The collected data described in Chapter 2 need to be saved and stored in a way that can facilitate the creation of an online bilingual dictionary, whether it is used as an encoding or as a decoding dictionary, for both Finnish, and Hungarian native speakers. In this research, it is attempted to create an automatically reversible Finnish–Hungarian–Finnish dictionary from automatically attained translation pairs and additional lexical data.

There exist several ways to store data. Data storage can be provided by different file formats, such as XML or JSON. However, using a relational database provides a more extensive and complex way of representing data. Relational databases are often supported by database management systems, which allow the users of the database to manipulate and maintain the data easily and effectively. A Structured Query Language (SQL) is used to perform operations on and manipulate the data in the database. Inserting new records into, retrieving data from, updating information in the database or even creating the database schema is possible with the help of this language.

As Tavast et al. (2018) mentions, there are certain disadvantages when it comes to XML databases in comparison to relational databases. Besides its performance, the incapability of simultaneous access by multiple users for editing is one of the reasons why in this research, the use of a relational database is preferred. Alnajjar et al. (2020) also mentions this benefit of relational databases when several lexicographers try to compose and edit a dictionary in an online environment. Another limitation of the XML data structure is that it does not allow the easy reversal of bilingual dictionaries, neither does it facilitate the linking of MWE to several entries (Mechura 2016).

One advantage of the XML format is that it is a standardized way of creating electronic dictionary-

ies, which would promote interoperability with other resources and tools. Nevertheless, converting a relational database into a well-formed XML document is an existing technique.

Due to these considerations, the extracted data are saved in a MariaDB-based relational database. As a result of the database structure, the information is stored in a language-independent, non-redundant way. The database schema that takes advantage of the benefits mentioned above is described in Section 3.4.

With the help of this database, several dictionaries can be built and designed, depending on the needs and requirements of the end users. The interface and the contents of the dictionary can easily adapt to different needs, since the information present in the database is not restricted or limited in any way.

The main requirement for the database in this research is to allow the storage of any kind of lexicographical data in a universal way, making it possible to add new pieces of information or new languages at any time in the future.

3.3 Data Model

One of the basic concepts of the proposed data model is `entity`. Lexical items and even combinations of lexical items (full sentences, MWE) are considered to be entities. This kind of information is stored in the `Entity` table (described in more detail in Section 3.4.1). This conceptual feature allows data handling to be general and universal. Entities have certain information associated with them, for example, unique identifiers, the type of data they represent (`lemma`, `mwe`, `sentence`, etc.), their frequency, among others.

Entities can have certain relations with other entities. Such a relation is described by the identifiers of the two entities participating in it and the type of the relation. This is stored in the `Relation` table (for more details, see Section 3.4.2). For example, two lemmata in the same language can be synonyms of each other, two entities (in different languages) can be translation equivalents of each other, or a lemma can be connected to an example sentence that demonstrates the way the lemma is used. The types of relations are stored in the `RelationType` table. The contents of this table can be found in Table 3.8.

The database stores all the information that has been extracted with the methods presented in Section 2, and it is used for the language learning application that is described in Section 5.4.

Since the structure of the database facilitates the universal and generic storing of data, it can serve as the base for both bilingual and monolingual dictionaries. The automatic reversibility of the bilingual dictionaries is a consequence of the database schema, and that the relation between entities is considered to be symmetric, since only perfect translations are kept in the manual validation phase.

3.4 Database Schema

The simplified database schema is depicted in Figure 3.1 in an Entity Relationship Diagram. The most important tables and the relations between them can be seen in this diagram. The `Relation` and `Entity` tables are connected, since a relation is composed of two entities. The `SourceType` table contains the methods described in Section 2.5 in order to track the source of translation candidates, synonym pairs, definitions and example sentences. This enables the editors of the dictionary to see detailed information about the automatic dictionary building methods and their performance. The `Users` table stores details about the users of the dictionary. The editors of the DWS can make comments on entities or relations, which are stored in the `Remark` table.

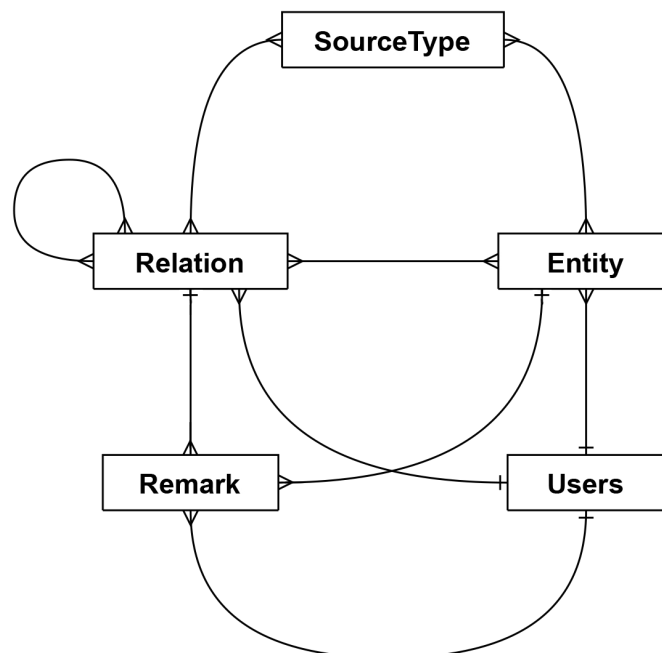


Figure 3.1: Simplified data model of the database.

The full database schema proposed in this research can be seen in Appendix A. In order to

successfully describe the nuances in this diagram and the details of the database schema, a few database concepts must be clarified.

There are 11 tables in the full Entity Relationship Diagram (see Appendix A), each represented by a rectangle. The name of the table can be seen in the header of the rectangle, and its set of fields below the table name. Next to certain fields the abbreviations PK and FK can be seen.

A primary key (PK) can be composed of one or more fields, which can uniquely identify the different records in a database table. Therefore, it must contain unique values (for example, an integer that automatically increases throughout the data set) and it cannot contain NULL (empty) values. A field in a database table can have a foreign key (FK) constraint, which means it can only contain a value that is present as primary key in another table (which is called *referenced table*). There is an arrow in the diagram going from the FK of a table to the referenced table's PK. Foreign key constraints therefore prevent the insertion of invalid data by only accepting values that match the primary key of the referenced table.

There are several data types that a field can be associated with in a table. Table 3.1 gives a short summary of the data types that were most commonly used in the database proposed in this work.

Data type	Description	Example
INT	a number without fractions	45
FLOAT	a single precision floating point number	6.8
VARCHAR	non-binary strings with variable size	kutya
TEXT	fixed length strings (up to 64 kB)	kutya
MEDIUMTEXT	fixed length strings (up to 16 MB)	kutya
BOOLEAN	0 or 1 (meaning false or true)	0
DATETIME	date and time	2022-01-23 12:34:56

Table 3.1: Some of the most common data types used in the database.

The most commonly used primary key type in this database is an integer field with the `AUTO_INCREMENT` attribute, which means that the integer in that field automatically increases each time a new record is inserted in the table, starting at 1.

In the following sections, the different tables of the proposed database will be described and illustrated with data already present in the tables.

3.4.1 Entity

The table that holds all information about all kinds of lexical data is the `Entity` table. It contains 15 fields in total that keep and manage details about each entity. The structure of the `Entity` table can be seen in Table 3.2.

Name of field	Type	Description	Example
<code>entity_id</code>	INT	auto_increment, primary key	64683
<code>wn_offset</code>	VARCHAR	semicolon separated list of wordnet offsets	09886220; 02084071
<code>text</code>	MEDIUMTEXT		kutya
<code>type_of_entity</code>	VARCHAR		lemma
<code>lang</code>	INT	foreign key (Languages)	2
<code>lemma</code>	VARCHAR		
<code>upos</code>	INT	foreign key (PartOfSpeech), part of speech tag (UD)	9
<code>ufeat</code>	VARCHAR	morphological features (UD)	
<code>label</code>	INT	foreign key (Label)	1
<code>frequency</code>	FLOAT	frequency per million tokens	97.1215
<code>inflection_type</code>	INT	foreign key (Inflection)	1
<code>inflection_remark</code>	VARCHAR		
<code>status</code>	VARCHAR		unchecked
<code>editor</code>	INT	foreign key (Users)	NULL
<code>last_mod</code>	DATETIME		2021-06-21 13:33:19

Table 3.2: Structure of the `Entity` table. Fields and their types are given in the first two columns, the ‘Description’ column gives more details about the fields and an example is provided in the fourth column.

The primary key of the `Entity` table is `entity_id`, which is an `AUTO_INCREMENT` field. This is used as a foreign key in other tables to refer to a certain entity.

The `wn_offset` field stores the wordnet offsets of a given lemma, separated by semicolons, if the word belongs to more than one synset. It does not contain anything in case of sentences and other entities which cannot be found in the wordnet of the given language.

The `text` field contains the word form, MWE or sentence.

As the name suggest, the `type_of_entity` field defines the type of the entity, which can be one of the following five values: `lemma`, `wordform`, `affix`, `mwe`, `sentence`.

Lemma is the most neutral, basic form of words (also called canonical form). The macro-structure of dictionaries contains the headword list, which is generally the same as the list of lem-

mata entered in the dictionary. This designated form depends on the lexicographic traditions of the language. In case of nouns and adjectives, the singular nominative case represents lemmata in both Finnish and Hungarian. Finnish verbs appear in the first infinitive form in dictionaries. For Hungarian, on the other hand, the canonical form of verbs is the third person singular present tense indefinite indicative form.

`Wordform` represents every other possible form of a lemma, any other non-canonical, inflected form.

`Affix` is a bound morpheme that cannot be used without a root. It was observed when validating the translation candidates, that Wiktionary contains affixes as headwords (such as *-ban*, *fel-*, or *-kin*). Their part of speech tag is defined as prefix or suffix in Wiktionary, which does not have an equivalent in the part of speech tagset of UD, so the type of the entity is set to `affix` and the part of speech information is undefined in these cases in the database.

Depending on the literature, fixed, or semi-fixed combinations of words, phrases, compounds, phrasal verbs, and/or idioms can be considered MWE. In this research (and hence, in the database), MWE can be any (compositional or non-compositional) more than one word unit in the broadest sense: i.e. an expression that consists of at least two words, which does not form a full sentence.

A sentence normally conveys a statement, a question, and consists of at least a main clause. In the database, the `sentence` entity type is assigned to entities that have a predicate, begin with a capital letter, and end with a period, an exclamation mark, or a question mark.

An example in both Finnish and Hungarian for each type can be seen in Table 3.3.

Type_of_entity	Hungarian	Finnish
lemma	egyszerű ‘simple’	filosofi ‘philosopher’
wordform	abba ‘that-ILL’	autan ‘help-PRS.1SG’
affix	-etlen ‘-less’	-mpi ‘-er’ (comparative suffix)
mwe	anyák napja ‘Mother’s Day’	samaan aikaan ‘at the same time’
sentence	Richárd elmondta nekünk az érzéseit. ‘Richard told us about his feelings.’	Voitko auttaa minua? ‘Can you help me?’

Table 3.3: Types of entities with examples.

The `lemma` field is empty, unless the `type_of_entity` is set to `wordform`, in which case the canonical form of that word should be provided.

The table contains several fields that function as foreign keys. Such fields are `lang`, `upos`, `label`, `inflection_type`, and `editor`. In Table 3.2 the foreign keys and their referenced tables are given in the `Description` column, and they can also be seen in the Entity Relationship Model in Appendix A.

The `upos` and `ufeat` fields use the tagset and annotation rules of UD in order to provide a framework which is based on universal, language-independent standards, enabling the interoperability between different tools and resources. The `upos` field has a foreign key constraint, referencing the values of the `PartOfSpeech` table (see Table 3.13 below), while the `ufeat` field can contain text data and must be in accordance with the UD notations for lexical and grammatical properties.

Each entity is associated with a register label, which is determined in the `label` field. The default value of this field is `not defined` (which is denoted by number 1), and the editors of the DWS have to change it to the identifier of a specific usage, temporal, or register label, such as “offensive”, “old-fashioned”, or “colloquialism”. For more information about the labels that can be used in this field, see Section 3.4.8.

The `frequency` field contains the occurrences for one million tokens (individual instances of a string), only if the type of the entity is `lemma`. This information was extracted with the help of the Parole corpus²⁸ for Finnish and the Hungarian Word Frequency List²⁹ based on the Hungarian National Corpus.

The number of lemmata to which frequency information could be gathered can be seen in Table 3.4, along with the total number of lemmata for each language.

Language	Number of Lemmata	Lemmata with Frequency
Finnish	307 004	56 218 (18.31%)
Hungarian	229 363	53 945 (23.52%)

Table 3.4: Number and percentage of lemmata where frequency data is known.

The `inflection_remark` field is created to contain any language-specific inflection in-

²⁸ retrieved 24 September, 2022 from <https://kaino.kotus.fi/sanat/taajuuslista/parole.php>

²⁹ retrieved 24 September, 2022 from <https://metashare.nytud.hu/repository/browse/hungarian-word-frequency-list/c1e16f3aaaaf11e3aa7c68b599c26a0615bcb16635874754bcf6b5717986c02e>

formation, like the consonant gradation type (A-M according to Kotus categories³⁰) for Finnish words, and the 4 vowel harmony categories (back vowels, front rounded, front unrounded vowels, and mixed vowels) for Hungarian according to Elekfi (1994) and its digital implementation: E-Szókincs³¹. The online Hungarian dictionary called “*A magyar nyelv nagyszótára*”³² (‘Comprehensive Dictionary of Hungarian’) also follows this system. To increase the interoperability of existing tools and this newly proposed database, the same values are used in the `inflection_remark` field.

The `status` field can take one of the following values: `unchecked`, `checking`, `checked`, `accepted`, `reported`, `correcting`, `merged`, `marked_as_deleted`, and `deleted`. A detailed description of each status can be found in Table 3.5. It shall be noted that the only difference between `checked` and `accepted`, as well as between `marked_as_deleted` and `deleted` is the role of the user who validated the entity.

Status	Description
<code>unchecked</code>	the initial state of every entity and relation, to be manually validated
<code>checking</code>	the entity is linked to an editor who is currently validating it
<code>checked</code>	an editor has validated the entity
<code>accepted</code>	an admin has validated the entity
<code>reported</code>	a user reported an error on the entity
<code>correcting</code>	an editor has been linked to the reported entity
<code>merged</code>	an entity already exists with the same or more information
<code>marked_as_deleted</code>	the entity has been marked as deleted by an editor
<code>deleted</code>	an admin has deleted the entity

Table 3.5: Types of entities and their meaning.

The `editor` field is a foreign key, which refers to the `Users` table, and contains the `user_id` of the editor who is currently editing the entity.

The `last_mod` field contains a timestamp that displays when the entity was modified.

Each row in this table must be unique considering their `type_of_entity`, `text`, `lang`, `upos`, `inflection_type` and `inflection_remark` fields. This means that there cannot be, for example, two separate Finnish lemma entities in the database sharing the same part of speech

³⁰ retrieved 27 September, 2022 from <https://kaino.kotus.fi/sanat/nykysuomi/astevaihtelutyypit.php>

³¹ retrieved 29 September, 2022 from <http://corpus.nytud.hu/cgi-bin/e-szokincs/alaktan>

³² retrieved 12 October, 2022 from <https://nagyszotar.nytud.hu/index.html>

with the same inflection type and remark, and the string in the `text` field. In case they share all these data, there must exist one and only one entity for that data point. It might mean several different concepts depending on the context it appears in, but this information will be encoded by relations, not by the entity itself.

To demonstrate this unique constraint, let us consider the Finnish lemma *laki*. It is many homonymous words with distinct meanings and as such (similarly to polysemes), they have many translation equivalents. In English, it can be translated as ‘law’, ‘code’, ‘act’, and even ‘summit’, ‘ceiling’, etc. When inspecting the details of *laki* respecting the fields that must be unique in the `Entity` table (see Table 3.6), it is clear that two entities must be inserted to satisfy the unique constraint, since the inflection type is different in the two cases (5 and 7). If the values of these fields were identical, it would not be necessary to create more than one entity in the database for this lemma, even if it has two or more diverse meanings.

Field name	Entity 1	Entity 2
text	laki	laki
type_of_entity	lemma	lemma
lang	Finnish	Finnish
upos	NOUN	NOUN
inflection_type	5	7
inflection_remark	D	D

Table 3.6: Information about the polysemous Finnish lemma *laki*.

3.4.2 Relation

This table contains all the relations between two entities or two relations. As can be seen in Figure 3.1, `Relation` has a self-referencing relationship. The structure of the `Relation` table can be seen in Table 3.7.

`Relation_id` is an automatically increasing integer field, that uniquely identifies every relation that gets inserted.

The `member_one` and `member_two` fields are not foreign keys (although they can refer to the primary key of the `Entity` table and hence, have the same type (i.e. INT)). However, a relation can also connect two relations (which is the reason for the self-referencing relationship). Since it is not possible to restrict the values of a field to the primary key of two different tables with the

Name of field	Type	Description	Example
relation_id	INT	auto_increment, primary key	883
member_one	INT		544
member_two	INT		632
member_type	VARCHAR	'entity' or 'relation'	entity
relation_type	INT	foreign key (RelationType)	3
m1_label	INT	foreign key (Label)	1
m2_label	INT	foreign key (Label)	1
status	VARCHAR		unchecked
editor	INT	foreign key (Users)	NULL
last_mod	DATETIME		2021-05-25 11:39:35

Table 3.7: The structure of the Relation table.

help of a foreign key constraint, the solution is to assign the type of the primary keys to the field, allowing any value that has the same type as these unique identifiers.

The self-referencing relationship of this table is not used in the current state of the database, but it ensures that in the future, the `Relation` table can define a connection between two relations, such as creating a relation between a translation pair (e.g. *levél = lehti*) and a sentence pair (e.g. *Itt az ősz, hullanak a levelek. = Syksy on täällä, lehdet tippuvat.*) as the latter (sentence) pair illustrating the usage of the former (lemma) pair. The need for this type of relation is underpinned by the fact that a pair of example sentences (which are translations of each other) can illustrate the usage of a lemma (or two lemmata) in a certain sense, a specific concept, which can be determined and restricted by the relation between two lemmata.

To distinguish the relations between two entities and two relations, a third field is defined (`member_type`), which can only take either 'entity' or 'relation' as value.

The `relation_type` field is restricted by a foreign key constraint, containing one of the primary keys of the `RelationType` table, see Table 3.8.

A translation pair is inserted into the database in two steps: first, the two lemmata of the translation pair are inserted into the `Entity` table, and then, a relation between them is inserted into the `Relation` table, using their `entity_id` identifiers in the `member_one` and `member_two` fields, and the `member_type` defined as 'entity'. The type of relation would refer to the `relation_id` of the `RelationType` table (number 3, because it is a translation pair).

To avoid data redundancy, and decrease the number of entities that are inserted in the database,

relation_id	type
1	synonym
2	antonym
3	translation
4	example_sentence
5	definition
6	abbreviation
7	description
8	alternative_spelling
9	hyponymy-hypernymy
10	meronymy-holonymy
11	collocation

Table 3.8: The contents of the RelationType table.

two other fields had to be defined in the `Relation` table. The `m1_label` and `m2_label` fields contain the specific usage, temporal, or register labels that are characteristic to the entities that are part of the relation. To illustrate the need for this, let us take the example of the translation pair *tonni* ‘metric ton’ = *lepedő* ‘bedsheet’. These lemmata in their most neutral sense would not be considered translation equivalents, yet, if used in slang, they can both refer to a ‘grand’ (or one thousand units of currency). Therefore, usage and register nuances do not exclusively appear on the entity level in the system. In this case, every piece of information about the entity would be duplicated, if the creation of new entities meant that only the style label is different from an already existing entity. Instead, the different labels can also be modified on the relation level. In this case, the label of the *lepedő* entity is the most neutral one (number 2), while in the `Relation` table, when creating a translation pair with the lemma *tonni*, the label is defined as slang (number 16).

The `status` of a relation can have the same values as an entity, for more details, see Table 3.5.

The `editor` and `last_mod` fields are also similar to those of the `Entity` table.

3.4.3 Source

There is a table in the database that stores the information about the algorithms that were used to generate the translation candidates. The `Source` table stores in case of each entity and relation the algorithm name which led to that entity or relation. In case there are more algorithms that resulted in the same entity or relation, the table stores them all, as separate records. The structure of the

Source table can be seen in Table 3.9.

The `source_type_id` is a foreign key, referring to the `SourceType` table. For more details, see Section 3.4.4.

The `member_id` is the identifier of either the entity or the relation, depending on what information is given in the `member_type` field (either ‘entity’ or ‘relation’), similarly to the field in the `Relation` table with the same name.

The `status` signals if the source of the entity is ‘ok’ or ‘deleted’, which may occur, when merging two entities. About merging, see Chapter 5.2.2.

Name of field	Type	Description	Example
<code>source_id</code>	INT	auto_increment, primary key	36761
<code>source_type_id</code>	INT	foreign key (SourceType)	1
<code>member_id</code>	INT		18916
<code>member_type</code>	VARCHAR	entity or relation	relation
<code>status</code>	VARCHAR	ok or deleted	ok

Table 3.9: The structure of the Source table.

3.4.4 SourceType

The `SourceType` table contains all the methods and algorithms which were applied, while collecting and extracting lexical data from different sources. Table 3.10 displays the contents of the `SourceType` table. A small remark must be made regarding the record with the algorithm name `WordNet` with `source_id` 2. This – just like the first record with the algorithm name `WordNet Connector` – refers to the `WordNet Connector` algorithm, with the difference that `WordNet Connector` refers to the connection between the Finnish and Hungarian language editions, while the other one (`WordNet`) refers to lexical relations that come from only one `WordNet`, such as Hungarian example sentences, or Finnish synonyms.

3.4.5 Remark

Every editor of the DWS can leave comments and remarks on entities and relations. These get saved in the `Remark` table (see Table 3.11), along with a unique identifier, whether it is an entity or a relation that the comment was made on, what is the ID of that entity or relation (`member_id`),

source_id	algorithm_name	source_name
1	WordNet Connector	WordNet
2	WordNet	WordNet
3	Wiktionary Parser	Wiktionary
4	wikt2dict extract	Wiktionary
5	wikt2dict triangulate	Wiktionary
6	OPUS Extractor	OPUS

Table 3.10: The contents of the SourceType table.

the status of this remark, the ID of the user who made the comment, and the date and time when it was written.

Name of field	Type	Description	Example
remark_id	INT	auto_increment, primary key	13
member_id	INT		538
member_type	VARCHAR	'entity' or 'relation'	entity
remark	TEXT		Pluralia tantum.
status	VARCHAR		active
user_id	INT	foreign key (Users)	6
date	DATETIME		2022-02-01 12:43:37

Table 3.11: The structure of the Remark table.

The status of the remarks can be 'active' and 'deleted'. If a remark is made only to indicate a problem or a mistake regarding an entity or a relation, after the issue is fixed, the remark itself can be marked as done, or 'deleted' in this case.

3.4.6 Users

The users of the DWS and language learning application are stored in the `Users` table (see Table 3.12). Each user has a unique identifier (an automatically increasing integer), a username a password, and an email address that the system asks for upon registering. The role of a user can be one of the following: 'admin', 'teacher', 'guest', 'editor' or 'player'. Depending on the role, the user will have the permission to access different pages and perform different actions. For instance, players can only access the language learning application and the user interface of the dictionary, while admins can delete entities or relations, and modify the role of other users in the DWS. The complete list of actions that can be performed by users with different levels of authentication can

be seen in Appendix D.

Name of field	Type	Description	Example
user_id	INT	auto_increment, primary key	43
username	VARCHAR		test_user
password	VARCHAR		5F4DCC3B5AA7...
email	VARCHAR		testuser@email.com
role	VARCHAR	'admin', 'teacher', 'guest', 'editor', 'player'	admin
logged_in	tinyint		0
last_request	DATETIME		2022-05-29 14:26:01

Table 3.12: The structure of the Users table.

The date and time of the last request when the user performed an action on the website is stored in the `last_request` field. With the help of this field, it is possible to automatically log out inactive users.

3.4.7 PartOfSpeech

The Universal Dependencies tagset is used to label the lemmata with their part of speech tags. A table is dedicated to store this information, in order to have a common tagset for each and every language in the database, and to ensure that the database and dictionary are language-independent.

All the part of speech tags in this table (see the full list in Table 3.13) come from the official website of the Universal POS tags³³, except the `NONE` tag. It is necessary to have a default value, when lemmata and sentences are inserted in the `Entity` table without a determined part of speech tag. The `NONE` tag serves as a not defined part of speech, since the `upos` field shall not be empty (`NULL`). Otherwise, it would be possible to insert duplicate entries (where the `upos` field is `NULL`, and the rest of the fields are identical) in the `Entity` table, which do not satisfy the unique constraint.

3.4.8 Label

In the case of rare, offensive, or dialectal headwords, dictionaries use a set of style and region labels to indicate that the word is marked considering its domain of application.

³³ retrieved 24 September, 2022 from <https://universaldependencies.org/u/pos/>

pos_id	pos_tag
1	NONE
2	ADJ
3	ADP
4	ADV
5	AUX
6	CCONJ
7	DET
8	INTJ
9	NOUN
10	NUM
11	PART
12	PRON
13	PROPN
14	PUNCT
15	SCONJ
16	SYM
17	VERB
18	X

Table 3.13: The contents of the PartOfSpeech table.

A list of usage, temporal, and register labels was collected, and inserted in the `Label` table (shown in Table 3.14). This list is extensible and new items can be added any time with the help of a simple `INSERT` statement.

The `label_id` field stores a unique identifier that automatically increases. The `label_name` field defines the label in a concise way in English, while an abbreviation is given to this label in the `label_abbrev` field. The first element (`not defined`) in this list is the default label that is assigned to newly inserted entities. During the design phase, the creation of label abbreviations was considered. These abbreviations facilitate the translation of the interface of the dictionary and the DWS into three languages: Finnish, Hungarian and English. The labels are translated into all three languages, and even if the term in English changes, the base for the translations (the abbreviation of the label) does not need to be modified, therefore, there would not be a need to change anything on the interface.

label_id	label_name	label_abbrev
1	not defined	nd
2	standard	std
3	colloquialism	colloq
4	literature	liter
5	old-fashioned	old
6	humorous	hum
7	law	law
8	impolite	imp
9	offensive	offen
10	derogatory	der
11	vulgar	vulg
12	dialectal, regional	dial
13	rare	rare
14	ironic	iron
15	child's word, expression	child
16	slang	slang
17	argot	arg
18	journalism	journ
19	official	offic
20	science	sci
21	finance, business	finan
22	informal discussion	inform
23	figuratively	fig
24	politics	pol
25	formal	form
26	maths	maths

Table 3.14: The contents of the Label table.

3.4.9 Languages

The database schema is designed and created to be used as a language-independent model, so that in the future, any other FU languages can be added without needing to restructure it. To make the list of languages extensible and flexible, hard coding the values needs to be avoided, hence, the list of languages is stored in the `Languages` table (see Table 3.15).

The `lang_id` is a unique identifier which is automatically increasing when inserting a new record into the table. The `lang_code` is the ISO 639-1 code of the language, while `lang_name` is the English name of the language.

To date, lexical data is extracted only for the Finnish–Hungarian language pair, however, it is

lang_id	lang_code	lang_name
1	fi	Finnish
2	hu	Hungarian

Table 3.15: The contents of the Languages table.

possible to extend the list of entities and relations with e.g. Estonian data. Hence, a new record can be added in the Languages table with the `et` language code, and `Estonian` as the value of the `lang_name` field.

3.4.10 Inflection

As agglutinative languages, both Finnish and Hungarian have an extensive case system, and the different word forms of nouns, adjectives, as well as verbs depend on the properties of the words. Inflection types can be determined in order to group different words together whose inflection patterns are identical. These types can help learners of such languages understand and acquire these (limited number of) patterns, instead of memorizing the paradigm of each lexeme.

The Finnish inflection system included in this database is based on the inflection types found in Kotus³⁴. This resource lists 49 noun (plus two additional types for compound nouns) and 27 verb inflection types, all of these categories are exemplified by a typical lemma, which belongs to that inflection type. There is a separate category for pronouns and a category that contains words with irregular inflection patterns or with defective paradigms.

The Hungarian types are based on the inflection tables of `e-szókincs` available online³⁵. This interface uses the system established by Elekfi (1994): in this system, there are 36 noun and 36 verb types. Some of these are described by the properties of the words belonging to the group. In contrast, others do not have any description and are denoted by a sole identifier (such as `n31` or `v25`). The users of the dictionary, however, will not encounter these descriptions, since the entries of the dictionary will point them to the corresponding websites where the learner can examine the inflection tables of different lemmata.

The inflection types are stored in a database table (`Inflection`) which consists of the fields

³⁴ retrieved 24 September, 2022 from <https://kaino.kotus.fi/sanat/nykysuomi/taivutustyyppit.php>

³⁵ retrieved 11 November, 2022 from <http://corpus.nytud.hu/cgi-bin/e-szokincs/alaktan>

listed in Table 3.16. The `inflection_id` field is an automatically increasing identifier that serves as primary key of this table. The `inflection_type` field stores the original identifiers that the chosen systems use to denote the types. The `lang` field is a foreign key that refers to a language to which the inflection type belongs. The `pos` field contains a list of part of speech tags that the inflection type can be applied to. The `description` field contains an explanation that makes the inflection type recognizable to editors: for Finnish types, it stores the example word of the type, and for Hungarian, it contains the short description applied by the inflectional system. A subset of the contents of this table can be found in Table 3.17. The entire table can be seen in Appendix C.

Name of field	Type	Description
<code>inflection_id</code>	INT	auto_increment, primary key
<code>inflection_type</code>	VARCHAR	
<code>lang</code>	INT	
<code>pos</code>	VARCHAR	
<code>description</code>	VARCHAR	

Table 3.16: The structure of the `Inflection` table.

<code>inflection_id</code>	<code>inflection_type</code>	<code>lang</code>	<code>pos</code>	<code>description</code>
1	NULL	NULL	NULL	NULL
2	1	1	NOUN, ADJ, NUM, PROP	valo
53	52	1	VERB	sanoa
83	1	2	VERB	iktelen alapminták
121	21	2	NOUN, PRON, NUM	változatlan magánhangzós tövű névmások, számnevek, hiányos főnevek

Table 3.17: A subset of the contents of the `Inflection` table.

In order to help learners of Finnish and Hungarian correctly inflect the newly acquired words, the inflection type is provided in the case of each lemma in the `Entity` table, referencing the values present in the `Inflection` table. The inflection type was assigned to each lemma automatically when populating the database (see Section 3.5). However, when such information can not be obtained, the first record of this table (where almost all fields contain the `NULL` value) gets automatically assigned by default. The editors are required to provide the inflection type to lemma

entities when validating them in the DWS, in order to further enrich the lexical data available in this database and provide learners with the appropriate information regarding as many lemmata as possible.

3.5 Populating the Database

To populate the database with hundreds of thousands of data, an insertion script was developed and used. The output of the three newly proposed methods in this research work (`WordNet Connector`, `Wiktionary Parser` and `OPUS Extractor`), as well as that of the `wikt2dict` methods, is processed by this script, which reads data from the given files, connects to the database, and inserts each entity and relation with all available information. Therefore, this script also assigns the inflection type and consonant gradation information to the Finnish lemmata using the *Kotimais-ten Kielten Keskuksen Nykysuomen sanalista* ('Word List of the Center for Domestic Languages in Modern Finland', Kotus for short)³⁶, which is a freely available XML file containing 94,110 word records enriched by inflection type and consonant gradation information where applicable. This vocabulary list is licensed under the GNU Lesser General Public License (LGPL), European Union Public Licence v.1.1 (EURL) and Creative Commons Attribution 3.0 Unported (CC BY 3.0) licenses. The number of lemmata where inflection type and consonant gradation information can be found in this resource is shown in Table 3.18. This table also contains statistics about the number of Finnish entities to which inflection types and consonant gradation categories could be assigned.

When populating the database with translation candidates, the insertion of data is conducted differently depending on the extraction method. The output of some of the methods contain only the translation candidate and part of speech information, while the output of the `WordNet Connector` algorithm contains more data, such as the wordnet offsets which are associated with the translation candidates.

As a first step, the insertion of the separate entities is carried out, with all available and obtainable information about each of them (language, type of entity, part of speech, as well as inflection type, and consonant gradation in case of Finnish entities).

The next step, after the insertion of both entities that form the translation candidate, is to insert

³⁶ retrieved 24 September, 2022 from <https://kaino.kotus.fi/sanat/nykysuomi/>

Source	Type	# of Instances
Kotus	Finnish lemmata	94,110
Kotus	inflection type and consonant gradation	11,738
Kotus	only inflection type	32,610
Kotus	only consonant gradation	0
Kotus	none	49,762
Database	Finnish lemmata	59,946
Database	inflection type and consonant gradation	5,063
Database	only inflection type	13,139
Database	only consonant gradation	0
Database	none	41,744

Table 3.18: Statistics about inflection type and consonant gradation for Finnish lemmata.

a relation with the `entity_id` of the two (newly inserted) entities, their type ('entity') and their type of relation (in this case number 3 to denote a 'translation' relation).

Similarly to translation relations, synonyms, definitions and example sentences are also processed and the `Entity` and `Relation` tables are populated. The `WordNet Connector` method extracts not only translation pairs, but also a list of synsets for both Finnish and Hungarian. The same data insertion script inserts the synonym relations between Hungarian, as well as between Finnish synonyms. The elements of each synset are combined with each other, and pairs are created that can be inserted in the `Relation` table as synonyms.

The Hungarian WordNet, and both Wiktionary editions contain example sentences and definitions for some of the lemmata. These sentences are uploaded into the `Entity` table with 'sentence' as their `type_of_entity`, and the relation between the sentence and the lemma also gets inserted into the `Relation` table afterwards.

The final step is to populate the `Source` table. Three records are inserted in total for each relation, two for the entities, and one for the relation between them. In all three cases, the method which led to the (translation, example sentence, definition, etc.) relation is indicated with the primary key of the appropriate record from the `SourceType` table (see Table 3.10). This is done automatically by looking up the name of the method in the `algorithm_name` field and selecting its primary key for insertion.

Another script was developed to add frequency information to lemmata. This procedure first calculates the frequency per million tokens for each lemma, and then updates this number for the

entities in the database. The Hungarian frequency was based on the Hungarian National Corpus frequency list³⁷. It is calculated on a 180 million word corpus. This list provides frequency information for 10,403,686 word forms. For obtaining frequency information for Finnish lemmata, the frequency list based on the Parole corpus³⁸ was used. This list contains 1,339,787 word forms, the corpus itself contains 17,604,995 tokens after normalizing and cleaning the data. Data normalizing consisted of the following steps:

1. converting upper case letters to lower case,
2. connecting sequences of numbers with the ‘_’ character,
3. removing punctuation (except the ‘-’ and ‘.’ characters inside the words, which connect suffix to an abbreviation, e.g. *USA:ssa*), and
4. removing strings which do not contain any alphanumeric characters.

To illustrate the insertion of entities into the database, an INSERT statement can be used such as the one in SQL query 1. The obligatory fields that need to be provided upon insertion of new entities are the `text` and `lang` fields, while the rest of the fields get the default values whenever they are not included in the query. For example, the `type_of_entity` field would store `lemma` for both *koira* and *kutya* in the example query. The list of default field values in the `Entity` table is provided in Section 3.4.1.

```
INSERT INTO Entity (text, lang, upos)
VALUES ('koira', 1, 9), ('kutya', 2, 9);
```

SQL query 1: Example query to insert entities in the `Entity` table.

To insert data into the `Relation` table, the INSERT statements were similar to the query that is presented in SQL query 2.

³⁷ retrieved 24 September, 2022 from <https://metashare.nytud.hu/repository/browse/hungarian-word-frequency-list/c1e16f3aaaaf11e3aa7c68b599c26a0615bcb16635874754bcf6b5717986c02e/>

³⁸ retrieved 24 September, 2022 from <https://kaino.kotus.fi/sanat/taajuuslista/parole.php>


```
INSERT INTO Relation (member_one, member_two, relation_type)
VALUES (127654, 144239, 3);
```

SQL query 2: Example query to insert relations into the `Relation` table.

3.6 Views

Views are SQL statements stored in the database. These statements usually process and perform actions on a table (or a combination of tables), which result in another table, consisting of the requested fields and combined data filtered by some potential conditions defined in the query. They allow the restructuring of data present in the tables of the database. Besides these, a view also provides a fast way to reapply often-used queries.

It was clear from the initial stages of designing the database that several complex queries would reappear in the code base. It occurs most frequently when information from the database must be accessed in a certain way or for a particular purpose. As a consequence, three views were created to enable the simplification of more complex queries.

3.6.1 ViewRelation

The `Relation` table (as described in Section 3.4.2) is where entities get connected to each other, creating pairs and defining the type of connection they represent. The way the `Relation` table was designed (to reduce data redundancy) does not allow access to the details of the entities directly. The table contains two references to the participating entities, which refer to the unique identifier `entity_id` in the `Entity` table.

However, many times it would be desired to display the special characteristics of the two relating entities, such as the strings in their `text` field, part of speech information, and their frequency. These pieces of information are not accessible from the `Relation` table. To see the information of both entities in the relation, two instances of the `Entity` table need to be joined with the `Relation` table. The linking of tables is made possible by the fields that act as foreign keys (`member_one` and `member_two` in the `Relation` table). Such a query is not only long but can also be quite complex, especially when adding all the necessary conditions and keywords to reach

the desired results (see SQL query 3).

```
SELECT E1.text AS m1_text,
       E2.text AS m2_text
FROM Relation
INNER JOIN Entity AS E1 ON E1.entity_id = Relation.member_one
INNER JOIN Entity AS E2 ON E2.entity_id = Relation.member_two
WHERE Relation.member_type = "entity"
AND Relation.relation_type IN (
    SELECT relation_id FROM RelationType WHERE type = "translation"
)
LIMIT 5;
```

SQL query 3: Querying the list of translation candidates present in the database.

The result of this query can be seen in Table 3.19. The `LIMIT` clause restricts the number of resulting records to five. It is important to mention that the resulting relations are not filtered by their status (whether they have been validated or not by the editors of the DWS), hence, they may include incorrect translation candidates as well.

Exploiting the power of views, it is possible to substitute this query with the query shown in SQL query 4. The use of nested queries can be avoided in many cases with the help of this view.

m1_text	m2_text
älytön	agyalágyult
sekopäinen	tökéletlen
hullunkurinen	bolondos
alentua	csitul
sekopäinen	lökött

Table 3.19: Results of SQL query 3. Note: these translation candidates are to be manually validated.

3.6.2 ViewSourceEntity

Important observations can be made and interesting conclusions can be drawn regarding the extracting algorithms if two tables (`Source` and `Entity`) are combined. To do that, the `View-`

```
SELECT m1_text, m2_text
FROM ViewRelation
WHERE rel_type = "translation";
```

SQL query 4: Querying the list of translation pairs with the help of the `ViewRelation` view.

`SourceEntity` view was created, which joins these two tables by matching the values of the `member_id` and `entity_id` fields. The query that is stored as the view can be seen in SQL query 5.

```
SELECT SourceType.algorithm_name,
       SourceType.source_name,
       Entity.*,
       Source.status AS source_status
FROM Entity
INNER JOIN Source ON Source.member_id = Entity.entity_id
                AND Source.member_type = "entity"
INNER JOIN SourceType ON SourceType.source_id = Source.source_type_id;
```

SQL query 5: Query that is replaced by the `ViewSourceEntity` view.

3.6.3 ViewSourceRelation

Similarly, displaying the relations in connection to the sources which they were obtained from can be valuable. It is important to mention that views can be used when creating other views, which was exploited here in order to also include the data available in the `Entity` table. In the same manner, as previously described in the case of the `ViewSourceEntity` view, with the help of the `ViewRelation`, `SourceType`, and `Source` tables, a third view was generated.

As it was expected in the early stages, these views are used several times in the web application, for instance, the number of mentions of the `ViewRelation` view is more than 30.

3.7 Triggers

A trigger is a set of SQL statements, a stored procedure, that is invoked automatically when a predetermined event happens. A trigger can be invoked when, for example, an `INSERT` or `UPDATE` statement is run. Triggers are used to check the integrity of data or manipulate the data that is about to be inserted into a table.

There are several triggers saved in the database, that facilitate the work of editors and make sure that the inserted data is valid.

One of the most important triggers is the `mwe_trigger`, which is called before each insertion of data into the `Entity` table. It ensures that the `type_of_entity` of a lemma that has at least two words is changed to `mwe`.

Two other triggers make sure that the changes that happen in the `Entity` and `Relation` tables are tracked and a log is created about each modification. These triggers are called `log_entity_trigger` and `log_relation_trigger`. Before an `UPDATE` statement is run on any of the two tables, the previous state of the entity or relation that is about to be modified gets saved in a log table. This enables tracking the changes made by a specific editor, backing up the data from an earlier state, or viewing how a specific entity or relation changed throughout a determined period of time.

3.8 Querying

Retrieving data from the database is feasible with `SELECT` statements in a language called Structured Query Language (SQL), as previously mentioned. These statements can be composed of many parts, depending on the complexity of the search.

As a result of how the database was structured and designed, data retrieval is quite straightforward and the tables, as well as the views, allow the extraction of meaningful data in a compact, easily comprehensible way. The rest of this section illustrates how the SQL statements can be and are used in the application (described in Chapter 5) to extract different subsets of the data. In the following, several example queries are given, along with what result is expected and provided in each case.

To start with a simple query, let us consider a scenario, in which the list of Hungarian lemmata

starting with the letter *p* is requested. A `SELECT` statement can be applied similar to what is shown in SQL query 6.

Since the `Entity` table stores all the lexical units, the `SELECT` statement selects the `text` field from this table. There are three conditions that need to be fulfilled which are specified in the `WHERE` clause:

1. the `type_of_entity` field needs to be `'lemma'`,
2. the `'p%'` pattern is searched for in the `text` field, and
3. the language of the entity must be Hungarian.

```
SELECT text FROM Entity
WHERE type_of_entity = 'lemma' AND text LIKE 'p%'
AND lang IN (
    SELECT lang_id FROM Languages WHERE lang_name = 'Hungarian'
);
```

SQL query 6: Retrieving the list of Hungarian lemmata starting with *p*.

There are two wildcard characters that can be used with the `LIKE` operator to compare strings: the percent sign (%) matches zero or more characters, while the underscore sign (_) matches exactly one character. The language identifier is selected in a nested query, which returns the corresponding language ID of the `'Hungarian'` language record in the `Languages` table (in this case, this would be 2).

As a more complex example, it is possible to evaluate the extraction methods regarding the example sentence relations. An SQL statement can be used to create a report about how many of the already validated relations have been marked as correct, and how many of them have been deleted. These numbers can be calculated by counting the number of relations for each language and status. A possible `SELECT` statement to achieve this is shown in SQL query 7.

This query exploits the power of already existing views. The `ViewSourceRelation` view joins the `ViewRelation` view to the `Source` and `SourceType` tables and selects a subset of all the fields in these three data collections to be contained in the view (views are described in more detail in Section 3.6).

```

SELECT
    COUNT(DISTINCT relation_id) AS num_of_relations,
    lang_name AS language,
    status,
    algorithm_name
FROM ViewSourceRelation
INNER JOIN Languages
ON Languages.lang_id = m1_lang
WHERE status IN ('accepted', 'checked', 'deleted',
                'marked_as_deleted')
AND rel_type = "example_sentence"
GROUP BY status, algorithm_name, lang_name
ORDER BY lang_name, algorithm_name;

```

SQL query 7: Querying the evaluation of example sentences.

The `COUNT` function determines the number of distinct relations by summing them for each combination of language, status, and algorithm (according to the fields that are determined after the `GROUP BY` statement). The number is only displayed in the resulting table if it is other than zero. The `ORDER BY` keyword sorts the results and organizes the records first by the language, followed by the algorithm names in alphabetical order. The following two conditions are defined in the query:

1. the status of the relation must be any of the following: 'accepted', 'checked', 'deleted', or 'marked_as_deleted', since they are the only ones that mark a relation as validated (see Table 3.5 for more details), and
2. the relation type must be 'example_sentence'.

This query produces the result set shown in Table 3.20.

Another interesting idea regarding the data set is to find out which are the most frequent lemmata in Finnish and which are in Hungarian. One possible solution to produce this kind of data is shown in SQL query 8, which takes advantage of the `UNION` operator. It is used to combine the results of two `SELECT` statements: in this case, querying the top ten most frequent lemmata of Finnish with the first `SELECT` statement, and then those of Hungarian with the second.

num_of_relations	language	status	algorithm_name
185	Finnish	accepted	Wiktionary Parser
26	Finnish	deleted	Wiktionary Parser
148	Hungarian	accepted	Wiktionary Parser
8	Hungarian	deleted	Wiktionary Parser
135	Hungarian	accepted	WordNet
105	Hungarian	deleted	WordNet

Table 3.20: Result of SQL query 7.

In the nested statements, the records are sorted by descending frequency, and the `LIMIT` clause only allows the first ten records to be displayed in the result set. The conditions that must be applied here are:

1. the status of an entity must be different from ‘deleted’ and ‘marked_as_deleted’,
2. the language should be ‘Finnish’ in the first nested statement,
3. the language should be ‘Hungarian’ in the second nested statement, and
4. the `type_of_entity` field must be ‘lemma’.

Table 3.21 presents the output of SQL query 8, showing the top ten most frequent lemmata in both Finnish and Hungarian.

text	lang	text	lang
olla	Finnish	a	Hungarian
ja	Finnish	és	Hungarian
ei	Finnish	az	Hungarian
hän	Finnish	ez	Hungarian
joka	Finnish	van	Hungarian
se	Finnish	egy	Hungarian
saada	Finnish	ha	Hungarian
tämä	Finnish	csak	Hungarian
tai	Finnish	már	Hungarian
suomi	Finnish	s	Hungarian

Table 3.21: Results of SQL query 8.

After observing the result set, it is perhaps necessary to refine the initial `SELECT` statement, in case we want to exclude function words (such as articles, pronouns, adpositions, adverbs, and conjunctions). An additional condition can be inserted in both nested queries which would restrict

```

SELECT text, lang FROM (
  (SELECT DISTINCT text, "Finnish" AS lang
   FROM Entity
   WHERE status NOT IN ("deleted", "marked_as_deleted")
   AND lang IN (
     SELECT lang_id FROM Languages WHERE lang_name = "Finnish"
   )
   AND type_of_entity = "lemma"
   ORDER BY frequency DESC
   LIMIT 10)
 UNION
 (SELECT DISTINCT text, "Hungarian" AS lang
  FROM Entity
  WHERE status NOT IN ("deleted", "marked_as_deleted")
  AND lang IN (
    SELECT lang_id FROM Languages WHERE lang_name = "Hungarian"
  )
  AND type_of_entity = "lemma"
  ORDER BY frequency DESC
  LIMIT 10)
)
AS most_frequent_lemmata;

```

SQL query 8: The ten most frequent Finnish and the ten most frequent Hungarian lemmata.

the `upos` field of lemmata to a small number of parts of speech. When the additional condition is applied, the following result table is produced (see Table 3.22).

3.9 Conclusion

In this chapter, research question 3 was analyzed in detail and the data structure, as well as the database schema, was presented. The summarized results will be presented in this section.

According to the relevant research question, two major requirements have to be fulfilled by the proposed database: it must be language-independent and it must facilitate the compilation of an

text	lang	text	lang
olla	Finnish	van	Hungarian
ei	Finnish	egy	Hungarian
saada	Finnish	kell	Hungarian
tai	Finnish	lesz	Hungarian
suomi	Finnish	tud	Hungarian
sanoa	Finnish	magyar	Hungarian
pitää	Finnish	szerint	Hungarian
tehdä	Finnish	lehet	Hungarian
mies	Finnish	év	Hungarian
ihminen	Finnish	mond	Hungarian

Table 3.22: Results of SQL query 8 with an additional condition applied to the part of speech of the resulting words.

automatically reversible bilingual dictionary. Therefore, when designing the database, it was of great importance to create a language-independent framework, which is able to store any kind of lexical information in a non-redundant way. The most suitable data representation type and format, in this case, was a relational database, which allows simultaneous access for multiple users who can modify its content and add new data records. Additionally, in order to facilitate the compilation of a dictionary that is automatically reversible, a universal data model needed to be designed. The proposed model steps away from traditional lexicography approaches, as the main unit of the database is not the dictionary entry, but the lexical entity. This ensures that any kind of lexical data (be it lemma, sentence, or MWE) in any language is processed in an identical way, and stored in the database without overcomplicating the data structure. The dictionary that is compiled from the database will be described in Chapter 5. The presented structure of the database is flexible and can be used to compile several types of dictionaries based on its contents. The architecture is extensible and easily adaptable.

The detailed schema of the proposed database was presented in Section 3.4. The views and triggers of the proposed schema, which are supposed to facilitate the work of lexicographers and reduce the amount of human error by automating repeating tasks, were described in Section 3.6 and Section 3.7, respectively. Different use cases were also presented, to illustrate how the database can be utilized and queried in real-life scenarios, see Section 3.8.

4 Computer-Assisted Language Learning

4.1 What is CALL?

In order to bring adult learners to a level where they can function minimally in the target language, a teacher is required to work 60 to 100 hours with them (Nerbonne 2005). The number of hours needed to reach higher proficiency levels (where learners are able to communicate more complex thoughts more efficiently and without long pauses) doubles at each level. Learners can (and are required to) also practice without an instructor beyond the classroom, since the availability and capacity of professional language teachers are limited. Computers can fulfill some of the purposes and tasks that emerge in a language class. Learners only need to have a certain level of motivation to study (Dörnyei 1998), and perfect their language skills on their own. The computer can offer a great number of exercises in order to do that. As Faltin (2003: 137) suggested: “[...] what could, better than a computer, repeatedly and relentlessly propose exercises to the learners?”. Not only can it continuously and relentlessly offer language tasks, but it is faster, more accurate, and more thorough than humans (Rundell 2012). As an additional advantage, portable, handheld devices can be used anytime and anywhere.

Therefore, electronic devices seem to be the appropriate medium and helpful instruments regarding foreign language teaching, learning and assessment. The combination of language learning and computer technology, in general, is called Computer-Assisted Language Learning (CALL). CALL is an interdisciplinary field, that aims to provide tools and programs that serve the purpose of improving the language skills of learners.

Throughout the literature, it can be seen that different authors have very similar, consistent viewpoints on how CALL can be defined. For instance, at the end of the 1990s, Levy defined it as “the search for and study of applications of the computer in language teaching and learning” (1997: 1). This definition is widely accepted among other CALL researchers. According to Egbert “CALL means learners learning language in any context with, through, and around computer technologies” (2005: 4). Another definition was given by Beatty, who emphasized the role of the learner in CALL, and focused on the improvement the learner can make with the help of CALL methods and programs. As he pointed out, CALL is “any process in which a learner uses a computer and, as a

result, improves his or her language” (2003: 7).

As all these definitions demonstrate, CALL is a broad term, encompassing a variety of issues and directions. The field of CALL covers areas such as language material development, testing, and assessment of language skills, as well as teacher training.

With the increasing number of learners and elevated need for language materials and exercises, automatizing language teaching processes, and generating tasks that facilitate the practice of any foreign language skills are the main objectives of CALL, and more specifically, NLP-enhanced CALL nowadays. This subarea of computer-aided language instruction will be presented in more detail in Section 4.2.2.

Advances in technology have led to significant developments in the field of CALL, which can be demonstrated by the constantly changing approach of how computers and programs can be integrated into and utilized in the language learning process. In the next section, these different approaches and the evolution of CALL will be presented, while in Section 4.2.1 the status of CALL systems in education will be examined.

4.2 Evolution of CALL

Since the beginning of the history of CALL, there have been many studies and research proposing newer and better techniques and methods to facilitate and optimize the language learning process. The different techniques and CALL approaches can be described chronologically, tracing the evolution of this field (see e.g. Warschauer and Healey (1998)). It is not surprising, that mainstream theories of different eras have a profound impact on this interdisciplinary field. However, it is also possible to categorize approaches according to the types of activities, the role of teachers, and the feedback that is offered by the software in use (see the analysis of Bax (2003)).

Behaviorist models of cognitive theory affected greatly language teaching methods. From the earliest stages of CALL (as early as the 1950s), this effect can be observed and hence, Warschauer and Healey (1998) call this phase “behaviouristic CALL”. According to Warschauer and Healey, during this stage, it was emphasized that learners should repetitively practice specific aspects of the language, and feedback (negative, as well as positive) played a central role in language instruction (Thomas et al. 2012). To relieve instructors of the creation and assessment of these repetitive tasks, and provide learners with exercises to repeatedly encounter the same material from differ-

ent perspectives, drill-and-practice exercises were developed. One of the earliest and best-known examples of tutorial systems with drill activities was incorporated in the PLATO educational computing system (Hart 1995).

Although Warschauer and Healey tried to attach dates to each stage they identified, the reader is reminded that these phases are difficult to link to historical periods, since they are not mutually exclusive, and several of the dominant ideas can coexist at the same time.

This contradiction is observed by Bax (2003), who proposed replacing the term “phase” with “approach” in order to resolve this inconsistency. In his analysis, it is mentioned that the naming of the first phase (“behaviouristic CALL”) causes conceptual confusion in the literature. Delcloque (2000) and Tang et al. (2009) uses the incorrect term “behavioral CALL” to refer to this first stage. Instead, he suggested that the approach be termed as “restricted CALL”, considering the types of activities and the teachers’ role that characterize this category.

With Web-based tools and personal computers becoming more and more prominent, drill exercises were succeeded by other kinds of activities. Warschauer and Healey (1998) call this phase (which emerged in the early 1980s) “communicative CALL”. This resonated with the methodological frameworks applied in the classrooms at the time. However, Bax (2003) argues that these activities actually did not exactly correspond to the communicative practices enacted in the classrooms around the estimated time of this phase, hence, the term “communicative” (referring to the approach of communicative language teaching) is unacceptable. As a solution, he revised the software and practices employed in that period, and according to their main characteristics, termed this approach “Open CALL”.

During the third phase (called “integrative CALL” by Warschauer and Healey (1998)), attempts are made to integrate various skills (speaking, listening, writing, and reading) and technology “more fully into the language learning process” (Warschauer and Healey 1998: 58). It is dated to the 21st century by Warschauer (2000), and according to him, the central aim of this stage is to encourage learners to participate in authentic discourses. Stating that increased emphasis is placed on the use of language in authentic contexts after the phase that is called “communicative CALL”, is odd according to Bax (2003). The approach that he suggests instead (“integrated CALL”), on the other hand, could only be reached, according to him, when computers are completely integrated into education, and they are used every day by language learners. In this approach, computer-mediated communi-

cation and e-mail are examples of what kinds of activities learners are encouraged to carry out in order to improve their performance. Bax argued that – at the time of writing – these technologies were only starting to be integrated, therefore their usage was not yet normalized.

4.2.1 CALL in Education

When examining the position of CALL systems and software in the syllabus of language classrooms, it is noticeable that these kinds of resources and tasks have become more and more integrated into the curriculum with the progression of history. Nowadays, many instructors try to include computers and the opportunities this technology provides in their language classes. This is also exemplified by the growing number of online language courses, which require the learners to use computers in order to participate in the class and practice the language that way. The editors of many language textbooks provide online exercises and additional material to help learners to deepen their knowledge even further, beyond the classroom. Instructors of language classes at university also try to include computer technology as much as possible in the curriculum. For instance, at the Faculty of Humanities at Eötvös Loránd University, the Canvas learning management system (LMS) is at the disposal of teachers since 2017. It enables them to organize assignments and course materials, and create different types of quizzes and exercises, as a supplement to the classroom-based courses. The list of quiz types can be seen in Figure 4.1. This LMS is just one example of those online learning platforms that facilitate the creation of CALL exercises. However, it is important to note that these exercises must be created manually, which is time-consuming and laborious.

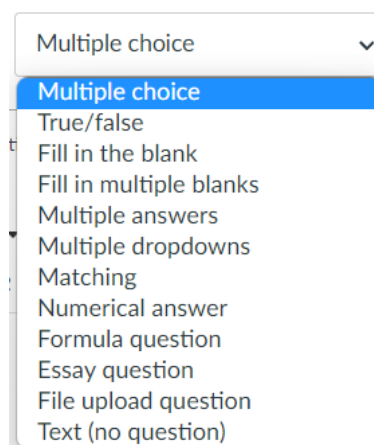


Figure 4.1: Quiz types in Canvas.

Vocabulary has a very important role in the understanding of foreign language texts and utterances, as explained in Section 1.2. Not only is it possible to learn lexical items with the help of computers, but it is also proven that the use of computers and CALL technologies have a positive effect on the development of lexical competence. The study conducted by Sharifi et al. (2015) revealed that students who used a computer-assisted vocabulary learning tool (the Rosetta Stone educational software) learned and remembered more vocabulary items than the control group. As a consequence, using a multimedia application has a positive impact on learning new words, in addition to being more entertaining and enjoyable, as the participants of the study confirmed.

Mobile-Assisted Language Learning (MALL) is a similar field to CALL, in a way more specialized, since it researches mobile technology use in language learning and how the use of mobile phones can enhance SLA. An example of a MALL application is the so-called `apPILcation` developed by Bobály et al. (2020), which was created to facilitate the learning of Mansi vocabulary items. Mansi is a small Uralic language with only 1,346 speakers (Russian Federal Service of State Statistics (Rosstat) 2021). This application offers a vocabulary practicing module and a thematic word guessing game, and is available on Android smartphones at the time of writing.

The creation of exercises and applications to help learners practice foreign languages would still be time-consuming and require the tedious work of a language instructor. This work can be facilitated by language technology algorithms and digital resources, resulting in a combination of NLP and CALL, which is described in the next section.

4.2.2 NLP-enhanced CALL

NLP methods and tools can improve the accuracy and efficiency of language learning software. On the one hand, certain tools and products of NLP (such as spell checkers or corpus look-up tools) can be used in order to improve specific skills of language learners without drastically modifying the program. On the other hand, NLP methods can be more specifically applied in a language learning setting. For example, it is possible to automatically generate personalized practice exercises and quizzes based on the learner's current proficiency level and areas of difficulty. Tutoring systems can support and provide automatic feedback on certain aspects of language (e.g. grammar, vocabulary, sentence structure).

Antonsen et al. (2009) presents a set of CALL programs named OAHPA!, which are based on three NLP tools, providing six language learning modules in total. First, this program was implemented for the Northern Saami language, which was followed by other Saami languages. The same framework was used by Uibo et al. (2015) for Estonian and Võro (which is also a Uralic language spoken in Estonia). To develop the same system for a new language, three core components are necessary: a morphological analyzer/generator, a finite state transducer, and constraint grammar rules.

Another example of NLP-enhanced CALL applications is the Revita platform created by Katinskaia et al. (2018). This tutoring system supports several languages, such as Finnish, Russian, and Swedish, as well as several endangered minority languages, including Komi-Zyrian, Udmurt, and Meadow Mari. One unique characteristic of this system is that it targets learners at the low-intermediate to advanced proficiency level, instead of the beginner level that the majority of CALL applications address.

It is imperative that none of these systems replace teachers and language instructors. The computer should be considered a support tool (Katinskaia et al. 2018), an aid to facilitate the language learning process with the available tools and programs. This technology is able to offer new opportunities for better language practice.

4.3 Limitations of CALL

Despite all the benefits mentioned above, CALL (as well as NLP-enhanced CALL) exhibits some limitations. In this section, a non-exhaustive list of challenges and pitfalls of these fields is provided.

Firstly, as was demonstrated in Section 4.2, theories and dominating paradigms have been constantly changing in the history of CALL, determining the types of activities and programs to be developed. This infinite transformation does not allow the applications to fully integrate into the practices of different institutions and teachers. As a further problem, it is impossible to report on the long-lasting effects of these systems, since there are not many longitudinal studies examining the benefits of CALL methodologies.

A second limitation of this field can be mentioned regarding the feedback these systems can provide. Both positive and negative evidence were supported and endorsed by the behaviorist model. There exist some applications that do not provide feedback to the learner at all, which does not seem

to be in line with what Chapelle (1998) summarized as the ideal conditions for SLA. According to her, learners need to notice errors in their output (either by internal or external feedback) and correct them.

Additionally, it is important that the system only returns correct (true positive and true negative) feedback to the learners. There are certain exercises and exercise types in which Multiple Admissibility is possible, which means that multiple alternative answers can be accepted as correct solutions. This phenomenon is analyzed in relation to a CALL tutoring system in Katinskaia and Ivanova (2019). In case the system is not capable of dynamically detecting alternative correct answers, great care must be taken to avoid producing incorrect feedback. False positive error detection can lead to the discouragement of the learner. As a consequence, corrections must be of very high precision, which was also highlighted in Katinskaia and Yangarber (2021: 135):

Providing feedback to the learner is difficult, due to the critical requirement of very high precision — providing incorrect feedback is much more harmful than no feedback at all.

Faltin (2003: 142) calls attention to the distinction between error diagnosis and error correction.

In CALL, in order to provide appropriate feedback, error diagnosis is more important than error correction. Indeed, knowing how to correct one's own errors based on appropriate information is part of the learning process.

Instead of giving the learners the correct solution to a task, a better approach according to her is to give binary feedback and only indicate whether the answer is correct or not.

Besides the design of the CALL system, the accuracy of NLP tools can also affect the correctness of the provided feedback, in case they are utilized. Since NLP tools cannot yet provide completely accurate results, the learners must be warned about the risks of using NLP-enhanced language learning applications, in which feedback is automatically generated. According to Faltin (2003: 151), NLP tools are useful for CALL, but it might also cause confusion if users “rely completely on the NLP tools without being critical of them”. To summarize, many studies have found that language learning tools and materials have risks if used inappropriately and without much consideration. Hence, informing and educating the prospective users of an application is a key factor when proposing a publicly available tool.

Apart from these limitations, the development of customized CALL software that best fits the needs of language learners can be expensive and require specialized technology and infrastructure, which may not be readily available or affordable in all language learning contexts. It is still unclear what factors predict successful and fast L2 acquisition, as SLA and SLT do not provide a unanimous theory about the optimal, “perfect” way of learning a foreign language. Most importantly, CALL applications – in their present state – cannot completely replace in-person activities and language instructors.

4.4 Conclusion

In this chapter, an introduction to the field of Computer-Assisted Language Learning was provided. CALL is a broad field that aims to support language learning processes with the help of computer technology. It encompasses activities and tasks such as the development of language material, assessment of language proficiency, and teacher training.

NLP-enhanced CALL is a specific area of CALL (presented in Section 4.2.2) in which language processing tools and software are employed in order to further facilitate the generation of learning materials and resources, among others. With the help of NLP methods, it is possible to automatically create certain types of exercises and provide feedback to the input of learners. Nowadays, computers are somewhat integrated into the curriculum of language classes, as was demonstrated in Section 4.2.1: the Canvas LMS system is utilized by language instructors at Eötvös Loránd University to provide learners with additional activities and exercises that they can use to practice the language beyond the classroom. Several examples of language learning platforms were introduced in this chapter, and these are often supported by NLP technologies. There exist many applications that incorporate FU languages and provide exercises to improve language skills in languages with complex morphology. However, surveys regarding the difficulties of learners of Hungarian and Finnish, which will be described in Section 5.4.1, reveal that the existing tools built for these languages do not take into consideration the grammar-specific challenges that L2 learners face. One of the greatest benefits of NLP approaches is that they can alleviate the manual efforts required when constructing language learning exercises.

In order to provide a language learning application that learners of either Finnish or Hungarian can use to enhance their language proficiency, and boost their performance in challenging areas of

these morphologically rich languages, the advantages of CALL, as well as its limitations mentioned in this chapter must be paid close attention to.

The CALL application that was developed during this research will be presented and discussed in Section 5.4.

5 Finno-Ugric Lexical Resources

5.1 Introduction

As discussed in Section 1.2.3, there is a definite gap regarding FU bilingual dictionaries. Maticsák and Laihonen (2011a) emphasized the need for continuously updating and revising dictionary material and creating high-quality resources that contain up-to-date information about the languages in question. These languages may have a certain amount of monolingual data freely available, and resources, such as bilingual dictionaries, to and from English and other well-resourced, popular languages (like German, Spanish, Chinese or French), but not with other (similarly less-resourced) languages.

Learning the grammar of a morphologically rich language is in many aspects a complex, and laborious task. Even if the native language of the learner is considered to be a (mostly) agglutinative language, understanding and memorizing the different morphological rules can be arduous. Most words taken from an arbitrary sentence have complex morphological structures both in Hungarian and Finnish. The production of grammatical sentences, the use of correct inflection in certain contexts, and the difficult rules of attachment of such suffixes to the appropriate form of the root may be facilitated and practiced by grammar exercises, as it is common practice and part of the curriculum of not only Finnish and Hungarian language courses but also in case of many other highly inflected languages. These exercises enable the learners to improve their skills regarding verb inflection, noun and adjective declension, as well as several other language-specific phenomena.

The previously presented data sets are stored in a MariaDB-based relational database as presented in Chapter 3. This, however, does not provide a user-friendly interface to the data, and giving users direct access to the database itself can pose security risks and unwanted and accidental data loss. Since language learners (in this case especially learners of Finnish and Hungarian) probably do not know how to query a database and communicate with it through SQL commands, a user-friendly interface would facilitate access to the valuable data collection. Besides these two functions, the interface also allows the validation of many kinds of data sets from the database and it serves as a DWS. This component was already presented in Section 2.6.2.

In order to utilize the extracted bilingual proto-dictionaries and reap all possible benefits they

can provide, an online framework has been created. It is based on the database that was described in Chapter 3. This framework has three modules: it consists of a DWS (Section 5.2), a Finnish–Hungarian–Finnish dictionary (Section 5.3), and a language learning application (Section 5.4). The name of the framework is Finno-Ugric Lexical Resources (FULR).

5.1.1 User Profile

In every dictionary project – just like when designing any other kind of product or service –, it is vital to give a clear definition of who the intended users of the planned dictionary are. Atkins and Rundell (2008: 28) outline several categories and questions along which the typical users of a dictionary can be characterized. The items of the following list have to be thoroughly considered in order to make well-informed decisions about the content and the presentation of the dictionary.

- Types of users: Will the users of the dictionary be...
 - adults or children,
 - native speakers or language learners,
 - general users or specialists,
 - using the dictionary in an educational, domestic, or professional setting?
- Types of use: Will the users use the dictionary...
 - for general reference purposes,
 - to study a particular subject,
 - to learn a language,
 - to translate text, or
 - to write essays or reports?
- Users' pre-existing skills regarding linguistic knowledge and familiarity with standard dictionary conventions:
 - Do they know regular morphology?
 - Do they understand abbreviations like *adj*?
 - Do they know how words are pronounced?

- Do they know the International Phonetic Alphabet?
- etc.

The present research work is aimed at providing a useful reference work for (adult) learners of Finnish and Hungarian, who study these languages, since, as was shown in Section 1.2.3, there does not exist a high-quality Finnish–Hungarian–Finnish online dictionary. In this case, it is expected from the prospective users to understand the basic terms of linguistics and be familiar with standard dictionary conventions.

The obtained data can be used for multiple purposes. Not only can it be the backbone of an online dictionary, but also a language learning application can be created from it automatically, which helps learners apply the rules and learned material by giving them exercises for the most challenging phenomena in Finnish and Hungarian.

In the next sections, each of the three modules of this web application will be presented in detail.

5.2 Dictionary Writing System

According to Kilgarriff (2006: 7), a DWS “is a piece of software for writing and producing a dictionary.” This concise definition may not adequately express the full range of tasks that can be facilitated and automated with the help of a DWS if it is set up correctly. It can also relieve lexicographers of needing to pay close attention to details (Abel 2012: 84). The structure, the abbreviations, and the layout (font size, weight, color, etc.) can be predefined in the system which produces the dictionary. This means that the editors of the entries do not have to look up or remember every detail present in the style guide that is “an essential resource in any dictionary project” (Atkins and Rundell 2008: 122), rather, they can focus on the content and lexical data that will be included in the dictionary.

There are some off-the-shelf DWS software, which can be used to create mono- or bilingual dictionaries. In the following section, various systems will be presented and the reason why there was a need for yet another in-house DWS will be given.

5.2.1 Off-the-shelf Dictionary Writing Systems

Lexonomy³⁹ is a web-based, free, open-source dictionary writing and publishing system (Měchura et al. 2017). The advantage of Lexonomy is that it can be edited with a working Internet connection and a web browser, there is no need to install any software or application. However, one of its disadvantages is that it cannot automatically create the reverse side of the dictionary. What it offers to deal with this issue is that the search can be done in the translations of entries.

TshwaneLex⁴⁰, a commercial off-the-shelf dictionary compilation software offers automated lemma reversal (Joffe and de Schryver 2004). Apart from the fact that it is not a free tool, another limitation of this system is that it runs only on Windows or Mac OS, while Linux is not supported.

EELex⁴¹ (Langemets et al. 2010) was created by the Institute of the Estonian Language. It enables authorized users to create their own dictionaries in a web environment. At the time of writing, this system is not accessible to ordinary end-users, and the interface is only available in Estonian, which makes the editing process hard – if not impossible – for lexicographers and dictionary editors who do not speak Estonian.

One of the goals of the present research work is to provide a DWS that merges the benefits of all of the previously mentioned applications and creates an interface to communicate with the lexicographical database presented in Chapter 3. Creating a new DWS was hence necessary because none of the already existing platforms offer an interface where the data sets that have been extracted for Finnish and Hungarian (presented in Section 2.5) can be adequately imported and edited. With the desired benefits in mind, the database has been structured in a way that allows the automatic reversal of the bilingual dictionary. The primary aim of this DWS is to provide an interface that helps lexicographers easily access the contents of the database and manipulate the data in a consistent, efficient way without requiring advanced IT skills and without any applications to be installed on their computers. An important factor that this DWS implements is the simultaneous multi-user access which enables users to edit and modify the contents of the dictionary at the same time as other users do. Another advantage of the system is that it tries to minimize human errors as much as possible. The proposed DWS and the process automation it includes can reduce the risk of human

³⁹ retrieved 10 December, 2022 from <https://www.lexonomy.eu/>

⁴⁰ retrieved 10 December, 2022 from <https://tshwanedje.com/tshwanelex/>

⁴¹ retrieved 10 December, 2022 from <https://eelex.eki.ee/>

errors with the help of prefilled and read-only fields, as well as drop-down lists (providing a set of items for the editor to choose from instead of requesting them to type in the information).

This system is free and easy to use. The interface is multilingual, it is translated into three languages (English, Finnish, and Hungarian) in order to maximize the number of potential editors who can contribute to the validation of the data set without any language barrier. The DWS is accessible via the Internet at <https://fulr.btk.ppke.hu/dws.php>.

It is worth noting that the users of the DWS are not necessarily the same as the dictionary users. The online dictionary is a publicly available platform, which can be used by anyone accessing the website without registration. However, to use the DWS, one must register and have a specific authorization level to execute certain actions on this interface. The authorization level is defined by the user role in the database which must be either ‘editor’, ‘teacher’, or ‘admin’ in the `Users` table, see Section 3.4.6. The expected users of the DWS (henceforth for the sake of simplicity ‘editors’, which does not refer to the role in the database, but to everyone who can access the functionalities of the DWS) do not necessarily have advanced computer literacy skills, nor are they, professional lexicographers. The only requirement for the editors is that they can speak both Hungarian and Finnish, and in order to access the UI, they must have a working Internet connection.

5.2.2 Manual Validation and Dictionary Editing

As described in Section 2.6.2, the manual validation of entities and relations was done using the DWS. The interface can be accessed by (registered) users⁴² with a certain authorization level, and entities, as well as relations can be edited by admins and editors. On the user interface, human validators can check the automatically obtained translation candidates, synonyms, definitions, and example sentences present in the database, and interact with the data in a straightforward, user-friendly way. There are several functionalities that can be used by editors of the system. The most important features and functionalities included in the DWS are:

- entity editing form that validates input data,
- relation validation page,
- displaying already validated entities and relations,

⁴² <https://fulr.btk.ppke.hu/dws.php>

- search form to access data in the database, and
- automatic evaluation of the methods.

Validating Entities As mentioned in Section 2.6.2, the first step is to validate the two separate entities participating in a relation (regardless of the type of relation). If both of the entities are existing words, MWE or correct sentences in the given language in a relation, the relation itself can be validated.

If an entity is incorrect, e.g. it is not an existing lemma of the language or it is an incomplete, ungrammatical sentence, the relations in which it is involved do not need to be checked manually, since they are connecting a non-existent, faulty entity to another. These relations can therefore be automatically ignored and they do not need to be validated by the editors. In order to make sure that the entities are labeled as correct first, a filter is applied to the list of relations to be validated: only those relations can be checked manually, where both entities are already categorized as correct. Then, in the following step, the relation can be validated by making sure the type of the relation, as well as the usage labels, is set correctly between the two entities (which is described in more detail below).

In order to validate an entity, collect all relevant, available information about it, and save it in the database, an entity editing form has been created. This form can be seen in Figure 5.1.

The first five fields (ID, Text, Type of Entity, Frequency, and Language), as well as the last four blocks (Label, Source, Show relations, and Remarks), are always shown in this form regardless of the type of entity. Out of these, the ID, Frequency, Language, Source, Show relations, and Remarks are non-modifiable, read-only fields.

The Source field, which appears in the form of all types of entities, contains the list of methods that extracted the given entity as part of any relation. After this block, a button (“Show relations”) makes it possible to display a list of all the relations in which this entity participates. This is a read-only text field, which helps to confirm the exact meaning of an entity. For example, in the case of homonyms, relations can help identify which term is being edited, and hence, what its correct part of speech is. Let us consider the following example: the Hungarian word *vár* can belong to two part of speech categories: when it is a verb, it means ‘wait’, while if it is a noun, it denotes ‘castle’. When an entity where the text field appears to be *vár* is edited, the editor can verify the exact sense

by looking at its relations, and modify the data of the entity accordingly. Due to the large number of relations that an entity has in the majority of cases, this block is hidden until the “Show relations” button is clicked.

The `Remarks` field shows all the comments that have been made to the given entity (in Figure 5.1, it is empty), and it is only possible to add new remarks to this read-only text block with the help of the field below it (which has the label `Add remark`) and the “Save remark” button.

The screenshot shows the following fields and values:

- ID: 5149
- Text: burgonya
- Correct a typo button
- Type of Entity: lemma
- Frequency: 10.7676
- Language: Hungarian
- WordNet Offset: 07710616; 12897493
- Part of Speech: noun
- Universal Features: [empty]
- Inflection type: tövégi magánhangzót váltakoztató főnevek
- Inflection remark: mély
- Label: standard
- Source: OpusExtractor, WiktionaryParser, WordNet, WordNetConnector, wikt2dict triangulate

Buttons at the bottom: Reset information, Withdraw, Save entity, Submit, Delete entity.

Figure 5.1: Entity editing form in the Dictionary Writing System.

The `Text` field can be edited only if there is a typographical error in the text. To reduce the risk of modifying the text incorrectly (for instance, changing a lemma into a totally different word), the editors first must click the button “Correct a typo” in order to unlock the read-only field, which is supposed to serve as a reminder to avoid making any other kind of correction in this field.

The `Type of Entity` may be changed since the insertion script only recognizes three out of five types of entities. Every entity that can represent a headword is automatically assigned the lemma type unless there is a whitespace character in the text, in which case the type becomes `mwe`. In the case of entities that had been extracted as example sentences or definitions (e.g. the definition section of a Hungarian headword in the Hungarian Wiktionary), the assigned type changes to `sentence`. However, `wordforms` and `affixes` are not automatically detected by this script. For instance, affixes were inserted with the lemma type, because they mostly appear in Wiktionary as headwords, which in most cases act as lemmata. Therefore, this must be modified by the editor to the correct category (i.e. `affix`) using the `Type of Entity` field.

The default value of the `Label` field in the `Entity` table is ‘not defined’. It means that, when a new entity is inserted, it gets that value by default. This must be changed to one of the actual labels from the drop-down list which appears in the form. The complete list of labels can be seen in Table 3.14.

The rest of the fields are displayed in this form depending on the type of entity. Some of them are mandatory when they are shown, some are optional. The list of fields can be seen in Table 5.1 indicating whether the field is mandatory (*), optional (▲), or is not displayed (empty cells) in the form when a certain type of entity is selected. Some fields are not shown (hiding them automatically as the `Type of Entity` field is changed), if the information is not relevant for that type (e.g. the inflection type field does not appear in case of sentences). This behavior is implemented in the system in order to reduce the amount of unnecessary information presented in the form, as well as to minimize human errors whenever possible.

`WordNet Offset` is a read-only field, where the synset identifiers (if relevant) are displayed. The `Lemma` field only appears in the case of word forms. A `wordform` is any non-canonical form of a word, hence, the canonical form (the lemma) must be given in this field.

The `Part of speech` is a mandatory field when the type of entity is `lemma`, `wordform`, or `mwe`. If the part of speech is set to `NONE`, the editor has to assign the correct part of speech tag

Type of Entity \ Field	lemma	wordform	mwe	affix	sentence
WordNet Offset	▲	▲	▲		
Lemma		*			
Part of Speech	*	*	*		
Universal Features	▲	▲			
Inflection Type	▲	▲			
Inflection Remark	▲	▲			

Table 5.1: List of fields that appear when a certain type of entity is edited. Mandatory (*) and optional (▲) fields are indicated.

to the given entity. This task is facilitated by a drop-down list, which shows every possible part of speech tag, in the selected language of the interface (English, Finnish or Hungarian).

`Universal Features` is a text field, where different morphological properties can be described about lemmata and word forms. The editor must type in the properties that are relevant to the entity, which makes it more error-prone than a pre-defined set of all possible elements. However, there is a function that validates the format of the field that runs every time a change happens. This function is created to avoid typographical errors and it ensures that the format of the field matches the rules set by the Universal Dependencies Guidelines⁴³.

The `Inflection Type` and `Inflection Remark` fields give more information about the inflection of lemmata and word forms. These fields are implemented as drop-down lists to ensure consistency and facilitate the work of editors.

Merging Entities The entities present in the database have been extracted from many resources using different methods. Two of these methods (the two functions of the `wikt2dict` method: `extract` and `triangulate`) do not assign part of speech information to words, which resulted in duplicate records in the `Entity` table. Oftentimes, there exists more than one entity with the same `text` field, one of them having a specific part of speech tag (such as `NOUN`), while the part of speech of the other one is undefined (`NONE`) since this information is not known at the time of insertion. When the terms denote the same concept and the reason for having two entities in the database is the lack of part of speech information, these entities must be merged. To merge them and

⁴³ retrieved 6 December, 2022 from <https://universaldependencies.org/format.html#morphological-annotation>

keep only one of them with the necessary, correct, and complete description, the relations in which they appear must also be modified, and the source of both the entities and all relevant relations must be unified in the `Source` table. To tackle this issue, a sophisticated algorithm was created with a user-friendly form in the DWS. When an entity is being edited, the system checks automatically whether there is a duplicate entry in the database with identical fields (except the part of speech information). If there is, a list of the duplicate entities appears in the form, warning the editor that there might be redundant entities that must be merged with the present one. Then, the editor has to investigate and find out whether the entities indeed need to be merged, and clicking the “Merge” button will show the entity merging form, see Figure 5.2. This form shows the relevant fields of the two entities side by side, with radio buttons to choose which information to keep in case of each field as the new, merged entity.

Upon submitting the form, the algorithm automatically updates the fields of the entity that is kept, changes the status of the “old” entity to `merged`, and goes through all of the relations of the “old” entity to change the `entity_id` to the identifier of the new entity. This way, none of the relations are lost, and the data present in the database still accurately reflect the results of the initial data extraction.

Splitting Entities As there can be two (duplicate) entities representing the same concept, there can also be one entity that should be split into two or more entities. If, for example, a lemma entity has actually two different senses, and this affects the information of the lemma in any way, such as that there are different forms in their paradigm, the creation of a second entity is necessary.

The Hungarian lemma *szél* (that can mean ‘wind’ or ‘edge’) is a perfect example where splitting has been utilized. The `Entity` table initially contained only one entity for this lemma, but the inflection of this word depends on its meaning (for instance, their plural nominative form is either *szelek* or *szélek*, respectively).

Splitting means that the entity is duplicated in this database, and at least one of the fields of this new entity differs from the original fields. It is important to emphasize that the unique constraint added to the `Entity` table does not allow the insertion of a new entity that has the exact same fields as an already existing entity. Hence, to split an entity, the editor must choose a field that differs from the current entity. There are three options in the entity editing form (shown in Figure 5.3) to

Finno-Ugric Lexical Resources

Main menu My entities My relations Dictionary Logout

Reset information Merge

ID

740183 434351

Text

őszül

Type of Entity

lemma lemma

Language

Hungarian

Frequency

1.30064 -1

Part of Speech

verb NONE

Universal Features

Inflection type

iktelen alapminták none

Inflection remark

ajakkerekítéses magas

Label

standard not defined

Source

OpusExtractor wikt2dict extract

Relations

translation: harmaantua (402076) translation: harmaantua (434309)
translation: harmentua (434183)

Remarks

Reset information Merge

Figure 5.2: Entity merging form in the Dictionary Writing System.

choose from before pressing the “Split entity” button: inflection, part of speech, or type of entity. In the case of the previous example, the *szél* entity was split by choosing the inflection option. The only difference between the two entities is in the `inflection_type` field, which indicates the difference in their paradigm.

The next step is to divide the relations according to the sense of the participating entities. All



Figure 5.3: Splitting an entity in the entity editing form.

the relations in which the initial entity appears are displayed, and the editor needs to select the ones where the sense is that of the new entity (regarding its inflection type or part of speech that has just been assigned). In the case of the *szél* entities, some example relations are shown in Table 5.2. Assuming that the initial entity had the inflection type that applies to the sense of ‘edge’, the relations where *szél* participates with the meaning of ‘wind’ have to be selected, as they will be moved to the new entity.

Relation	szél 1	szél 2
szél – reuna	✓	
szél – tuuli		✓
szél – levegőmozgás		✓
szél – perem	✓	
szél – A lap szélére írta a megjegyzéseket.	✓	
szél – Fúj a szél.		✓

Table 5.2: Relations can be assigned to exactly one sense of *szél*.

New records are inserted into the `Source` table to indicate that the newly created entity has been extracted from the same sources as the initial entity. The reason why it is an important step is that some relations of the initial entity may be moved to the new one (as shown above). This means that both entities were present in these resources, and it was due to the lack of additional information that only one entity was inserted into the `Entity` table.

Validating Relations Manual relation validation consists of the confirmation of the type of relation between the two entities participating in the relation, and – if necessary – the further specification of the usage labels that the two entities are characterized by with regards to their exact sense in that relation.

All the relations that belong to a certain editor are listed in the `My relations` page in the DWS, where relations appear as shown in Figure 5.4. The two entities are clickable links, which

lead to a new page that displays the details of the entity. Where the part of speech information is determined, it is shown as a subscript next to the text of the entity.

The relation-specific labels can be chosen under the entities from drop-down lists, whenever it is not identical to the label of the entity. Next to the two entities, the type of relation appears. The correct relation type can be easily chosen from the list of potential relation types. It is possible to add remarks to the relations, too, similarly to the remarks that can be added to entities.



Figure 5.4: Relation validation page.

5.2.3 Results

The results of the manual validation of entities and relations were introduced in Section 2.6. Since the specifics of the database were presented after that, in Chapter 3, in this section, a more detailed evaluation can be provided.

The detailed results can be seen in Table 5.3. Precision is provided for every language and type of entity separately, along with the number of entities that were validated. Note, that this table is a more detailed version of Table 2.9.

Language	Type of Entity	# of Entities	Precision
Hungarian	affix	7	100.000%
Hungarian	lemma	549	99.818%
Hungarian	mwe	54	100.000%
Hungarian	wordform	3	100.000%
Hungarian	sentence	854	91.452%
Finnish	lemma	497	99.799%
Finnish	mwe	23	100.000%
Finnish	sentence	425	92.941%

Table 5.3: Detailed results of entity validation.

Figure 5.5 visualizes these results, and it is conspicuous that sentences are far less accurate than lemmata or other types of entities. Nevertheless, the lowest precision is 91.452% which is yielded

by the Hungarian sentences.

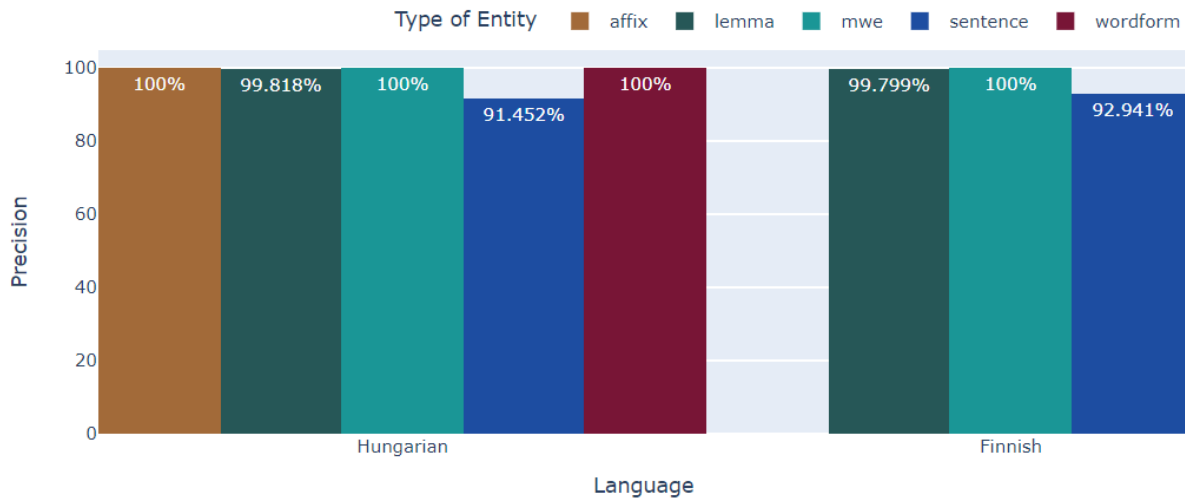


Figure 5.5: Bar chart of the entity validation results.

The few number of affixes and word forms in Hungarian can be explained by the fact that the database was initially populated with lemmata, MWE, and sentences. Therefore, when the given type failed to adequately describe the entity, the modification of the type of entity was conducted. The `affix` entity type was added after it had been observed that several entities could not be described by the initially planned 4 types (`lemma`, `wordform`, `mwe`, `sentence`). Keeping this kind of information in the database may be useful for languages with inflectional morphology since these affixes can be added to a word and change its grammatical function (sometimes even its sense). During the validation, 7 affixes and 3 word forms were identified that were inaccurately labeled as lemmata. However, the correct type has been assigned to them by the editors.

The evaluation of algorithms regarding the relations was introduced in Section 2.6.2. Here, a united table summarizes the attained precision for each dictionary building method and type of relation (see Table 5.4).

The most precise translation candidates have been obtained by the `Wiktionary Parser` method. Synonyms were only extracted from the two wordnets, with around 70% precision. The best-quality definitions are obtained from the Hungarian WordNet (the Finnish WordNet did not contain definitions). On the other hand, the `Wiktionary Parser` tool performed better regarding the example sentences. The `WordNet Connector` algorithm was less successful with this type of data due to the construction of the Hungarian WordNet. In this resource, there is at most one

Method Name	Type of Relation	# of Validated Relations	Precision
WordNet Connector	translations	881	72.645%
Wiktionary Parser	translations	261	98.851%
OPUS Extractor	translations	408	93.137%
wikt2dict extract	translations	258	98.062%
wikt2dict triangulate	translations	617	72.609%
WordNet Connector	synonyms (Finnish)	279	72.043%
WordNet Connector	synonyms (Hungarian)	307	69.707%
Wiktionary Parser	definitions (Finnish)	211	88.679%
Wiktionary Parser	definitions (Hungarian)	234	80.342%
WordNet Connector	definitions (Hungarian)	213	96.714%
Wiktionary Parser	example sentences (Finnish)	211	87.678%
Wiktionary Parser	example sentences (Hungarian)	200	74.000%
WordNet Connector	example sentences (Hungarian)	241	56.017%

Table 5.4: Evaluation of methods based on the validated relations.

example sentence provided for each synset, regardless of the number of lemmata in the synonym set. In the worst-case scenario, this example sentence illustrates the usage of only a single lemma of the whole set. However, the `WordNet Connector` method connects each of the lemmata with this sole example sentence that belongs to the synset, which leads to many imprecise results.

Automatic Evaluation and Data Visualization Thanks to the structure of the database, with the help of PHP functions and SQL queries, it is possible to evaluate entities and relations automatically and to visualize the information with the help of interactive charts. A data evaluation and visualization page have been developed to demonstrate the results of the manual evaluation and to provide some statistics about the extracted data and the database.

The information displayed on this evaluation platform is regenerated and recalculated every time the page is refreshed, hence, it always presents an up-to-date analysis of the current state of the database. The following kinds of information and statistics are available on the evaluation page:

- number of correct and incorrect entities, and the attained precision for every language and type of entity,
- number of correct and incorrect relations (translation pairs, synonym pairs, lemma–definition pairs, and lemma–example sentence pairs) and the attained precision for the applied methods, and

- bar charts provided for three types of relations (translations, definitions, example sentences).

An example of the bar charts (which demonstrates the precision of the definition relations) that appear on the evaluation page can be seen in Figure 5.6.

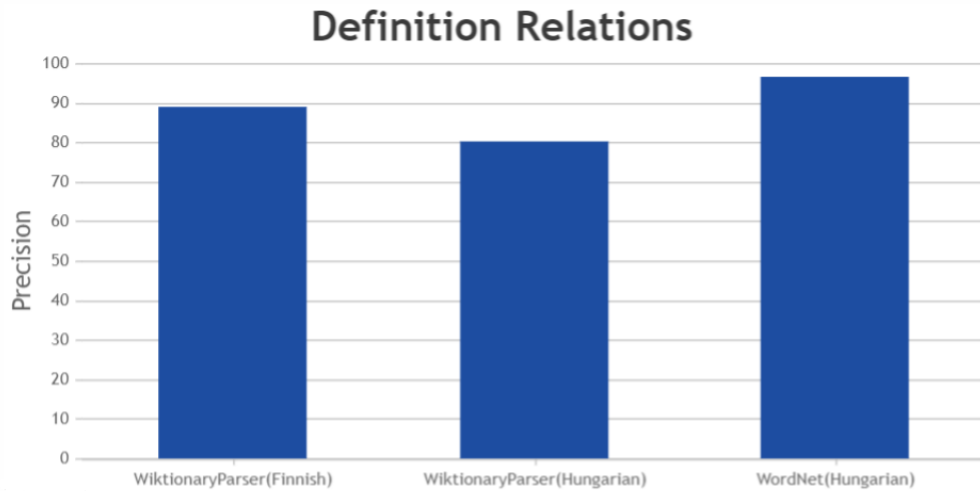


Figure 5.6: Bar chart example in the evaluation page.

5.2.4 Future Work

The DWS presented in the previous sections has many features, and it facilitates the work of editors and lexicographers who validate the translation pairs and other lexical information in the database. However, there are possible ways to improve this application. Possible lines of future development regarding the DWS are pointed out in this section.

The automatically extracted data was used to populate the database. However, living languages are constantly undergoing change, and the meaning of words can change over time, as well. Neologisms may also appear and fall into mainstream usage. To follow the evolution of languages and ensure that the dictionary contains up-to-date information about the vocabulary of these languages, editors should be able to add new entities and create relations in the DWS, apart from being able to modify already existing ones.

Entity creation must be preceded by ensuring that the entity does not exist in the database yet. After inserting a new record in the `Entity` table, the entity can be edited by the editors using the previously presented entity editing form. However, in order to insert this new element into the

network of existing relations, another module must be developed. A user-friendly interface should enable editors to connect existing entities by creating new relations, and to choose their type of relation. To date, adding new entities and relations to the database is only possible with the help of SQL statements. Creating a user-friendly interface that does not require advanced IT knowledge is left for future work.

Human error is natural, and as such, it can be expected when using the interface of the DWS. One of the tasks where such an error could happen is the selection of those relations that describe the sense of the newly created duplicated entity, when splitting an entity. Relations can be chosen accidentally, and relations can be left out unintentionally. These errors must be corrected by replacing the incorrectly chosen entity with the other entity (that is the result of the splitting) in the erroneous relations. This function must be restricted to those relations and entities, where the entity has been split, as the modification of any other entities participating in a relation would be undesired since it would distort the results of the evaluation of algorithms.

5.3 Dictionary

To access the information present in the database, an online bilingual dictionary was created. This dictionary is publicly available at <https://fulr.btk.ppke.hu>. It is carefully designed with its prospective users in mind. The target audience of the dictionary has been defined in Section 5.1.1.

Apart from the intended users of the dictionary, its macrostructure and microstructure are also key concepts, which determine the content of the dictionary and the structure of the entries. Macrostructure refers to the organization of the headword list in a dictionary. Microstructure, on the other hand, specifies the structure and the different components of the entries. In the next sections, these two aspects of the proposed dictionary will be presented.

A huge advantage of online dictionaries is that the interface can be dynamic, and customizable and the information displayed in the entries can be defined by each user to their own liking. For instance, the language of the interface can be chosen (and modified at any time) by the user to provide information in a language that the user is most comfortable with.

5.3.1 Macrostructure

In a traditional, print dictionary, the most frequently used organization is the alphabetical order of headwords. Users need to know the exact order of letters in a language to be able to find the word they want to know more about.

In online lexicons, the sequence of headwords is not relevant anymore. The way information is stored varies from dictionary to dictionary, and the method how users access different entries does not depend on how the items are stored. Instead, most frequently, entries that match the search conditions given by the user are shown. However, new questions arise with regard to online dictionaries: what parts of the entries are searchable? How can users find information that they are interested in? How can dictionary designers make it easier and more straightforward for dictionary users to find the appropriate, relevant entries?

Online dictionaries can employ many strategies to direct users to the right entries. While in a paper dictionary, the users need to look for the right headword, digital technologies transform the look-up process and allow the users to browse the data set in new ways. The list of components within an entry that users are allowed to search for is determined by the designers of the dictionary.

Users are most generally allowed to search for headwords (similarly to the way paper lexicons work), with the difference that they can type in the lemma of the word they are looking for. Sometimes, the searching algorithm is more sophisticated and allows the users to look for a part of the word, or any word form, not only its canonical form, or lemma. In this new setting, more advanced search techniques can be implemented which help users find lexical information more quickly and effortlessly.

In the proposed dictionary, a detailed search module is provided to the users. With the help of this feature, it is possible to filter the list of entries based on many conditions. The search bar of the dictionary is shown in Figure 5.7. It must be noted that the language of the dictionary interface can be chosen by the user, as previously mentioned, and in the following examples, this language is set to English. The language of the interface determines the language used in the search bar and in certain parts of the entries, e.g. the part of speech tags. Apart from the headword (first text field), the language (second field) and the part of speech information (fourth field) can be searched and filters can be applied to them. It is possible to try to match the beginning, the end, or any part of

the headwords or to find headwords that match exactly the given search string. This can be defined in the third field of the search bar, which is a drop-down list with the above-mentioned 4 options.

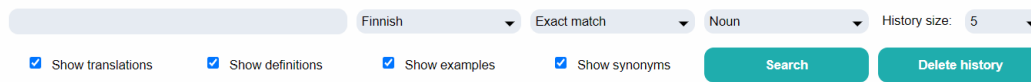


Figure 5.7: The search bar of the dictionary.

The search word can contain wildcard characters: the percent sign (%) matches zero or more characters, while the underscore (_) matches a single character. This allows users to look for headwords with certain patterns. For instance, providing the pattern ‘_t_ %’ as a search string could match headwords whose second character is *t* and contains at least 3 characters in total.

5.3.2 Microstructure

Each entry in a dictionary follows a specific structure. This structure determines, among others, the exact place of the components of the entry (Atkins and Rundell 2008). The organization of the elements of entries can be referred to as the microstructure of the dictionary.

One of the many benefits of electronic dictionaries is that they can be customizable. To let users of the proposed dictionary personalize and customize the online dictionary layout, different parts of the entry can be shown or hidden (see Figure 5.7 above). If, for instance, a user wants to hide the definitions, the “Show definitions” checkbox can be unchecked, and the definitions will not appear in any of the entries.

The microstructure of the dictionary (in case all of the components are to be shown) can be seen in Figure 5.8.



Figure 5.8: Microstructure of dictionary entries.

Next to the headword (*vasen* ‘left’), the part of speech information is provided in the language

of the interface (*Adjective*). Below the headword, the translations appear (*bal* in Figure 5.8). Under the translations, the definitions are shown in bold. The example sentences appear below the definitions (in italics). Synonyms are listed at the end of the entry (following the ‘synonyms’ label in the language of the interface). The translations and the synonyms are hyperlinks leading to the entries of the given entity. This allows fast navigation between entries, and for the same purpose, the history of previous searches can be also seen under the search bar. This enables dictionary users to go back and forth between previously searched words and specific entries. The size of the history can be modified in the search bar (in the range of 1 to 10).

To the right side of the part of speech tag, a number indicating the inflection type appears, whenever it is provided in the database to the given entity. The user can hover over this number, which will display the example word used in the Kotimaisten kielten keskus (Kotus) inflection types for Finnish words. This number is a clickable hyperlink, which opens up the Kotus website⁴⁴ where these inflection types are illustrated. Consonant gradation also appears in the case of some Finnish words where it is relevant, displaying a letter next to the inflection type. This is also a hyperlink leading to the Kotus page which describes the different consonant gradation types⁴⁵. In the case of a Hungarian headword, instead of the number, a small icon appears. The inflection type and the vowel harmony type appear by hovering over the icon (see Figure 5.9). Clicking this icon opens the inflection tables in the E-Szókincs⁴⁶ webpage.

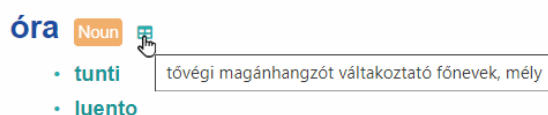


Figure 5.9: Hungarian inflection information when hovering over the icon.

In case an entity has more than one sense, they appear below each other. To illustrate this, a more complex entry is shown in Figure 5.10. The word *kertoa* has two senses (‘tell’, and ‘multiply’). The second meaning illustrates how usage labels appear in front of the translation when the usage label of the entity in a certain sense differs from the default usage label. The consonant gradation category (K) appears after the inflection type (52) next to the part of speech tag. In this figure it

⁴⁴ retrieved 24 September, 2022 from <https://kaino.kotus.fi/sanat/nykysuomi/taivutustyyppit.php>

⁴⁵ retrieved 27 September, 2022 from <https://kaino.kotus.fi/sanat/nykysuomi/astevaihtelutyypit.php>

⁴⁶ retrieved 29 September, 2022 from <http://corpus.nytud.hu/cgi-bin/e-szokincs/alaktan>

can be observed that the definitions are listed below the translations, they are not connected to any of the senses specifically. To improve the quality of the dictionary in this regard, part of the future work is to connect relations to other relations in the database. In this specific case, the relation that defines the sense of *kertoa* by connecting it to *szoroz* ‘multiply’, shall be linked to the relation of *kertoa* and its relevant definition (*Tehdä kertolaskutoimitus.*) which describes the same concept (i.e. ‘multiply’).



Figure 5.10: Microstructure of dictionary entries with more than one sense.

Each entity in the database is shown as a different headword in the dictionary. Hence, the primary cut of the proposed dictionary is on the basis of grammar, rather than meaning (Atkins and Rundell 2008: 247). Homonymous words that belong to different word classes are represented by multiple entities in the database, therefore, they appear as homograph headwords (see Figure 5.11). The same applies to entities where the difference is solely in the paradigm of the words, when the inflection type is the only difference between them (as it was the case for the two Hungarian homonymous entities *szél*, for more details, see Section 5.2.2 above).

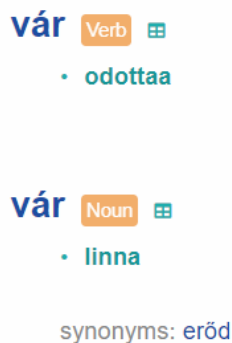


Figure 5.11: Homograph headwords are displayed as separate entries in the dictionary.

The order in which translations appear in the entry is also an important part of the microstructure. There are three ways how the order of dictionary senses can be determined within the entry

according to Atkins and Rundell (2008: 250). These are:

1. historical order,
2. frequency order, and
3. semantic order.

Historical order requires adequate information about in what order the senses entered the language and developed since then. In the present work, such information is not provided.

Senses can be ordered by their frequency. Language learners are most likely looking for the most common sense of words first, and therefore this may prove to be a good technique in the case of learner's dictionaries.

Semantic order means that the core, psychologically salient meaning comes first, followed by the senses that are semantically the closest to this one, and so on.

Most dictionaries use this latter option when it comes to ordering the senses in an entry. In the proposed dictionary, however, it is not possible to manually decide and assign a specific order to the translation equivalents, since the entries are generated automatically, and the relations are independent of one another. To tackle this issue, readily available information is exploited in the database: the frequency of the translation equivalents. Ordering the translations according to their frequency is very similar to the second possibility of how senses can be ordered described by Atkins and Rundell (2008). The only difference is that the frequency of senses is not measured on the corpus of the source language, but it is provided by the number of occurrences of the target language equivalents in a target language corpus.

This allows learners to encounter the most frequent translations first (most probably what they are looking for), while less frequent equivalents are at the bottom of the entry. For every entity, the translation relations are selected from the database. The translations are then assigned to "sense groups": each group describes a concept, similarly to a synset in a wordnet. It is carried out automatically by utilizing the synonym relations in the database. Synonymous words are assigned to the same group. Simultaneously, the maximum frequency among the elements of the sense group is assigned to the group, and the order of the sense groups is determined by these frequencies. The group of translations which contains the lemma with the highest frequency appears on top, followed

by groups with less frequent lemmata. The translations within a group are ordered alphabetically, as shown in Figure 5.12.



Figure 5.12: The order of senses is defined by their frequency, while translations within one sense appear in alphabetical order.

5.3.3 Automatic Entry Creation and Formatting

One of the major advantages of the proposed DWS and dictionary interface is that editors do not need to craft entries from the existing entities: the code base of the dictionary takes care of the automatic composition of entries. The automatic entry creation and the structure of the data model make it possible to reverse the bilingual dictionary without manual editing. The formatting of dictionary entries is also automatic: the layout and design are applied by the system, editors are not required to arrange the contents of the entries and define the microstructure by hand. Not only does it reduce the risk of human error, but it also facilitates the work of lexicographers, by letting them focus more on the content, rather than the design aspects of the dictionary. Editors only need to validate and correct the entities, as well as the relations between them, and the structure of each entry is automatically created according to the predefined rules and styling. The content and the presentation of the dictionary are entirely separated. This means that the structure of the components and their design can also be modified easily, without needing to change or adjust the content. Furthermore, additional interfaces can be created on top of the same database, to satisfy the needs of other kinds of user groups and create dictionaries for an entirely different audience.

5.3.4 Future Work

Regarding the dictionary interface, there are many minor changes and developments that can be considered as potential directions for future work.

First, the search function in the dictionary interface can be further improved. When the user looks for different word forms, instead of the lemma of words, the system should lead to the appropriate entry with the help of a lemmatizer of the given language. To date, the dictionary does not automatically conduct lemmatization on the search words, hence, when users try to look up inflected word forms, no entries appear in the dictionary. The text field where the user types in the search word can also give suggestions by providing a list of words that match the string that has already been entered.

Secondly, the interface can be made even more customizable. Some of the components can be hidden if the user does not need them (e.g. definitions, synonyms), but the inflection type, as well as the part of speech tag and the usage labels, is always present in the entries. Furthermore, when certain fields are disabled by the user, there are some headwords that appear without any additional information. It could be also optional whether to show these “empty” entries or hide them in case they do not have the desired relations.

Finally, as already mentioned, connecting relations to other relations would make it possible to link definitions and example sentences to the exact sense of the headword, by placing them under the appropriate list of translations. To date, definitions appear after all of the translations, and example sentences follow the list of definitions. To create relations between relations, the DWS should have a suitable interface for this task, while the structure of the dictionary entries should be reorganized to display the information correctly. The database has been designed with this feature in mind, hence, there are no necessary steps to be taken concerning the database schema in this regard.

5.4 Computer-Assisted Language Learning Application

The complex morphology of Finnish and Hungarian poses a great challenge for L2 learners. The different morphophonological alternations happening in these languages (e.g. consonant gradation in Finnish and vowel epenthesis in Hungarian) further complicate and slow down the learning process. Therefore, language learners are often encouraged to practice the target language beyond the classroom. Many instructors recommend and encourage students to use different platforms (such as Quizlet, Memrise, or in-house exercises prepared by the instructor). Since the dictionary proposed in this work was primarily designed with language learners in mind, and the extracted data can be used as a resource for various CALL exercises for the same target audience, two kinds of language

learning modules have been developed during this research (Ferenczi 2022).

5.4.1 Motivation

To master languages with a rich morphological system always requires lots of time and effort. The learners must be able to understand the information encoded in different word forms of the same root and generate the correct word form to express certain syntactic functions and grammatical relations by conjugating a verb or declining a noun, an adjective, or a pronoun. When the structure of one's native language is far from that of Finno-Ugric languages, it is even harder to understand the necessary concepts. As Laakso (2015: 183) puts it: "it is obvious that the rich morphology of the Finno-Ugric languages challenges the learner, especially students who have earlier only studied languages like English". However, native speakers of Hungarian have also difficulties when learning Finnish, and vice versa. As mentioned in Section 1.2, one of the greatest obstacles (pointed out by Laakso (2015)) is vocabulary when learning Finnish or Hungarian with the other language being the mother tongue of the learner (i.e. Hungarian or Finnish, respectively).

Another obstacle that can be mentioned is the complexity of grammar and the huge number of cases present in these FU languages. One way to enhance one's language skills is through exercises that focus on certain aspects of grammar. When one concept is understood and practiced thoroughly, the learner can go on to master more complex parts of the grammar.

Finnish and Hungarian possess some linguistic and grammatical characteristics – besides their extensive case system – that make language learning even more difficult for FFL and HFL learners. In the case of Finnish, such characteristics include for example consonant gradation and the three cases that the grammatical object of a sentence can appear in. One important example of grammatical agreement in Hungarian – which is quite rare among the world's languages (Durst and Janurik 2011) – is the existence of two separate morphological paradigms: the definite and indefinite conjugation of verbs.

According to the survey of Karlsson and Chesterman (2008), the difference between the vocabulary of Finnish and that of any Indo-European language is one of the many surprises that FFL learners have to face when trying to master Finnish. The possible cases that a Finnish object can appear in also cause some confusion. Since the rules, that define which case is used in a certain situation, are quite complex, it takes some time and practice to really understand their correct usage.

Differentiating between the three past tenses, that Finnish has, also requires a lot of attention, as does understanding how the passive construction is formed and when it is used. What makes it even more difficult to understand is that this construction diverges from the “prototypical” passives that the learners might have encountered when learning other languages like English or German.

Korhonen (2012) studied learners of Finnish with different native languages and origins (in total 45 countries, including Hungary). She surveyed what properties are considered to be difficult in learning Finnish by the language learners. The participants filled up a questionnaire, and the results of native speakers of Hungarian are reported here. 7 out of 32 FFL learners with Hungarian L1 found some parts of the sound system hard to learn, while 10 Hungarians said that inflection, Finnish morphology, and consonant gradation are difficult. Another significant observation is that the use of partitive case in Finnish was considered to be difficult to learn by more than half of the Hungarian participants (18 out of 32). Korhonen not only measured what is complicated about Finnish but also surveyed learners’ opinions about what they think is easy. 17 Hungarians thought that pronunciation is a less difficult part of Finnish language learning since it is very similar to that of Hungarian.

Máté (1999) conducted a survey and observed that HFL learners often experience difficulties when they learn about definite and indefinite verb conjugation in Hungarian. Some of the learners also struggle while trying to learn the proper usage and meaning of verbal prefixes, and when trying to understand the possessive construction. Durst and Janurik (2011) found that the similarities between Hungarian and Erzya-Mordvin (a language which also distinguishes between definite and indefinite verb conjugation) probably do not facilitate the acquisition of this phenomenon in Hungarian. Laakso (2015) states that the Hungarian object conjugation and possessive suffixes are exotic in this language, which is completely in line with what Máté (1999) observed.

To help FFL and HFL learners practice these particular aspects of Finnish and Hungarian, a CALL application is proposed. Word cards have proved to be effective in helping learners acquire new vocabulary items (cf. Nation (1980) and Elgort (2011)). For that reason, virtual flashcards were automatically created. Fill-in-the-blank tasks (otherwise known as cloze tasks) that cover the most difficult parts of Finnish and Hungarian grammars (based on the observations of Máté (1999), Karlsson and Chesterman (2008), and Korhonen (2012)) have also been developed using different NLP methods and querying techniques. The flashcards and the examples of the cloze exercises are

generated automatically from the previously obtained Finnish and Hungarian data sets presented in Chapter 2, which will be further detailed in the next section.

5.4.2 Extracted Data

Several resources were used to obtain lexical information for Finnish and Hungarian. These sources and the methods that were utilized were presented in Section 2.5. The details of the extracted data and the database in which the data is stored were described in Section 2.6 and Chapter 3. In this section, a short summary is provided about the collected data, which will be utilized in the CALL application.

The three main resources of data extraction were the Finnish and Hungarian WordNets, the Finnish, Hungarian, and English Wiktionary editions, and the OPUS corpus.

The collection of bilingual proto-dictionaries was executed by three newly proposed methods (WordNet Connector, Wiktionary Parser, and OPUS Extractor), as well as two methods of an already existing tool (`wikt2dict extract` and `wikt2dict triangulate`). Some of the methods also obtained example sentences and definitions to certain headwords. Due to the structure of WordNet, synonyms could be generated and stored in the database.

The data can be utilized to create a language learning application automatically. Bilingual translation pairs can serve as the material for bilingual word cards, while monolingual flashcards can contain headwords on one side and their definition in the same language on the other. The set of example sentences can be utilized to create word formation (fill-in-the-blank) exercises in which the learners have to give the correct form of a masked-out word. Sentences can supposedly provide sufficient context in order to determine what case or person and number one must choose to make the sentence grammatical. This supposition will be further examined in the following sections.

The proposed application will therefore have two modules: a virtual flashcard module (which will be presented in Section 5.4.4) and a cloze tasks module (which will be described in Section 5.4.5).

5.4.3 Accessing the Application

When a user wants to access the CALL application, they first need to create an account on the FULR website⁴⁷. Registering a new account can be done by providing a username, a password, and an email address. When a new user is created in the database, the default role that is automatically assigned to it is ‘player’. To modify this, an administrator has to assign a new authentication level to the user in the DWS interface, on the `Manage users` page. Users are informed and required to consent to the processing of the data which is collected when using the language learning application. Without opting in, the account cannot be created. After logging in, the user is automatically redirected to the language learning application (which is accessible at <https://fulr.btk.ppke.hu/call.php>). In the main menu, there are several options: the users can view their profile, choose to practice grammar (fill-in-the-blank exercises) or vocabulary (flashcards), and there is also a hyperlink to the dictionary to be able to quickly look up unknown words.

5.4.4 Virtual Flashcard Module

A popular way to acquire new vocabulary items or memorize difficult ones is to use so-called flashcards. On one side of these cards, there is in general a new, so far not known item (e.g. a word in the target language that the learner does not know yet). On the other side which provides an explanation of the new word or phrase, there is either the translation in the source language or a target language definition of that item. Therefore, depending on the language of the explanation, flashcards can be either bilingual or monolingual. Elgort (2013) shows that intermediate proficiency learners of English perform significantly better when the vocabulary items are presented with their translation equivalents in the L1 (Russian). However, this observation is not so significant in the case of more advanced learners. Jo (2018) also noticed that learners achieved higher scores on post-tests when the L1 equivalents were used instead of target language definitions.

Considering the medium in which these cards can be created, there are at least two main categories. They can be written on paper that the learners can take in their hands and turn over to check the equivalent in their native language or the definition in the target language. Besides this traditional solution, there exist digital flashcards, that are accessible anywhere and anytime with the

⁴⁷<https://fulr.btk.ppke.hu/>

help of mobile devices (such as laptops, phones, or tablets).

In the web application presented in this work, learners are given two options to choose from as the explanation side of the cards: it can contain the source language equivalent or the target language definition. This serves two reasons: with the help of this information, it will be possible to test the effectiveness of the language of the explanation on the learning process in the future. On the other hand, users are able to choose their preferred kind of flashcards: either bilingual or monolingual, according to their level or their personal preference. After selecting the target language and choosing an explanation option (L1 equivalent or L2 definition), the flashcard module incorporates two modes: in practice mode, the learners need to get acquainted with the new items, they can turn over the cards as many times as needed, while in test mode, the learners' active-productive knowledge (Laufer et al. 2004) is tested. They need to recall the newly acquired items by typing in the expression they just learned, when a given L1 equivalent or L2 definition appears on the screen. The order of the cards that appear during test mode is randomized.

In Figure 5.13, the two sides of a monolingual Finnish virtual flashcard are shown: the target word (*lapsi* 'child') is given on one side of the card, and its L2 definition (*ihmisen jälkeläinen* 'a human descendant') is given on its other side. Note that while on the figure, the sides of the flashcard are shown side by side, on the web interface, the learners can only see one side at a time, and turn it over (make its other side appear) by clicking on the card.

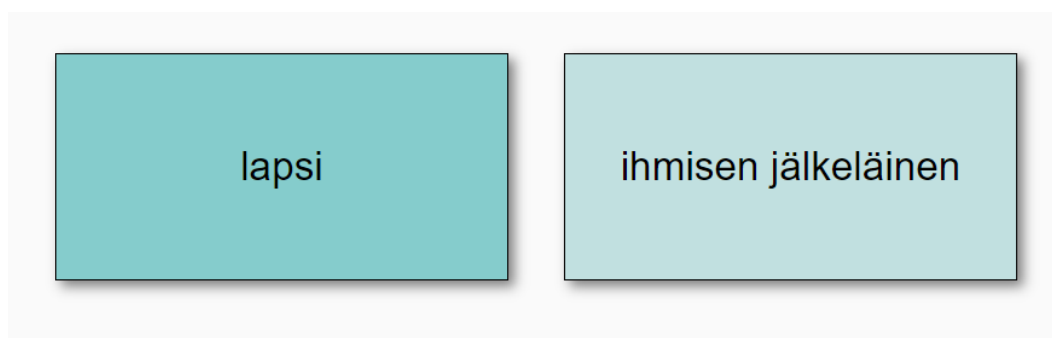


Figure 5.13: Example of a monolingual Finnish flashcard.

In the test phase, when the learners submit their answers, feedback is given immediately by the system. It is highly important to mention that the application only accepts the originally shown word or expression as the correct answer. Synonyms of the target word are not accepted, although they may be potentially correct translations of a given L1 expression.

5.4.5 Cloze Tasks Module

There are several solutions to help language learners practice target language grammar. One of these is to create cloze exercises with automatic methods. The task is to reconstruct the missing parts of a sentence or paragraph of an L2 (target) text. In some cases, the lemma of the missing word is given and the only task is to complete the sentence with the correct word form of this lemma, in other cases, it is entirely up to the learners to guess what lexeme in which form (case, number, tense, etc.) fits most in the context. It is possible to automatically evaluate the performance of the learners after the answers are submitted.

In the proposed CALL application, several aspects of Finnish and Hungarian grammar are considered. Cloze exercises are provided to help learners practice a selected subset of the characteristic features of these languages. For Finnish, the following three grammar phenomena were processed: the different cases in which the object can appear, the three past tenses, and the passive verb conjugation. For Hungarian, exercises were built to help practice definite and indefinite verb conjugation, the usage of verbal prefixes, and possessive constructions. In the following sections, each of these aspects will be discussed in detail, the different rules that allow the automatic selection of the sentences from the database which contain the specific grammatical concepts are provided, and the structure of the exercises are described.

To automatically generate cloze exercises, some kind of target language text or corpus is needed. As mentioned in Section 2.5 and Section 5.4.2, a great number of Finnish and Hungarian example sentences was collected from different resources. These sentences can be utilized in a word formation task where one of the words is hidden and the learners are asked to reconstruct the original sentence by typing in the missing word in the most suitable word form. After observing the data, it was noticed that the sentences vary in length and quality. Since only grammatical, complete sentences should be used in these tasks, a condition is applied to filter out undesired data without manual editing. Only those example sentences are considered in the tasks which contain at least 3 words, which start with a capital letter and end with punctuation (full stop, exclamation mark, question mark, etc.), and which do not contain special characters, such as <, >, = or \$.

These conditions reduce the number of sentences that can be used in the exercises. In the case of Finnish, 18,043 sentences meet these conditions, and in the case of Hungarian, the number is

reduced to 17,450. Manually selecting which sentences can be utilized in a certain grammar exercise would be time-consuming. Fortunately, it is possible to automatize this process. In order to define which sentences can be part of a certain exercise, their morphosyntactic structure needs to be specified. Therefore, the sentences must be tokenized and lemmatized, as well as morphologically analyzed and dependency parsed. To achieve this, three tools were used: the Hungarian `emtsv` pipeline, `omorfi` for tokenization, lemmatization, and morphological analysis of the Finnish sentences, and the dependency parsing was conducted with `uralicNLP` for Finnish. These tools were presented in more detail in Section 2.4. It is also necessary to manually define a rule, a certain pattern for each type of exercise. SQL queries can be used to look for sentences in the database, in which one of these patterns can be found. Then, the obtained subset of sentences can be included in the given task type.

The output of the language processing tools and analyzers is stored in the same database where the linguistic data can be found. However, to avoid the repetition and redundancy of analyzed sentences in the database, new tables are created for the different levels of analyses. These tables are queried during the extraction of the matching sentences for the different types of exercises. The database tables which were created for the CALL application are described in the next sections.

To avoid storing duplicate entries for any identical analysis of the same string appearing in multiple sentences, two tables were created. One of these tables stores the analysis that concerns tokens (`TokenAnalysis`), while the other table connects these analyses of tokens to the sentences in which they appear (`Analysis2Sentence`).

TokenAnalysis The morphological analyses that belong to each token are stored in the `TokenAnalysis` table. This is designed to reduce redundancy and has the structure shown in Table 5.5. Each analysis is stored only once and is identified by an automatically increasing integer, the `word` field stores the actual word form that appears in a sentence. The lemma of the word is saved in the `lemma` field, while its language, part of speech information, and other lexical and grammatical properties appear in the respective fields.

Analysis2Sentence The connections between the words of a sentence and their respective analysis are stored in this table. The structure of the `Analysis2Sentence` table can be seen in Table 5.6.

Name of field	Type	Description	Example
word_id	INT	auto_increment, primary key	2050
word	VARCHAR		gyorsan
lemma	VARCHAR		gyors
lang	INT	foreign key (Languages)	2
upos	INT	foreign key (PartOfSpeech)	2
feats	VARCHAR		Case=Ess Degree=Pos Number=Sing

Table 5.5: The structure of the TokenAnalysis table.

The relation has a unique identifier, the identifier of the sentence (`sentence_id`) is denoted by the integer from the referenced table. The field containing the analysis (`word_id`) also has a foreign key constraint, which refers to the table presented above. Besides these, the word position is given (where 1 means the first word of a sentence) and the dependency information is stored in the `deprel` (dependency relation) and `head` fields.

Name of field	Type	Description	Example
relation_id	INT	auto_increment, primary key	7321
sentence_id	INT	foreign key (Entity)	744
word_id	INT	foreign key (TokenAnalysis)	2050
word_position	INT		3
deprel	VARCHAR		MODE
head	INT		4

Table 5.6: The structure of the Analysis2Sentence table.

Task Generation In this application, language learning (fill-in-the-blank) exercises are generated following the steps described in this section.

A file with JavaScript Object Notation (JSON) format is used in order to create a universal description for types of exercises. In this JSON file, the keys are the names of the exercise types. They are associated with objects, which contain the following properties:

1. `condition`,
2. `checkTense`,
3. `checkNumber`,
4. `checkPerson`, and

5. showLemma.

The rule that is used to obtain sentences with SQL statements is defined in the `condition` field.

The rest of the fields can take boolean values (either true or false), which determine the way the generated tasks appear. For example, if the `showLemma` field is set to true, the lemma of the missing word will appear between parentheses.

Using this JSON file, the application selects the subset of sentences that satisfy the given condition, and the word that matches the pattern is removed from the sentence. It is replaced by an input text field, where learners can enter their answers. To make the exercises as unambiguous as possible, the lemma and several features of the missing word form can be given. For example, since Finnish and Hungarian are pro-drop languages, the subject might be omitted in certain sentences. For verb conjugation tasks, the presence of this feature in the FU languages requires that the person and number be given along with the missing lemma. Otherwise, there would not be only one potentially correct solution for most of the examples.

To make sure that learners are provided with only correct sentences and tasks, the data undergo a two-step manual validation process. This process also serves the purpose of proving the hypothesis (mentioned in Section 5.4.2) according to which a sentence can provide sufficient context to decide what word form is masked out. In the first step, the raw data is checked, as well as the output of the language processing tools. The proto-dictionaries were generated automatically, which also applies to example sentences. Grammatical and typographical errors may occur in these sentences, which must be omitted. The second step is a validation of the created tasks. Only sentences that proved to be grammatically correct in the first step qualify for inclusion in this step. This step is necessary in order to give an objective evaluation of the exercise generation and CALL methods.

The evaluation of the sentences validated in this process can be seen in Section 5.4.6.

Finnish Exercises In the CALL application, three types of cloze exercises were developed that help language learners practice different aspects of Finnish grammar.

Finnish objects can appear in 4 cases: nominative, partitive, accusative and genitive. This poses a big challenge to learners of Finnish, and hence, the creation of a fill-in-the-blank exercise for this specific task is justified. First, a subset of the data is queried from the database using the

JSON file described above, and then, the object is removed from the sentence. The condition in the JSON file for this task can be described as follows. Sentences containing a noun, adjective, pronoun, or numeral with the `DOBJ` dependency tag are selected, when one of the four possible cases (`Case=Nom`, `Case=Par`, `Case=Acc`, `Case=Gen`) can be found among the morphological codes of the object. This word is replaced by an input field, and the learners are asked to put the given lemma in the correct case. The lemma of the missing word is given between parentheses after the input field, as can be seen in Figure 5.14.



Figure 5.14: Example task to choose the correct case for a Finnish object.

Another grammar issue that FFL learners face is the existence of three past tenses in Finnish. These are: simple past (*imperfekti*), present perfect (*perfekti*), and past perfect (*pluskvamperfekti*). The two latter are composed of the auxiliary verb *olla* (in either present or simple past) and the past participle of the second verb. Knowing which past tense to use in a certain environment requires a lot of practice. For this reason, an exercise was created where sentences containing any of the three past tenses appear, while the verb gets replaced by a text input field. The learners need to choose the correct past tense and conjugate the given verb into the correct form. One of the following two conditions must be met by a Finnish sentence to be part of this exercise: it either needs to contain a verb in simple past form (`Tense=Past`), or its main verb has to be *olla* (in either present or simple past tense) and there must be an active past participle form (marked by the `Connegative=Yes` and `Tense=Past` morphological codes) in the same sentence. These words are replaced by text fields on the interface so that learners can reconstruct the correct verb tense. It has been observed that in some cases, two text fields appear because the main verb *olla* and the past participle can separate from each other in the compound tenses. Therefore, another rule is necessary to eliminate these sentences from the query results, since one out of three past tenses (i.e. the simple past) can immediately be excluded when the learner encounters two separate input fields. The aim of the task is to let the learners choose which past tense is the most suitable in the given context. Hence, an additional condition is established to ensure that the two verbs are adjacent in the compound tenses, so that each sentence contains only one text field, not giving any hints about which tense may be correct. Due to the presence of the pro-drop feature in Finnish, the person and number of the subject

are also provided along with the first infinitive form of the verb. An example task can be seen in Figure 5.15.

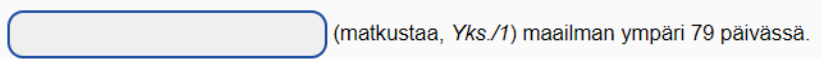


Figure 5.15: Example task to conjugate the verb in the correct past tense in Finnish.

Passive voice is used quite often in Finnish. One of the reasons is that the first person plural form of the present tense indicative is replaced by the passive form in colloquial, spoken Finnish. Correctly conjugating the verbs in this form is therefore essential for language learners. To build cloze exercises that help FFL learners practice this construction, sentences containing a verb in the passive form are selected from the database. This appears among the morphological codes of verbs as *Voice=Pass*. The passive verb is then replaced by a text input field and the first infinitive form of the verb is given between parentheses as can be seen in Figure 5.16.

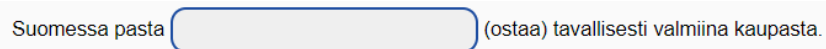


Figure 5.16: Example task to practice passive construction in Finnish.

The Finnish passive has a distinct set of morphological markers in the present and past tenses. To make the exercise unambiguous, these two tenses were separated, and the learners can choose which tense they want to practice in connection with the passive construction.

The JSON file contains the values shown in Table 5.7 in the case of the Finnish exercises. The *condition* field was already described above in the case of each exercise type.

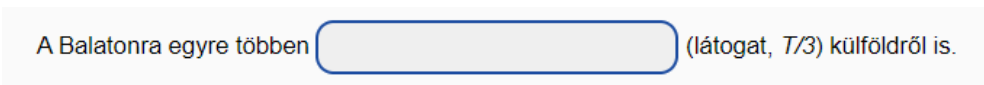
Parameter	Object	Past Tenses	Passive
<i>checkTense</i>	false	false	false
<i>checkNumber</i>	true	true	false
<i>checkPerson</i>	false	true	false
<i>showLemma</i>	true	true	true

Table 5.7: Parameters in the JSON file for Finnish exercises.

Hungarian Exercises Similarly to the Finnish language learning exercises, three Hungarian grammar aspects were observed and processed in order to provide HFL learners with practice ma-

terial. This material can help learners master these three topics, which were selected based on previous research of Máté (1999).

In Hungarian, transitive verbs have two paradigms: a definite and an indefinite conjugation. These verbs agree with their objects in definiteness (Coppock and Wechsler 2012). This increases the number of possible suffixes in the case of some verbs and further complicates the decision regarding the selection of the correct inflection in the given context. To practice the difference between these two conjugations, example sentences that contain a transitive verb are queried from the database. First, the list of transitive verbs is queried. To determine what transitive verbs are, it is assumed that they appear in at least one sentence with the `Definite=Def` pattern. After listing the transitive verbs with the help of this rule, sentences are selected that contain any of these verbs. There is no restriction regarding the paradigm that the verbs appear in in these instances, they can have a definite (`Definite=Def`) or indefinite (`Definite=Ind`) conjugation since the task of the learner is to decide which one of the two paradigms is the correct one. After the extraction of sentences, the verb is replaced with an input text field, and the lemma of the verb is given between parentheses. Since Hungarian is also a pro-drop language, the necessary information regarding the person and number of the subject is given after the lemma of the verb, see Figure 5.17.



A Balatonra egyre többen (látogat, T/3) külföldről is.

Figure 5.17: Example of the Hungarian definite and indefinite conjugation task.

The correct use of verbal prefixes (or preverbs) can also be challenging. Preverbs in Hungarian can appear both preverbally and postverbally, depending on the structure and word order of the sentence. It is a debated topic exactly which words belong to this category (Kalivoda 2021), but in this application 13 words are defined as preverbs: *be, ki, le, fel, meg, el, át, bele, ide, oda, szét, össze, vissza*. The list contains the 6 prototypical preverbs, as well as 7 of the central verbal prefixes defined in Kalivoda (2021). This list can be augmented or reduced as necessary in the future. The condition, that defines which Hungarian sentences are suitable for this exercise, is the following: the sentence must contain any of these 13 words either as an isolated word, or attached to the beginning of a verb. In the case of the former, the `deprel` field of the isolated word must be `PREVERB` in the `Analysis2Sentence` table, to prevent the inclusion of sentences where homographs of the preverbs would otherwise appear (such as *ki* which can also be a pronoun meaning ‘who’ and has

10 distinct `deprel` values in the database).

The manual validation of 300 sentences that were obtained with this initial condition led to the observation that the beginning of some verbs was incorrectly marked as preverbs, although they were part of the root. The exclusion of such sentences was conducted by adding a stop list to the condition that contains these verbs. The total list of excluded lemmata is shown in example (7).

(7) <i>becsül</i> ('appreciate')	<i>beleng</i> ('swing')
<i>beles</i> ('peek')	<i>beszél</i> ('speak')
<i>felejt</i> ('forget')	<i>felel</i> ('respond')
<i>kiabál</i> ('shout')	<i>lehel</i> ('breathe')
<i>lesz</i> ('will be')	<i>megy</i> ('go')

Two substrings could also be observed that often caused false positive tagging at the beginning of the verbs and as a consequence, incorrectly generated tasks. In this case, only a substring at the beginning of the verbs shall be eliminated, to be able to exclude multiple verbs without adding them one by one to the condition. In this case, the beginning of matching verbs is actually a preverb, but since this preverb is not yet included in the list of verbal prefixes, and another, shorter preverb is defined in the condition, the task generation is unsuccessful (e.g. *felülmúl* is divided incorrectly into preverb and verb as *fel-ülmúl*, instead of *felül-múl*). Verbs that start with the strings shown in example (8) were therefore excluded from this exercise to increase the quality of the generated examples. These preverbs can be added to the list of verbal prefixes in the future.

- (8) a. *felül* ('above', 'over', etc.)
excluded lemmata: *felülmúl, felülvizsgál*
number of excluded sentences: 3
- b. *ellen* ('against')
excluded lemmata: *ellenez, ellenjavall, ellenkezik, ellenszegül, ellensúlyoz, ellentmond, ellenáll, ellenőríz*
number of excluded sentences: 34

An example task regarding the preverbs can be seen in Figure 5.18.

A betörés híre bosszantotta a szomszédomat.

Figure 5.18: Example of the Hungarian preverb task.

One way how the Hungarian language expresses possession is by adding the possessive suffix to the possessed object. This suffix depends on the person of the possessor, whether the possessed object is singular or plural, and whether the possessed noun ends with a vowel or a consonant. Besides these, vowel harmony also affects the suffix. All of these factors are responsible for the high number of allomorphs the possessive suffix has. For instance, in the case of a third person singular possessor, these allomorphs are *-a*, *-e*, *-ja*, or *-je*. Furthermore, the suffix on the possessed object may cause some changes to the root of the noun, see example (9).

- (9) a. *név* ‘name’ → *nev-em* ‘my name’
b. *bokor* ‘bush’ → *bokr-od* ‘your bush’
c. *kutya* ‘dog’ → *kutyá-ja* ‘his/her dog’

The possessive suffix is marked by the morphological analyzer with the `Number[psor]` and `Person[psor]` features on the possessed noun. If a sentence contains a word that has these features among its morphological codes, it can serve as an example sentence in this task. To help learners only focus on the selection of the right allomorph, and the production of the correct word form, only sentences containing singular nominative nouns as the possessed object are queried from the database. However, a possible direction of future research is to create an exercise that combines declension with possession. These shall let learners practice not only the possessive construction but also how other suffixes (e.g. the inessive case ending) can be attached to the base form of the possessed noun (e.g. *nev-em-ben*).

After retrieving the compatible sentences from the database, the words which express the possessed object are replaced with text input fields and the lemma of these words is given. The person of the possessor is also given between parentheses because the pronominal possessors are most of the time pro-dropped in Hungarian, see Figure 5.19. This is possible with the help of the parameters (`checkNumber` and `checkPerson`) set in the JSON file, which in this case processes the `Number[psor]` and `Person[psor]` features in the `feats` field.

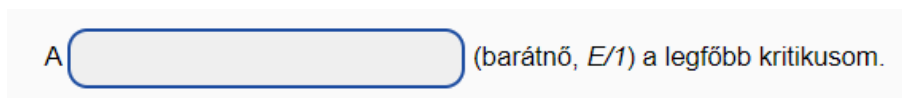


Figure 5.19: Example task for the Hungarian possessive constructions.

Regarding the Hungarian exercises, the parameters set in the JSON file can be summarized as displayed in Table 5.8.

Parameter	Definite/indefinite	Preverbs	Possessive construction
checkTense	true	false	false
checkNumber	true	false	true
checkPerson	true	false	true
showLemma	true	false	true

Table 5.8: Parameters in the JSON file for Hungarian exercises.

In Section 4.3, it was mentioned that feedback is one of the drawbacks of CALL applications. It was shown that error diagnosis is more important than the correction of errors. This is why the CALL system that has been built within this research project does not display the stored solutions when the learners submit the answers and ask for the evaluation of their solutions. It only notifies the learner about the exercises in which the initial word form that has been masked out from the sentence matches the one given by the learner, and marks the rest of the answers as possibly wrong, which the learners can then revisit.

5.4.6 Evaluation and Error Analysis

Monolingual flashcards introducing the target words with their definition were created for both Finnish and Hungarian. The number of monolingual Finnish flashcards is 111,529, while Hungarian data resulted in 70,110 flashcards. Bilingual flashcards, which present target words with their translation equivalents in the source language, were also proposed in this CALL application. 796,584 such flashcards can be generated from the collected data set. Since automatic data extraction does not always lead to perfectly accurate data, the information that is presented to the language learners must first be checked and validated. This can cause a decrease in the number of flashcards, since to date, many relations have not yet been validated (see Section 5.2.3 for more details). The number of validated items can be seen in Table 5.9.

Type of Flashcard	# of Validated Data
Hungarian monolingual	388
Finnish monolingual	188
Finnish–Hungarian bilingual	680

Table 5.9: Number of validated data that can be utilized as flashcards.

Using the data set that was automatically extracted with the help of different methods described in Section 2.5, and the queries that define the structure which belongs to a certain type of exercise, the retrieval of thousands of examples was possible from the database. The exact number of sentences that can be utilized for each task can be seen in Table 5.10.

Exercise Type	# of Sentences Matched
Finnish objects	6,831
Finnish past tenses	4,438
Finnish passive construction	1,914
Hungarian definite and indefinite conjugation	8,601
Hungarian verbal prefixes	4,275
Hungarian possessive construction	3,801

Table 5.10: Number of sentences obtained for each exercise type.

It is of high importance that language learners shall only encounter correct linguistic data and feedback in this application, since erroneous input may negatively affect the language learning process. This is why the results of manual validation are indicated for each sentence in the database, and only correct data can be found in the exercises.

The validation is conducted in a stratified fashion. First, editors need to ensure that the sentence is a well-formed sentence in the given language. Sentences are tagged as not well-formed, when they are not complete, correct sentences, or when they contain typographical errors. Then, the validation of the next level of analysis takes place, which is lemmatization. If a sentence is well-formed, and the lemmatization is correct for each token, the part of speech information is checked. After validating the output of the part of speech tagger for each word in the sentence, the lexical and grammatical properties can be checked, followed by the output of the dependency parser. If a sentence proves to be correctly analyzed, it is marked as a correct analysis, and the sentence can be used in the CALL application.

The validation of a subset of sentences was executed as described above. The sentences were

selected with a random sampling technique. The results are presented in Table 5.11.

	Finnish	Hungarian
Not well-formed sentences	12 (2.51%)	14 (3.74%)
Erroneous lemmatization	142 (29.71%)	12 (3.21%)
Erroneous part-of-speech tag	4 (0.84%)	13 (3.48%)
Erroneous morphological features	15 (3.14%)	9 (2.41%)
Erroneous dependency analysis	10 (2.09%)	25 (6.68%)
Correct sentences	295 (61.72%)	301 (80.48%)
Total number of validated sentences	478 (100%)	374 (100%)

Table 5.11: Details of manual validation regarding analysis of sentences.

In total, 478 sentences have been validated for Finnish and 374 for Hungarian, and the output of the applied language processing tools (`emtsv`, `omorfi` and `uralicNLP`) was analyzed. According to the preliminary results, the most erroneous output was given by the lemmatizer of the `omorfi` tool (29.71% of Finnish sentences contain at least one incorrectly lemmatized token). The Hungarian natural language processing tool (`emtsv`) gives better overall performance than the Finnish tools (80.48% and 61.72% precision, respectively). In case of the Hungarian NLP pipeline, the least precision was reached by the dependency parser module (`emDep`), which produced incorrect output for 6.68% of the validated sentences.

Table 5.12 gives an example sentence for each type of error shown in Table 5.11 and indicates the error, as well as the correct analysis.

Type of Error	Example	Error indication
Not well-formed sentence	Az iringófélékről semmit sem <u>tudoot</u> mondani.	*tudoot → tudott
Lemmatization	Ebben a konkrét kérdésben nem <u>értünk</u> egyet.	lemma: *‘‘ér’’ → ‘‘ért’’
Part of speech tag	Tönkrement a felejtő <u>tár</u> a számítógépe-men.	*VERB → NOUN
Morphological features	Saara on minun <u>lapseni</u> .	*plural → singular
Dependency analysis	Katsoitko BBC:n <u>dokumentin</u> leijonista?	head: *4 → 1

Table 5.12: Examples for sentences where the language processing tools made mistakes.

By observing big amounts of incorrectly analyzed sentences, it is possible to understand certain properties of the data set. Some of the incorrect analyses can be explained by certain patterns that

can be discovered in the structure of sentences. After data inspection, the following conclusions could be drawn.

Many times the reason for incorrect lemmatization of Finnish words is that the sentence is a written version of what is called *puhekieli* or ‘spoken language’, a colloquial variety of Finnish, in which the applied rules differ from that of standard Finnish. The `omorfi` tool is not capable of handling this kind of input. Other times, the lemma is incorrectly identified due to the high number of potential lemmata. In highly inflected languages, there are numerous grammatical homonyms. To illustrate this phenomenon, the three possible lemmata of the Finnish word form *teillä* is shown in example (10).

- (10) a. *te* ‘you-AD.PL’
 b. *tie* ‘street-AD.PL, way-AD.PL’
 c. *tee* ‘tea-AD.PL’

Similarly to the Finnish case, Hungarian lemmata are also frequently misidentified. The disambiguation tool yields a stem that is indeed a possible lemmatization of the word form. However, in the given context (which is restricted to the sentence in question), the result of the disambiguation is incorrect. Hence, the sentence analysis is marked as incorrect and the overall precision of the morphological analysis is lower due to lemmatization.

Considering the high proportion of inaccurately analyzed sentences (38.28 % for Finnish and 19.52% for Hungarian), a more precise estimate of how many correct sentences will match the rules defined by the SQL queries can be given. The expected number of correctly analyzed sentences is provided for each exercise type in Table 5.13.

Exercise Type	# of Expected Sentences
Finnish objects	4,216
Finnish past tenses	2,739
Finnish passive construction	1,181
Hungarian definite and indefinite conjugation	6,922
Hungarian verbal prefixes	3,441
Hungarian possessive construction	3,059

Table 5.13: Number of expected sentences where the results of language processing are supposedly correct.

After the validation of the output of language processing tools, a small sample of the automatically generated tasks was examined. The purpose of the second validation was threefold: to measure the quality of the proposed queries, to ensure that the automatic selection of sentences works as expected, and to verify that there is only one grammatically correct answer to each task. This is an especially important factor since losing credibility because of a falsely flagged, but otherwise correct solution is unwanted in any language learning application. As a consequence, each sentence must be carefully inspected before they appear in the application. This sample was made up of sentences whose analysis had already been validated and accepted as correct since the main aim of this evaluation is to measure the precision of the queries that have been defined, rather than the output of the text processing tools themselves. In case the erroneous analyses were not excluded from this validation process, the evaluation of the task generation and SQL queries would be distorted. On the other hand, this validation is necessary: it needs to be ensured that the transformation of a full sentence into a cloze exercise has been successfully done. In the case of each exercise type, a subset of sentences appeared and the validators were asked to mark them as either correct or incorrect, according to the guidelines that consisted of the following points:

1. Mark a sentence incorrect if it is not related to the given part of the grammar (e.g. in the case of the Hungarian preverb task, the sentence does not contain any verbal prefixes).
2. Mark a sentence incorrect if another word of the sentence shall be masked out instead of what has been, or if the masking shall include more words.
3. Mark a sentence incorrect if more than one potentially correct solution can be provided or if the sentence does not provide sufficient context to be able to successfully guess the missing word and its correct form.
4. Mark a sentence correct otherwise.

The results of this validation process can be seen in Table 5.14.

As the precision of each exercise type suggests, most of the rules that were defined and applied to create the fill-in-the-blank exercises proved to produce high-quality results. In fact, the sample size is rather small, but this evaluation was limited due to time restrictions and the unavailability of more human correctors.

Exercise Type	# of Validated Sentences	Precision
Finnish objects	101	93.07%
Finnish past tenses	100	91.00%
Finnish passive construction	100	98.00%
Hungarian definite and indefinite conjugation	105	91.43%
Hungarian verbal prefixes	100	85.00%
Hungarian possessive construction	100	98.00%

Table 5.14: Details of the manual validation of automatically generated tasks.

Examples are provided in example (11) for both Finnish and Hungarian to illustrate what kinds of sentences were marked incorrect. In these examples, the (sub)word that was masked out and replaced by a text field in the sentence is underlined. Some sentences were labeled problematic due to the lack of sufficient context. In these cases, the high number of potentially correct, acceptable solutions is the reason why sentences are marked as unsuitable for language learning purposes.

- (11) a. Poiketaankos tuohon kuppilaan sumpille?
 b. Jotkin linnut ovat menettäneet lentokykynsä.
 c. Koko yhteiskunnan kerma oli kutsuttu linnanjuhliin.
 d. Utólagos beiratkozásra csak a doktori iskola elnöke vagy alenöke adhat engedélyt.
 e. A beszélgetésüket kihallgatták.

In the case of the Finnish exercises, it was observed that some sentences cannot be included in the exercises due to the presence of some discourse clitics on the missing word, which would be unguessable by the learners. Example (11a) illustrates this in the case of a passive construction, where the ending *-kos* is not part of the passive marker, rather, it is the combination of the interrogative clitic (*-ko*) and the *-s* particle that – besides having many other functions – helps to create familiarity and soften commands.

Other times, it was not a clitic, but a possessive suffix (see example (11b)). These sentences had to be rejected because in Finnish, three word forms are identical when a possessive suffix appears at the end of the words: nominative singular and plural, as well as genitive singular forms. To illustrate this phenomenon, examine the three following sentences, where the word form *autoni* ‘my car’ expresses different syntactic functions. In example (12a), it substitutes a singular nominative

form, while in example (12b), it is a plural nominative, which is also signaled by the third person plural form of the verb. Example (12c) shows the genitive singular usage. The three word forms are identical because of the presence of the possessive suffix (-ni) at the end of the word. If there is a possessive suffix at the end of a masked-out word, first of all, the learners would not be informed that they need to attach the correct possessive suffix, leading to an incorrect solution. Secondly, even if the possessive suffix was indicated, the word forms would mostly be identical, which is disadvantageous, assuming that the task is to select the correct case of the word.

- (12) a. Autoni on sininen. ('My car is blue.')
- b. Autoni ovat rikki. ('My cars are broken.')
- c. Autoni rengas on puhjennut. ('The tire of my car is flat.')

As mentioned earlier, many times, the reason why a sentence was marked incorrect was due to the lack of sufficient context. In example (11c), the simple past tense (imperfekti) of the verb is not the only possible solution, since the sentence does not provide sufficient background information that would specify which past tense shall be used.

The cloze exercises were designed in such a way that only one word is masked out, hence, wherever another word in the sentence would give a hint about the solution and help the learner choose the correct word form, the exercise was rejected in order to keep only useful, educative tasks. An example of this type of error is shown in example (11d). The masked-out word (*elnöke*) is a substring of another word displayed in the sentence (*alelnöke*).

In example (11e), more than one potentially correct preverbs can be given (e.g. *lehallgat*, *kihallgat*). Whenever the learner types in the preverb that is not the initial, missing verbal prefix, the system would flag the answer as incorrect, although it might be an alternatively correct answer. Since this behavior is undesired, sentences with multiple potentially correct answers are eliminated.

5.4.7 Conclusion and Future Work

In the previous sections, two CALL modules were presented which had been implemented for two FU languages: Finnish and Hungarian.

Both monolingual and bilingual virtual flashcards were created automatically to help learners acquire new vocabulary items. These cards are based on the data set extracted with the methods presented in Section 2.5. The learners can choose the target language they want to learn (Finnish or Hungarian) and whether the explanation side of the card should display the translation equivalent of the target word or the definition in the target language. The module consists of two phases: in the first, learners are provided new vocabulary items with their explanations, and in the second, these new elements are tested in an active–productive setting.

The cloze (fill-in-the-blank) task module includes several grammar exercises for both Finnish and Hungarian, focusing on those parts which pose the biggest challenges for learners. Three types of exercises are developed for Finnish, and three for Hungarian. The case of the object in a Finnish sentence depends on many factors, and it is one of the difficulties learners need to overcome. There exist three kinds of past tenses in this language, while speakers of Hungarian mostly use and encounter only one in their native tongue. Understanding which one to use in a certain scenario requires practice. The third type of exercise for Finnish concerns passive construction, which fulfills many functions in this language. The most difficult parts of Hungarian grammar – as observed by Máté (1999) – seemed to include the existence of two verbal paradigms in the case of transitive verbs (definite and indefinite conjugation), verbal prefixes, and possessive construction. Exercises have been generated and examples have been automatically selected and composed in an application that allows learners to practice these challenging parts of the two FU languages.

This section, therefore, addressed three of the initial research questions, repeated here:

- RQ 4. Is it possible to create language learning exercises with automatic methods and predefined rules in order to help learners practice the most difficult aspects of these languages?
- RQ 5. Can a large number of exercises be generated with the help of different rules that would substitute the manual creation of such exercises done by foreign language instructors?
- RQ 6. How accurate are the examples of such an application? Are the applied rules general enough to cover multiple examples, but at the same time specific enough to only include suitable examples?

RQ 4. was addressed with the creation and generation of virtual flashcards and cloze exercises using automatic methods.

The number of sentences that can be used in order to automatically generate cloze tasks was presented in Table 5.10. The number of exercises would be decreased by 20 to 40% due to the incorrectly analyzed data points, as shown in Table 5.13. However, composing exercises in the automatically achieved quantity would require weeks and maybe even months of manual work. The same applies to virtual flashcards. As presented in Section 5.4.6, more than 70,000 and 110,000 monolingual flashcards, as well as nearly 800,000 bilingual flashcards, can be generated for Finnish and Hungarian with the help of the automatically extracted data set. RQ 5. can therefore be considered confirmed, as the automatic data extraction and the application of SQL queries and rules led to thousands of exercises. Furthermore, the results found in Table 5.14 support the hypothesis that in general, a sentence provides enough context to guess what element or elements are masked out in it.

The collected example sentences, definitions, and translation pairs only appear in the public interface of the language learning application once their precision is ensured by manual validators. The initial validation of such exercises made it possible to estimate the precision of the examples of the CALL application, in order to respond to the last research question (RQ 6.). Correctly analyzed examples underwent a second revision, to examine the quality of the queries and rules that generate fill-in-the-blank exercises. The results of this revision can be seen in Table 5.14. It demonstrates that the automatic extraction of examples for each exercise type is of high quality.

Data validation, however, is continuous and ongoing in this project with the help of speakers of both Finnish and Hungarian. The application is made available for the public at <https://fulr.btk.ppk.e.hu/call.php>.

One of the shortcomings of the language learning application is the limited feedback it can produce. In the flashcard module, only the originally displayed target words are considered to be correct, although one definition or word in the native language of the learner may have more than one corresponding words in the target language. Similarly, regarding the cloze exercises, other potentially correct answers may also be considered grammatically correct, besides the words that are masked out from the sentences by automatic methods. However, instead of providing a list of all possible answers and including more sentences in the application, only sentences with a sole solution are marked as accepted during validation. Therefore, the system does not flag potentially correct answers as incorrect, ensuring the higher precision of the feedback.

As recommended above, the list of Hungarian verbal prefixes is limited to 13 elements in this work. However, more sentences could be comprised in this task if other preverbs were added to the list. Further studies need to be carried out in order to validate the results of this extended list and to ensure the high precision of the generated tasks.

An additional contribution of this application is expected to be the collected learners' data: the system collects and stores data from language learners that can enable researchers to further investigate the difficulties that learners face when learning Finno-Ugric languages, or even empirically prove (or disprove) the hypotheses relating to these languages with complex morphology.

5.5 Conclusion

In this chapter, the proposed framework (called Finno-Ugric Lexical Resources) was described and its main components (a DWS, an online bilingual dictionary, and a CALL application) were presented in detail.

The aim of the proposed DWS is to facilitate the work of lexicographers and provide an interface that does not require advanced IT skills to communicate with and modify the data in the database. This component was described in Section 5.2.

The dictionary in this framework is a public interface that has been developed for language learners who can access the manually validated data available in the database. It is a highly customizable interface, which can fulfill the needs of a wide target audience. Most of the benefits of online dictionaries have been implemented in this bilingual dictionary, including a sophisticated search mechanism and easy navigation between entries. The dictionary interface, along with its most salient features, was presented in Section 5.3.

The language learning application is composed of two main modules: a virtual flashcard and a cloze exercise module. Monolingual and bilingual flashcards are proposed to help language learners acquire new vocabulary items in Finnish or Hungarian. Example sentences extracted with the help of the `Wiktionary Parser` and `WordNet Connector` algorithms can be utilized in order to generate fill-in-the-blank (also called cloze) exercises, where one component of the sentence is masked out and the learner needs to fill in the missing word form by conjugating a verb or declining a noun or adjective. The morphological analyzers and NLP tools applied to the Finnish and Hungarian example sentences have some difficulty providing high-quality output (shown in Table 5.11), which

causes a loss of data in the case of the generated cloze examples. The SQL queries and word removal methods led to several thousand data points for each language learning exercise type even after this decrease in data (see Table 5.13). These examples must be validated before including them in the publicly available language learning application, as it is imperative that learners only encounter grammatical, correct sentences. More information about the CALL application can be found in Section 5.4.

6 Conclusions

6.1 Summarized Results

In the present dissertation, a framework has been developed for a relatively rare, less frequent language pair (consisting of two Finno-Ugric languages: Finnish and Hungarian), while providing insight into the structure and characteristics of these languages. There are numerous resources and materials available online for Finnish and Hungarian, however, the primary goal of this research was to improve the interoperability between these sources, and create a dictionary and language learning application for the learners of these two languages. With the help of natural language processing tools and methods, an online bilingual dictionary was compiled using a database schema and dictionary writing system that can facilitate lexicographic work. This system utilizes a language-independent annotation framework (Universal Dependencies) in order to be extensible with further FU languages in the future.

Chapter 1 offered a short introduction to lexicography and vocabulary acquisition methods. Some of the main problems of already existing online dictionaries for Finnish and Hungarian were highlighted, and the need for a system that incorporates the most important features of these morphologically rich languages was expressed. Several examples of language learning applications and websites have been mentioned, emphasizing that the two FU languages are quite underrepresented in the most widely used programs.

In Chapter 2, the resources which had been used in the data extraction phase were presented, and the applied bilingual lexicon induction methods, as well as the resulting data sets, were described in detail. Several automatic dictionary building methods have been proposed during this research work to obtain hundreds of thousands of bilingual translation candidates for the Finnish–Hungarian language pair automatically. With the help of automatic tools, the cost and required time of manual dictionary building could be reduced. In addition to these newly proposed methods, an already existing tool (developed by Ács et al. (2013)) was applied to obtain additional translation pairs. The main results of this chapter can be summarized as follows:

1. I created `Wiktionary Parser`, a script that parses the Finnish and Hungarian Wiktionary editions and collects bilingual translation candidates with their part-of-speech information,

as well as monolingual lemma–definition and lemma–example sentence pairs. The script is freely available and the resulting data set is uploaded to the FULR database, where every data point will be manually validated.

2. I developed `WordNet Connector`, a script that links the Finnish and Hungarian WordNets. It collects bilingual translation candidates by connecting the synsets of these two databases. The algorithm can also extract the elements of synsets as tab-separated values, as well as Hungarian definitions and example sentences from the Hungarian database. The Finnish WordNet does not contain these kinds of data. The script is freely available and the resulting data set is uploaded to the FULR database, where every data point will be manually validated.
3. I created a script called `OPUS Extractor` that extracts bilingual word pairs from the Finnish and Hungarian word alignments. It is possible to sort the word pairs by their co-occurrence number or in alphabetical order. The script is freely available.
4. I lemmatized the data set extracted from the OPUS corpus, since the word alignments were generated from running texts, resulting in only 6.25% precision. As it was also observed by Simon and Mittelholcz (2017), it was found that the extracted translation candidates contained many suffixed word forms, which resulted in a low quality proto-dictionary. Lemmatization led to a 73.13% decrease in the number of word pairs, and resulted in 93.137% precision. This result clearly shows that the precision of bilingual translation pair extraction from running texts in morphologically rich languages can greatly benefit from lemmatization, when the word pairs are intended to be included in a dictionary as headwords.
5. By validating hundreds of translation candidates, synonym pairs, definitions and example sentences, it was possible to compare the precision of the applied methods. I showed that the highest precision could be achieved by the `Wiktionary Parser` method proposed in this dissertation, followed by one of the modes (`extract`) of the `wikt2dict` tool developed by Ács et al. (2013). This proves that the crowd-sourced resource, Wiktionary, contains mostly high-quality, reliable data.

Chapter 3 presented the details of a language-independent lexicographical database that has been created to store the data extracted by the above-mentioned methods. The results of this chapter are

the following:

6. I designed and developed a lexicographical, MariaDB-based relational database that stores data in a language-independent way. This repository of data stores all kinds of information (translation candidates, example sentences, definitions, and synonyms) in a universal way. The main units of the proposed structure are entities, as opposed to traditional lexicography approaches which consider entries as the basic units of the dictionary editing process. This database schema makes it possible to create an automatically reversible bilingual dictionary from the obtained data set.

Chapter 4 gave insights into the field of Computer-Assisted Language Learning by first providing an overview of what kinds of activities and tasks can be included in this interdisciplinary field, and then presenting two distinct perspectives to grasp the different approaches of CALL. This chapter also focused on the degree of integration of computer technology and CALL applications into the curriculum. Despite all the benefits that can be attributed to CALL, the language learning applications often face some restrictions which were presented in Section 4.3. Furthermore, the chapter emphasized the need for a Finnish–Hungarian language learning application supported by language processing tools and methods that focuses on the parts of these Finno-Ugric languages which pose the biggest challenges for learners.

The different components of the proposed framework were detailed in Chapter 5. The framework consists of an online bilingual dictionary, a dictionary writing system, and a language learning application. These are all designed to provide a common and universal interface, where learners of Finnish and learners of Hungarian can find valuable data and practice specific aspects of these languages. The main contributions of this chapter are the following:

7. I created the Finno-Ugric Lexical Resources platform, which consists of three components: an online bilingual dictionary, a dictionary writing system and a language learning application.
8. I designed the online bilingual dictionary interface for learners of Finnish and learners of Hungarian. This dictionary builds on the data set stored in the language-independent database described earlier. The generation of the dictionary entries happens automatically from the entities and relations found in the database defined by certain rules.

9. I developed the dictionary writing system that allows editors to easily manipulate the data, without advanced IT knowledge.
10. I created the language learning application that incorporates two modules: a flashcard module and a module with cloze exercises. The flashcard module facilitates vocabulary acquisition and offers two types of cards: monolingual cards with the help of lemma–definition pairs, and bilingual flashcards with translation pairs. The cloze exercises can help learners practice different grammar aspects. I determined some rules in order to automatically generate task types and examples for these types regarding 3-3 challenging aspects of the Finnish and Hungarian grammar.
11. A subset of the examples for each cloze exercise has been validated and the predetermined rules proved to generate the tasks very precisely (with above 90% precision in all task types except one). This method can be applied to alleviate the tedious manual creation of such exercises by language instructors.

6.2 Directions for Further Research

This research work provided a detailed analysis of two Finno-Ugric languages with extensive case systems from a lexicographic perspective. A language-independent database has been constructed and populated with automatically extracted data sets from different resources, and the quality of the applied methods has been evaluated. Furthermore, a language learning application was proposed, in which a large number of flashcards and cloze exercises was generated from the same data sets with the help of queries and pre-set filters. The effects of lemmatization were also closely examined with regard to morphologically rich languages. It was found that natural language processing tools can improve the quality of bilingual translation pairs extracted from unlemmatized corpora. However, there are still many directions for further research. The possible improvements were proposed for future work in the respective chapters regarding the specific components of this framework, while some new directions regarding dictionary building and computer-assisted language learning will be pointed out in this section.

Apart from the algorithms proposed in this work, further methods can be developed and applied to these languages. Further research can be done for example in the direction of the pivot language

based method that is recommended, implemented, and perfected by many researchers in the field of NLP (Murakami 2019; Steingrímsson et al. 2021). It would be of great value to use this technique in relation to resources other than Wiktionary for Finnish and Hungarian and to improve the quality of the `wikt2dict` triangulating algorithm. It is vital to handle polysemous words in the pivot language properly and link their translations in different languages correctly by employing different constraints.

New approaches to representing words as vectors (Mikolov et al. 2013; Pennington et al. 2014; Bojanowski et al. 2017; Devlin et al. 2018) have become popular in recent years, which can open up novel avenues of future research. These methods have proved to achieve significant performance, however, they require a high computational cost. Due to the fairly limited computing resources available in this research work, low computational cost was preferred. Nevertheless, vector representations can be considered as potential directions for future work.

As the database has been designed in a language-independent way, inserting lexical information in other FU languages into the FULR database is a possible way to expand the scope of the project in the future.

The prospective learners of the CALL application presented in this thesis will provide valuable information to those who study FU languages (at the present state of the project, only Finnish and Hungarian) and who are interested in the most common mistakes that are made by the learners of these highly inflectional languages. With the help of the FULR database, it will be possible to easily carry out quantitative studies based on learner data.

References

- Abel, A. (2012). Dictionary Writing Systems and Beyond. In Granger, S. and Paquot, M. (Eds.) *Electronic Lexicography*, (pp. 83–106). Oxford: Oxford University Press.
- Ács, J., Pajkossy, K., and Kornai, A. (2013). Building Basic Vocabulary Across 40 Languages. In *Proceedings of the Sixth Workshop on Building and Using Comparable Corpora*, (pp. 52–58). Sofia, Bulgaria: Association for Computational Linguistics.
- Ahn, D., Jijkoun, V., Mishne, G., Müller, K., de Rijke, M., Schlobach, S., Voorhees, M., and Buckland, L. (2004). Using Wikipedia at the TREC QA Track. In *Text REtrieval Conference*. Gaithersburg, MD: National Institute of Standards and Technology.
- Al A'amiri, B. F. K. and Jameel, A. F. (2019). Morphological Typology: A Comparative Study of Some Selected Languages. *Journal of College of Education/Wasit*, 1(37):709–724.
- Alnajjar, K., Hämäläinen, M., and Rueter, J. (2020). On Editing Dictionaries for Uralic Languages in an Online Environment. In *Proceedings of the Sixth International Workshop on Computational Linguistics of Uralic Languages*. Wien, Austria: The Association for Computational Linguistics.
- Ammar, W., Mulcaire, G., Tsvetkov, Y., Lample, G., Dyer, C., and Smith, N. A. (2016). Massively Multilingual Word Embeddings. *arXiv:1602.01925*.
- Anderson, R. C. and Freebody, P. (1981). Vocabulary Knowledge. In Guthrie, J. T. (Ed.) *Comprehension and teaching: Research reviews*, (pp. 77–117). Newark, Delaware: International Reading Association.
- Antonsen, L., Huhmarniemi, S., and Trosterud, T. (2009). Interactive Pedagogical Programs Based on Constraint Grammar. In *Proceedings of the 17th Nordic Conference of Computational Linguistics (NODALIDA 2009)*, (pp. 10–17).
- Atkins, B. S. and Rundell, M. (2008). *The Oxford Guide to Practical Lexicography*. Oxford: Oxford University Press.

- Baroni, M., Bernardini, S., Ferraresi, A., and Zanchetta, E. (2009). The WaCky Wide Web: a Collection of Very large Linguistically Processed Web-crawled Corpora. *Language Resources and Evaluation*, 43(3):209–226.
- Bauer, L. and Nation, I. S. P. (1993). Word families. *International Journal of Lexicography*, 6(4):253–279.
- Bax, S. (2003). CALL — Past, Present and Future. *System*, 31(1):13–28.
- Beatty, K. (2003). *Teaching & Researching: Computer-Assisted Language Learning*. London: Pearson Education Limited.
- Beck, I. L., Perfetti, C. A., and McKeown, M. G. (1982). Effects of Long-term Vocabulary Instruction on Lexical Access and Reading Comprehension. *Journal of educational psychology*, 74(4):506.
- Bengio, S. and Heigold, G. (2014). Word Embeddings for Speech Recognition. In *Proceedings of the 15th Conference of the International Speech Communication Association, Interspeech*. Singapore: International Speech Communication Association (ISCA).
- Benko, V. (2014). Aranea: Yet Another Family of (comparable) Web Corpora. In *International Conference on Text, Speech, and Dialogue*, (pp. 247–256). Brno, Czech Republic.
- Bergenholtz, H. and Nielsen, J. S. (2013). What is a Lexicographical Database? *Lexikos*, 23:77–87.
- Bergmann, A., Hall, K. C., and Ross, S. M. (2007). *Language Files: Materials for an Introduction to Language and Linguistics*. Columbus: The Ohio State University Press.
- Bobály, G., Horváth, C., and Vincze, V. (2020). apPILcation: an Android-based Tool for Learning Mansi. *Proceedings of the Sixth International Workshop on Computational Linguistics of Uralic Languages*.
- Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. (2017). Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.
- Branch, H. (s.a.). Is Finnish a Difficult Language? <https://finland.fi/life-society/is-finnish-a-difficult-language/>. [Online; accessed November 3, 2022].

- Burkhanov, I. (1998). *Lexicography: A Dictionary of Basic Terminology*. Rzeszow: Wyższa Szkoła Pedagogiczna.
- Carter, R. and McCarthy, M. (Eds.) (2013). *Vocabulary and Language Teaching*. New York: Routledge.
- Chang, B., Danielsson, P., and Teubert, W. (2002). Extraction of Translation Unit from Chinese-English Parallel Corpora. In *COLING-02: The First SIGHAN Workshop on Chinese Language Processing*. USA: Association for Computational Linguistics.
- Chapelle, C. (1998). Multimedia CALL: Lessons to Be Learned from Research on Instructed SLA. *Language Learning & Technology*, 2(1):21—39.
- Christodouloupoulos, C. and Steedman, M. (2015). A Massively Parallel Corpus: the Bible in 100 Languages. *Language Resources and Evaluation*, 49(2):375–395.
- Clément, R. (1980). Ethnicity, Contact and Communicative Competence in a Second Language. In GILES, H., ROBINSON, W. P., and SMITH, P. M. (Eds.) *Language*, (pp. 147–154). Amsterdam: Pergamon.
- Coppock, E. and Wechsler, S. (2012). The Objective Conjugation in Hungarian: Agreement Without Phi-features. *Natural Language & Linguistic Theory*, 30(3).
- Cotterell, R., Mielke, S. J., Eisner, J., and Roark, B. (2018). Are all languages equally hard to language-model? In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, (pp. 536–541). New Orleans, Louisiana: Association for Computational Linguistics.
- Crookes, G. and Schmidt, R. W. (1991). Motivation: Reopening the Research Agenda. *Language Learning*, 41(4):469–512.
- de Marneffe, M.-C., Manning, C. D., Nivre, J., and Zeman, D. (2021). Universal Dependencies. *Computational Linguistics*, 47(2):255–308.
- de Schryver, G.-M. (2003). Lexicographers' Dreams in the Electronic-Dictionary Age. *International Journal of Lexicography*, 16(2):143–199.

- Delcloque, P. (2000). The History of Computer Assisted Language Learning Web Exhibition. http://www.ict4lt.org/en/History_of_CALL.pdf. [Online; accessed November 12, 2022].
- Des Tombe, L. (1992). Is Translation Symmetric? *Meta: Journal des Traducteurs*, 37(4):791–801.
- Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*.
- Dörnyei, Z. (1998). Motivation in Second and Foreign Language Learning. *Language Teaching*, 31(3):117–135.
- Dörnyei, Z. (2005). *The Psychology of the Language Learner: Individual Differences in Second Language Acquisition*. New York: Routledge.
- Dörnyei, Z. (2009). The L2 Motivational Self System. In Dörnyei, Z. and Ushioda, E. (Eds.) *Motivation, Language Identity and the L2 Self*, (pp. 9–42). Bristol, Blue Ridge Summit: Multilingual Matters.
- Durst, P. and Janurik, B. (2011). The Acquisition of the Hungarian Definite Conjugation by Learners of Different First Languages. *Lähivõrdlusi. Lähivertailuja*, (21):19–44.
- Egbert, J. L. (2005). Conducting Research on CALL. In Egbert, J. L. and Petrie, G. M. (Eds.) *CALL Research Perspectives*, (pp. 3–8). Mahwah, New Jersey: Lawrence Erlbaum Associates, Publishers.
- Elekfi, L. (1994). *Magyar ragozási szótár [Dictionary of Hungarian Inflections]*. Budapest, Hungary: Magyar Tudományos Akadémia.
- Elgort, I. (2011). Deliberate Learning and Vocabulary Acquisition in a Second Language. *Language Learning*, 61(2):367–413.
- Elgort, I. (2013). Effects of L1 Definitions and Cognate Status of Test Items on the Vocabulary Size Test. *Language Testing*, 30(2):253–272.

- Faltn, A. V. (2003). Natural Language Processing Tools for Computer Assisted Language Learning. *Linguistik online*, 17(5):137–153.
- Ferenczi, Zs. (2021). Finn–magyar fordítási párok kinyerése automatikus módszerekkel [Automatic Extraction of Finnish and Hungarian Translation Pairs]. In Grácsi, T. E. and Ludányi, Z. (Eds.) *Doktoranduszok tanulmányai az alkalmazott nyelvészet köréből*, (pp. 131–150). Budapest: Nyelvtudományi Kutatóközpont.
- Ferenczi, Zs. (2022). Nyelvtanulást elősegítő feladatok automatikus előállítására finn és magyar nyelvekre [Automatic Generation of Finnish and Hungarian Language Learning Exercises]. In *XVIII. Magyar Számítógépes Nyelvészeti Konferencia*, (pp. 213–226).
- Ferenczi, Zs., Mittelholcz, I., Simon, E., and Váradi, T. (2018). Evaluation of Dictionary Creating Methods for Finno-Ugric Minority Languages. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.
- Fishman, J. A. (1991). *Reversing Language Shift: Theoretical and Empirical Foundations of Assistance to Threatened Languages*. Clevedon, UK: Multilingual Matters.
- Forsberg, U.-M., Kovács, M., Vecsernyes, I., Markus, K., Manner, S., and Kovács, O. (2015). *Suomi–unkari-sanakirja; Finn–magyar szótár [Finnish–Hungarian Dictionary]*. Helsinki, Finland: Suomalaisen Kirjallisuuden Seura.
- Francis, W. and Kucera, H. (1982). *Word Frequency Counts of Modern English*. Providence, RI: Brown University Press.
- Fuertes-Olivera, P. A. and Tarp, S. (2014). *Theory and Practice of Specialised Online Dictionaries*. Berlin/Boston: De Gruyter.
- Fung, P. (1995). Compiling Bilingual Lexicon Entries from a Non-Parallel English-Chinese Corpus. In *Third Workshop on Very Large Corpora*. Cambridge, MA.
- Fung, P. and Yee, L. Y. (1998). An IR Approach for Translating New Words from Nonparallel, Comparable Texts. In *The 17th International Conference on Computational Linguistics*. Montréal, Canada.

- Gaál, P. (2012). Szempontrendszer online szótárak minősítéséhez [Evaluation Criteria for Online Dictionaries]. *Magyar Terminológia*, 5(2):225–250.
- Gaál, P. (2016). Onlineszótár-használat Magyarországon (OHM): Egy kérdőíves szótárhasználati felmérés eredményei I [Online Dictionary Use in Hungary: Results of a Questionnaire Survey of Dictionary Use I.]. *Alkalmazott Nyelvtudomány*, 16(2).
- Gardner, R. C. (1960). *Motivational Variables in Second Language Acquisition*. PhD thesis, McGill University.
- Gardner, R. C. and Lambert, W. E. (1959). Motivational Variables in Second-Language Acquisition. *Canadian Journal of Psychology*, 13(4):266–272.
- Ghaffar Ahmed, N. and Mwai, A. (2014). The Role of Finnish Language in the Integration Process Among Immigrant Women.
- Granger, S. and Paquot, M. (Eds.) (2012). *Electronic Lexicography*. Oxford: Oxford University Press.
- Hart, R. S. (1995). The Illinois PLATO Foreign Languages Project. *CALICO Journal*, 12(4):15–37.
- Haverinen, K., Nyblom, J., Viljanen, T., Laippala, V., Kohonen, S., Missilä, A., Ojala, S., Salakoski, T., and Ginter, F. (2014). Building the essential resources for finnish: the turku dependency treebank. *Language Resources and Evaluation*, 48:493–531.
- Hazenberg, S. and Hulstun, J. H. (1996). Defining a Minimal Receptive Second-language Vocabulary for Non-native University Students: An Empirical Investigation. *Applied Linguistics*, 17(2):145–163.
- Héja, E. (2010). The Role of Parallel Corpora in Bilingual Lexicography. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*. Valletta, Malta: European Language Resources Association (ELRA).
- Héja, E. (2015). *Quantifying Translational Equivalence: The Usability of Language Technology Methods and Parallel Corpora in Bilingual Lexicography*. PhD thesis, Eötvös Loránd University.

- Hämäläinen, M. (2019). UralicNLP: An NLP Library for Uralic Languages. *Journal of Open Source Software*, 4(37):1345.
- Indig, B., Sass, B., Simon, E., Mittelholcz, I., Vadász, N., and Makrai, M. (2019). One Format to Rule Them All – The emtsv Pipeline for Hungarian. In *Proceedings of the 13th Linguistic Annotation Workshop*, (pp. 155–165). Florence, Italy: Association for Computational Linguistics.
- Irvine, A. and Callison-Burch, C. (2017). A Comprehensive Analysis of Bilingual Lexicon Induction. *Computational Linguistics*, 43(2):273–310.
- Jackson, H. (2002). *Lexicography: An Introduction*. London: Routledge.
- Jo, G. (2018). English Vocabulary Learning with Wordlists vs. Flashcards; L1 Definitions vs. L2 Definitions; Abstract Words vs. Concrete Words. *Culminating Projects in English*, 132.
- Joffe, D. and de Schryver, G.-M. (2004). TshwaneLex, a State-of-the-art Dictionary Compilation Program. In Williams, G. and Vessier, S. (Eds.) *Proceedings of the 11th EURALEX International Congress*, (pp. 99–104). Lorient, France: Université de Bretagne-Sud, Faculté des Lettres et des Sciences Humaines.
- Jones, F. R. (1995). Learning an Alien Lexicon: a Teach-yourself Case Study. *Second Language Research*, 11(2):95–111.
- Kalivoda, Á. (2021). *Igekötős szerkezetek a magyarban [Preverb Constructions in Hungarian]*. PhD thesis, Pázmány Péter Katolikus Egyetem.
- Karlsson, F. and Chesterman, A. (2008). *Finnish: An Essential Grammar*. London: Routledge.
- Katinskaia, A. and Ivanova, S. (2019). Multiple Admissibility: Judging Grammaticality using Unlabeled Data in Language Learning. In *Proceedings of the 7th Workshop on Balto-Slavic Natural Language Processing*, (pp. 12–22). Florence, Italy: Association for Computational Linguistics.
- Katinskaia, A., Nouri, J., and Yangarber, R. (2018). Revita: a Language-Learning Platform at the Intersection of ITS and CALL. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

- Katinskaia, A. and Yangarber, R. (2021). Assessing Grammatical Correctness in Language Learning. In *Proceedings of the 16th Workshop on Innovative Use of NLP for Building Educational Applications*.
- Kilgarriff, A. (2006). Word from the Chair. In De Schryver, G.-M. (Ed.) *DWS 2006: Proceedings of the Fourth International Workshop on Dictionary Writing Systems*, (p. 7).
- Korhonen, S. (2012). *Oppijoiden suomi. Koulutettujen aikuisten käsitykset ja kompetenssit [Perceptions and Competences of Adult Learners of Finnish]*. Helsinki: Helsingin yliopisto.
- Kornai, A. (2013). Digital Language Death. *PLOS ONE*, 8(10):1–11.
- Körtvélyessy, L. (2017). *Essentials of Language Typology*. Košice: Univerzita Pavla Jozefa Šafárika in Košice. [Online; accessed December 12, 2022].
- Krizhanovsky, A. A. and Smirnov, A. V. (2013). An Approach to Automated Construction of a General-purpose Lexical Ontology Based on Wiktionary. *Journal of Computer and Systems Sciences International*, 52(2):215–225.
- Laakso, J. (2015). The Finno-Ugric Foundations of Language Teaching. *Lähivõrdlusi. Lähiverailuja*, 25:172–190.
- Lample, G., Conneau, A., Denoyer, L., and Ranzato, M. (2018). Unsupervised Machine Translation Using Monolingual Corpora Only. In *6th International Conference on Learning Representations, ICLR 2018*. Vancouver, BC, Canada: OpenReview.net.
- Langemets, M., Loopmann, A., and Viks, U. (2010). Dictionary Management System for Bilingual Dictionaries. In Granger, S. and Paquot, M. (Eds.) *eLexicography in the 21st Century: New Challenges, New Applications*, (pp. 425–429). Louvain-la-Neuve: Presses universitaires de Louvain.
- Laufer, B., Elder, C., Hill, K., and Congdon, P. (2004). Size and Strength: Do We Need Both to Measure Vocabulary Knowledge? *Language Testing*, 21(2):202–226.

- Laufer, B. and Ravenhorst-Kalovski, G. C. (2010). Lexical Threshold Revisited: Lexical Text Coverage, Learners' Vocabulary Size and Reading Comprehension. *Reading in a Foreign Language*, 22:15–30.
- Levy, M. (1997). *Computer-Assisted Language Learning: Context and Conceptualization*. Oxford: Oxford University Press.
- Lewis, M. and Simons, G. (2010). Assessing Endangerment: Expanding Fishman's GIDS. *Revue Roumaine de Linguistique*, 55(2):103–120.
- Lindén, K. and Carlson, L. (2010). FinnWordNet–Finnish WordNet by Translation. *LexicoNordica–Nordic Journal of Lexicography*, 17:119–140.
- Maticsák, S. (2017). A Magyar nyelv és kultúra oktatása finnországi egyetemeken [Teaching Hungarian and Hungarian Culture at the Universities of Finland]. *Gerundium*, 8(3):58–76.
- Maticsák, S. and Laihonen, P. (2011a). Fejezetek a finn–magyar lexikográfia történetéből.[Chapters on the History of Finnish-Hungarian Lexicography]. *Magyar nyelvjárások*, 49(49):129–156.
- Maticsák, S. and Laihonen, P. (2011b). Milestones in the History of Hungarian/Finnish Bilingual Lexicography. In Fábrián, Z. (Ed.) *Hungarian lexicography I. Bilingual dictionaries*. Budapest: Akadémiai Kiadó.
- McEnery, T. and Xiao, R. (1999). Domains, Text Types, Aspect Marking and English-Chinese Translation. *Languages in Contrast*, 2(2):211–229.
- Mechura, M. (2016). Data Structures in Lexicography: from Trees to Graphs. In *RASLAN*, (pp. 97–104).
- Měchura, M. B. et al. (2017). Introducing Lexonomy: an Open-source Dictionary Writing and Publishing System. In *Electronic Lexicography in the 21st Century: Lexicography from Scratch. Proceedings of the eLex 2017 conference*, (pp. 19–21).
- Mihalcea, R. (2007). Using Wikipedia for Automatic Word Sense Disambiguation. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, (pp. 196–203).

- Miháltz, M., Hatvani, C., Kuti, J., Szarvas, G., Csirik, J., Prószéky, G., and Váradi, T. (2008). Methods and Results of the Hungarian WordNet Project. In *Proceedings of The Fourth Global WordNet Conference*, (pp. 311–321). Szeged, Hungary.
- Mikhailov, M. and Cooper, R. (2016). *Corpus Linguistics for Translation and Contrastive Studies: A Guide for Research*. New York: Routledge.
- Mikolov, T., Chen, K., Corrado, G. S., and Dean, J. (2013). Efficient Estimation of Word Representations in Vector Space. In *International Conference on Learning Representations*.
- Miller, G. A. (1995). WordNet: A Lexical Database for English. *Communications of the ACM*, 38(11):39–41.
- Morin, E. and Prochasson, E. (2011). Bilingual Lexicon Extraction from Comparable Corpora Enhanced with Parallel Corpora. In *Proceedings of the 4th Workshop on Building and Using Comparable Corpora: Comparable Corpora and the Web*, (pp. 27–34).
- Mosel, U. (2011). Lexicography in endangered language communities. In Austin, P. K. and Sallabank, J. (Eds.) *The Cambridge Handbook of Endangered Languages*, Cambridge Handbooks in Language and Linguistics, (p. 337–353). Cambridge: Cambridge University Press.
- Moseley, C. (Ed.) (2010). *Atlas of the World's Languages in Danger*. Paris, France: UNESCO Publishing.
- Moshagen, S., Pirinen, T. A., and Trosterud, T. (2013). Building an open-source development infrastructure for language technology projects. In *Proceedings of the 19th Nordic Conference of Computational Linguistics (NODALIDA 2013)*, (pp. 343–352).
- Murakami, Y. (2019). Indonesia Language Sphere: an Ecosystem for Dictionary Development for Low-Resource Languages. *Journal of Physics: Conference Series*, 1192(1).
- Muzny, G. and Zettlemoyer, L. (2013). Automatic Idiom Identification in Wiktionary. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, (pp. 1417–1421).

- Máté, J. (1999). A magyar nyelv elsajátításának nehézségei a finn anyanyelvű tanulók szempontjából [The Difficulties of Learning the Hungarian Language from the Point of View of Finnish Native Speakers]. *Hungarologische Beiträge*, 12:91–112.
- Nation, I. S. P. (1980). Strategies for Receptive Vocabulary Learning. *Guidelines*, 3(1):18–23.
- Nation, I. S. P. (2006). How Large a Vocabulary Is Needed For Reading and Listening? *The Canadian Modern Language Review*, 63(1):59–82.
- Nation, I. S. P. and Waring, R. (1997). Vocabulary Size, Text Coverage and Word Lists. In Schmitt, N. and McCarthy, M. (Eds.) *Vocabulary: Description, Acquisition and Pedagogy*, (pp. 6–19). Cambridge: Cambridge University Press.
- Nerbonne, J. (2005). Natural Language Processing in Computer-Assisted Language Learning. In *The Oxford Handbook of Computational Linguistics*.
- Nguyen, M. V., Lai, V., Veyseh, A. P. B., and Nguyen, T. H. (2021). Trankit: A Light-Weight Transformer-based Toolkit for Multilingual Natural Language Processing. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*.
- Orasmaa, S., Petmanson, T., Tkachenko, A., Laur, S., and Kaalep, H.-J. (2016). Estnltk - nlp toolkit for estonian. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, (pp. 2460–2466).
- Oravecz, C., Váradi, T., and Sass, B. (2014). The Hungarian Gigaword corpus. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, (pp. 1719–1723). Reykjavik, Iceland: European Language Resources Association (ELRA).
- Otero, P. G. (2007). Learning Bilingual Lexicons from Comparable English and Spanish Corpora. In *Proceedings of Machine Translation Summit XI: Papers*.
- Papp, I. (1962). *Finn-Magyar Szótár [Finnish–Hungarian Dictionary]*. Budapest: Akadémiai Kiadó.

- Pennington, J., Socher, R., and Manning, C. D. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, (pp. 1532–1543).
- Pirinen, T. A. (2015). Development and Use of Computational Morphology of Finnish in the Open Source and Open Science Era: Notes on Experiences with Omorfi Development. *SKY Journal of Linguistics*, 28:381–393.
- Qi, P., Zhang, Y., Zhang, Y., Bolton, J., and Manning, C. D. (2020). Stanza: A Python Natural Language Processing Toolkit for Many Human Languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. Online: Association for Computational Linguistics.
- Qian, D. D. (2002). Investigating the Relationship between Vocabulary Knowledge and Academic Reading Performance: An Assessment Perspective. *Language learning*, 52(3):513–536.
- Rapp, R. (1995). Identifying Word Translations in Non-Parallel Texts. In *Proceedings of the 33rd Annual Meeting on Association for Computational Linguistics, ACL '95*, (pp. 320–322). USA: Association for Computational Linguistics.
- Rezaeinia, S. M., Rahmani, R., Ghodsi, A., and Veisi, H. (2019). Sentiment Analysis Based on improved Pre-trained Word Embeddings. *Expert Systems with Applications*, 117:139–147.
- Richman, A. E. and Schone, P. (2008). Mining Wiki Resources for Multilingual Named Entity Recognition. In *Proceedings of ACL-08: HLT*, (pp. 1–9).
- Rundell, M. (2012). The Road to Automated Lexicography: An Editor’s Viewpoint. In Granger, S. and Paquot, M. (Eds.) *Electronic Lexicography*, (pp. 15–30).
- Russian Federal Service of State Statistics (Rosstat) (2021). Том 5. «Национальный состав и владение языками». Таблица 4. Владение языками и использование языков населением [Volume 5. ”National Composition and Language Skills”. Table 4. Language Proficiency and Use of Languages by the Population]. https://rosstat.gov.ru/storage/mediabank/Tom5_tab4_VPN-2020.xlsx. [Microsoft Excel spreadsheet; accessed May 26, 2023].

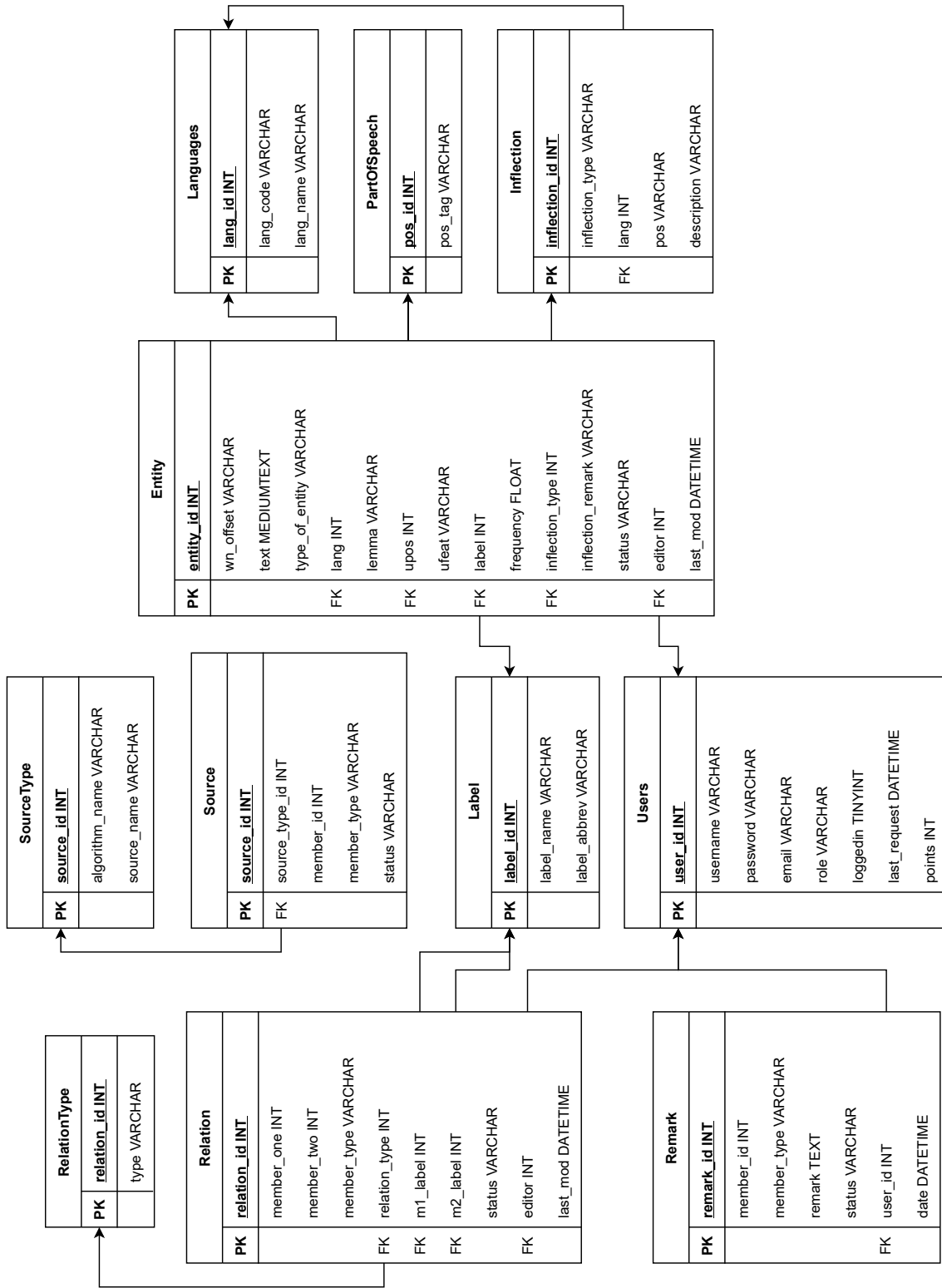
- Saralegi, X., Manterola, I., and San Vicente, I. (2011). Analyzing Methods for Improving Precision of Pivot Based Bilingual Dictionaries. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, (pp. 846–856).
- Schmitt, N. (1998). Tracking the incremental acquisition of second language vocabulary: A longitudinal study. *Language Learning*, 48(2):281–317.
- Sharifi, M., Azizifar, A., Jamalinesari, A., and Gowhary, H. (2015). The Effect of Rosetta Stone Computer Software on Vocabulary Learning of Iranian Elementary EFL Learners. *Procedia - Social and Behavioral Sciences*, 192:260–266. The Proceedings of 2nd Global Conference on Conference on Linguistics and Foreign Language Teaching.
- Shirai, S. and Yamamoto, K. (2001). Linking English Words in Two Bilingual Dictionaries to Generate Another Language Pair Dictionary. In *Proceedings of ICCPOL*, (pp. 174–179).
- Siitonen, K. and Wessel, K. A. (2020). Finnish Language and Culture in German Universities. *Finnish-German Yearbook of Political Economy, Volume 2*, (p. 95).
- Simon, E., Benyeda, I., Koczka, P., and Ludányi, Zs. (2015). Automatic Creation of Bilingual Dictionaries for Finno-Ugric Languages. In *1st International Workshop on Computational Linguistics for Uralic Languages*. Tromsø, Norway.
- Simon, E. and Mittelholcz, I. (2017). Evaluation of Dictionary Creating Methods for Under-Resourced Languages. In *International Conference on Text, Speech, and Dialogue*, (pp. 246–254). Prague, Czech Republic: Springer.
- Sinclair, J. (1985). Lexicographic Evidence. In Ilson, R. F. (Ed.) *Dictionaries, Lexicography and Language Learning*, (pp. 81–94). New York: Pergamon Press.
- Sinclair, J. (1991). *Corpus, Concordance, Collocation*. Oxford: Oxford University Press.
- Sinclair, J. and Renouf, A. (1988). A Lexical Syllabus for Language Learning. In Carter, R. and McCarthy, M. (Eds.) *Vocabulary and Language Teaching*, (pp. 140–160). London: Longman.
- Sjöbergh, J. (2005). Creating a Free Digital Japanese-Swedish Lexicon. In *Proceedings of PALING*, (pp. 296–300). Tokyo, Japan.

- Søgaard, A., Ruder, S., and Vulić, I. (2018). On the Limitations of Unsupervised Bilingual Dictionary Induction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, (pp. 778–788). Melbourne, Australia: Association for Computational Linguistics.
- Stamenkovska, T., Llerena, C. L. A., and Györi, J. G. (2022). Exploring the Motivation of International Students to Learn Hungarian: A Qualitative Study. *Hungarian Educational Research Journal*, 12(2):213 – 228.
- Steingrímsson, S., Loftsson, H., and Way, A. (2021). PivotAlign: Leveraging High-Precision Word Alignments for Bilingual Dictionary Inference. In *Proceedings of TIAD-2021 Shared Task – Translation Inference Across Dictionaries co-located with the 4th Language, Data and Knowledge Conference (LDK 2021)*. Zaragoza, Spain.
- Szinnyei, J. (1884). *Finn–magyar szótár [Finnish–Hungarian Dictionary]*. Budapest, Hungary: Magyar Tudományos Akadémia.
- Tanaka, K. and Umemura, K. (1994). Construction of a Bilingual Dictionary Intermediated by a Third Language. In *The 15th International Conference on Computational Linguistics*, (pp. 297–303). Kyoto, Japan: The Association for Computational Linguistics.
- Tang, H.-W. V., Yin, M.-S., and Lou, P.-J. (2009). A Meta-analytic Review of Current CALL Research on Second Language Learning. In *2009 IEEE International Symposium on IT in Medicine & Education*, volume 1, (pp. 677–684).
- Tavast, A., Langemets, M., Kallas, J., and Koppel, K. (2018). Unified Data Modelling for Presenting Lexical Data: The Case of EKILEX. In Čibej, J., Gorjanc, V., Kosem, I., and Krek, S. (Eds.) *Proceedings of the XVIII EURALEX International Congress: Lexicography in Global Contexts*, (pp. 749–761). Ljubljana, Slovenia: Ljubljana University Press, Faculty of Arts.
- Teubert, W. (1996). Comparable or Parallel Corpora? *International Journal of Lexicography*, 9(3):238–264.
- Thomas, M., Reinders, H., and Warschauer, M. (Eds.) (2012). *Contemporary Computer-Assisted Language Learning*. London/New York: Bloomsbury.

- Thorndike, R. L. (1973). *Reading Comprehension Education in Fifteen Countries: An Empirical Study*. New York: Wiley.
- Tiedemann, J. and Nygaard, L. (2004). The OPUS Corpus - Parallel and Free. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*. Lisbon, Portugal.
- Tkachenko, A., Petmanson, T., and Laur, S. (2013). Named entity recognition in estonian. In *Proceedings of the 4th Biennial International Workshop on Balto-Slavic Natural Language Processing*, (pp. 78–83).
- Trujillo, J. P. C., Mohammed, P. J., and Saleh, S. T. (2020). Students' Motivations to Study Abroad: The Case of International Students at the University of Debrecen. *Central European Journal of Educational Research*, 2(1):76–81.
- Turian, J., Ratinov, L., and Bengio, Y. (2010). Word Representations: a Simple and General Method for Semi-Supervised Learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, (pp. 384–394). Uppsala, Sweden: Association for Computational Linguistics.
- Uibo, H., Pruulmann-Vengerfeldt, J., Rueter, J., and Iva, S. (2015). Oahpa! Õpi! Opiq! Developing free online programs for learning Estonian and Võro. In *Proceedings of the Fourth Workshop on NLP for Computer-Assisted Language Learning*, (pp. 51–64).
- UNESCO (2003). *Language Vitality and Endangerment*. Document submitted to the International Expert Meeting on the UNESCO Programme Safeguarding of Endangered Languages. Paris, 10–12 March 2003.
- Varga, I. and Yokoyama, S. (2009). Bilingual Dictionary Generation for Low-resourced Language Pairs. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, (pp. 862–870). Singapore: Association for Computational Linguistics.
- Vincze, V., Nagy, Á., Horváth, C., Szilágyi, N., Kozmács, I., Bogár, E., and Fenyvesi, A. (2015). FinUgRevita: Developing Language Technology Tools for Udmurt and Mansi. *Septentrio Conference Series*, (2):108–118.

- Waring, R. and Nation, I. S. P. (2004). Second Language Reading and Incidental Vocabulary Learning. *Angles on the English Speaking World*, 4:97–110.
- Warschauer, M. (2000). The Death of Cyberspace and the Rebirth of CALL. *English Teachers' Journal*, 53(1):61–67.
- Warschauer, M. and Healey, D. (1998). Computers and Language Learning: An Overview. *Language Teaching*, 31(2):57–71.
- Weschler, R. and Pitts, C. (2000). An Experiment Using Electronic Dictionaries with EFL Students. *The Internet TESL Journal*, 6(8):56–67.
- Wilkins, D. A. (1972). *Linguistics in Language Teaching*. London: Edward Arnold.
- Wu, D. and Xia, X. (1994). Learning an English-Chinese Lexicon from a Parallel Corpus. In *Proceedings of the First Conference of the Association for Machine Translation in the Americas*, (pp. 206–213). Columbia, Maryland, USA.
- Zhang, Y. L. (2018). A Study on Motivation of Chinese Students Learning Hungarian in Hungary. Master's thesis, ELTE BTK Angol-Amerikai Intézet.

Appendix A Entity Relationship Diagram of the Database



Appendix B Resources and tools created during this research

WORDNET CONNECTOR

- Algorithm that links the Finnish and Hungarian WordNets
- Extracts bilingual translation pairs with additional information
- Extracts monolingual data such as synonym pairs, example sentences and definitions
- Detailed information in Section 2.5.1
- Licensed under the GNU AGPL v3.0 License, META-SHARE Commons BY NC ND License v1.0 and Creative Commons Attribution (CC-BY) 3.0 License.
- https://github.com/ferenczizsani/connect_wordnets

WIKTIONARY PARSER

- Algorithm that extracts bilingual translation equivalents, example sentences and definitions from the Finnish and Hungarian Wiktionaries
- Detailed information in Section 2.5.2
- Licensed under the GNU AGPL v3.0 License.
- https://github.com/ferenczizsani/wiktionary_parser

OPUS EXTRACTOR

- Algorithm that extracts bilingual word pairs from the OPUS word alignments and filters the data
- Detailed information in Section 2.5.3
- Licensed under the GNU AGPL v3.0 License.
- https://github.com/ferenczizsani/opus_extractor

FINNISH–HUNGARIAN RESOURCES

- Finnish and Hungarian data collected in this research and validated by speakers of Finnish and Hungarian
- Lists of bilingual translation pairs, monolingual synonym pairs, example sentences and definitions
- Detailed information in Section 2.6

- Licensed under the Creative Commons Attribution (CC-BY) 3.0 License.
- https://github.com/ferenczizsani/fin_hun_resources

FULR (FINNO-UGRIC LEXICAL RESOURCES)

- Online bilingual dictionary, dictionary writing system and language learning application for Finnish and Hungarian
- Detailed information in Chapter 5
- <https://fulr.btk.ppke.hu>

Appendix C Contents of the Inflection Table

inflection_id	inflection_type	lang	pos	description
1	NULL	NULL	NULL	NULL
2	1	1	NOUN, ADJ, NUM, PROP	valo
3	2	1	NOUN, ADJ, NUM, PROP	palvelu
4	3	1	NOUN, ADJ, NUM, PROP	valtio
5	4	1	NOUN, ADJ, NUM, PROP	laatikko
6	5	1	NOUN, ADJ, NUM, PROP	risti
7	6	1	NOUN, ADJ, NUM, PROP	paperi
8	7	1	NOUN, ADJ, NUM, PROP	ovi
9	8	1	NOUN, ADJ, NUM, PROP	nalle
10	9	1	NOUN, ADJ, NUM, PROP	kala
11	10	1	NOUN, ADJ, NUM, PROP	koira
12	11	1	NOUN, ADJ, NUM, PROP	omena
13	12	1	NOUN, ADJ, NUM, PROP	kulkija

14	13	1	NOUN, ADJ, NUM, PROPN	katiska
15	14	1	NOUN, ADJ, NUM, PROPN	solakka
16	15	1	NOUN, ADJ, NUM, PROPN	korkea
17	16	1	NOUN, ADJ, NUM, PROPN	vanhempi
18	17	1	NOUN, ADJ, NUM, PROPN	vapaa
19	18	1	NOUN, ADJ, NUM, PROPN	maa
20	19	1	NOUN, ADJ, NUM, PROPN	suo
21	20	1	NOUN, ADJ, NUM, PROPN	filee
22	21	1	NOUN, ADJ, NUM, PROPN	rosé
23	22	1	NOUN, ADJ, NUM, PROPN	parfait
24	23	1	NOUN, ADJ, NUM, PROPN	tiili
25	24	1	NOUN, ADJ, NUM, PROPN	uni
26	25	1	NOUN, ADJ, NUM, PROPN	toimi
27	26	1	NOUN, ADJ, NUM, PROPN	pieni

28	27	1	NOUN, ADJ, NUM, PROPN	käsi
29	28	1	NOUN, ADJ, NUM, PROPN	kynsi
30	29	1	NOUN, ADJ, NUM, PROPN	lapsi
31	30	1	NOUN, ADJ, NUM, PROPN	veitsi
32	31	1	NOUN, ADJ, NUM, PROPN	kaksi
33	32	1	NOUN, ADJ, NUM, PROPN	sisar
34	33	1	NOUN, ADJ, NUM, PROPN	kytkin
35	34	1	NOUN, ADJ, NUM, PROPN	onneton
36	35	1	NOUN, ADJ, NUM, PROPN	lämmin
37	36	1	NOUN, ADJ, NUM, PROPN	sisin
38	37	1	NOUN, ADJ, NUM, PROPN	vasen
39	38	1	NOUN, ADJ, NUM, PROPN	nainen
40	39	1	NOUN, ADJ, NUM, PROPN	vastaus
41	40	1	NOUN, ADJ, NUM, PROPN	kalleus

42	41	1	NOUN, ADJ, NUM, PROPN	vieras
43	42	1	NOUN, ADJ, NUM, PROPN	mies
44	43	1	NOUN, ADJ, NUM, PROPN	ohut
45	44	1	NOUN, ADJ, NUM, PROPN	kevät
46	45	1	NOUN, ADJ, NUM, PROPN	kahdeksas
47	46	1	NOUN, ADJ, NUM, PROPN	tuhat
48	47	1	NOUN, ADJ, NUM, PROPN	kuollut
49	48	1	NOUN, ADJ, NUM, PROPN	hame
50	49	1	NOUN, ADJ, NUM, PROPN	askel
51	50	1	NOUN, ADJ, NUM, PROPN	isoäiti
52	51	1	NOUN, ADJ, NUM, PROPN	nuoripari
53	52	1	VERB	sanoa
54	53	1	VERB	muistaa
55	54	1	VERB	huutaa
56	55	1	VERB	soutaa
57	56	1	VERB	kaivaa
58	57	1	VERB	saartaa
59	58	1	VERB	laskea

60	59	1	VERB	tuntea
61	60	1	VERB	lähteä
62	61	1	VERB	sallia
63	62	1	VERB	voida
64	63	1	VERB	saada
65	64	1	VERB	juoda
66	65	1	VERB	käydä
67	66	1	VERB	rohkaista
68	67	1	VERB	tulla
69	68	1	VERB	tupakoida
70	69	1	VERB	valita
71	70	1	VERB	juosta
72	71	1	VERB	nähdä
73	72	1	VERB	vanheta
74	73	1	VERB	salata
75	74	1	VERB	katketa
76	75	1	VERB	selvitä
77	76	1	VERB	taitaa
78	77	1	VERB	kumajaa
79	78	1	VERB	kaikaa
80	99	1	NULL	Sana on taipumaton tai vaillinaisesti taipuva
81	101	1	PRON	Pronominit
82	1	2	NOUN, ADJ, PRON, NUM	változatlan magánhangzós tövű főnevek
83	1	2	VERB	iktelen alapminták
84	2	2	NOUN, ADJ, PRON, NUM	nyílt kötőhangzójú mássalhangzós tövű főnevek

85	2	2	VERB	iktelenek elhangzós múltidőjellel
86	3	2	NOUN, ADJ, PRON, NUM	zárt kötőhangzójú mássalhangzós tövű főnevek
87	3	2	VERB	feltételes mód, főnévi igenév előhangzóval
88	4	2	NOUN, ADJ, PRON, NUM	zárt kötőhangzós főnevek puszta tárgyraggal
89	4	2	VERB	a felszólító mód nem j-vel
90	5	2	NOUN, ADJ, PRON, NUM	többelseji időtartamot váltakoztató főnevek
91	5	2	VERB	a felszólító mód más tőváltozatból
92	6	2	NOUN, ADJ, PRON, NUM	tővégi magánhangzót váltakoztató főnevek
93	6	2	VERB	hangzóhiányos változatú l végűek
94	7	2	NOUN, ADJ, PRON, NUM	mássalhangzós végű hangzóhiányos tőváltozatú főnevek
95	7	2	VERB	hangzóhiányos változatú g, r, ll végűek
96	8	2	NOUN, ADJ, PRON, NUM	v-vel bővülő tövű főnevek
97	8	2	VERB	hangzóhiányos változatú z végűek
98	9	2	NOUN, ADJ, PRON, NUM	rendhagyó főnevek
99	9	2	VERB	rendhagyó és hiányos iktelenek

100	10	2	NOUN, ADJ, PRON, NUM	kettős ragozású főnevek
101	10	2	VERB	kettős ragozású iktelenek
102	11	2	NOUN, ADJ, PRON, NUM	változatlan magánhangzós tövű melléknevek
103	11	2	VERB	puszta toldalékkal ragozott ikések
104	12	2	NOUN, ADJ, PRON, NUM	nyílt kötőhangzójú mássalhangzós tövű melléknevek
105	12	2	VERB	ikések előhangzós múltidő-jellel
106	13	2	NOUN, ADJ, PRON, NUM	zárt kötőhangzójú mássalhangzós tövű melléknevek
107	13	2	VERB	előhangzós feltételes módjelű ikések
108	14	2	NOUN, ADJ, PRON, NUM	mássalhangzós tövű melléknevek puszta tárgyraggal
109	14	2	VERB	felszólító módjukban nem j-s ikések
110	15	2	NOUN, ADJ, PRON, NUM	mássalhangzós melléknevek, fakultatív előhangzójú tárgyrag
111	15	2	VERB	felsz-ban módosult szótári tövű ikések
112	16	2	NOUN, ADJ, PRON, NUM	tővégi magánhangzót váltakoztató melléknevek

113	16	2	VERB	hangzóhiányos változatú l-tövű ikések
114	17	2	NOUN, ADJ, PRON, NUM	magánhangzós tövű, ingadozó többesű vagy rendhagyó fokozású melléknevek
115	17	2	VERB	egyéb hangzóhiányos tövű, valamint hiányos ragozású ikések
116	18	2	NOUN, ADJ, PRON, NUM	magánhangzós tövű, előhangzós többesszámú melléknevek
117	18	2	VERB	hangzóhiányos z-tövű ikések
118	19	2	NOUN, ADJ, PRON, NUM	váltakozó tövű, rendhagyó vagy hiányos melléknevek
119	19	2	VERB	rendhagyó ikések
120	20	2	NOUN, ADJ, PRON, NUM	kettős toldalékolású melléknevek
121	21	2	NOUN, ADJ, PRON, NUM	változatlan magánhangzós tövű névmások, számnevek, hiányos főnevek
122	22	2	NOUN, ADJ, PRON, NUM	nyílt kötőhangzós névmások, számnevek, hiányos főnevek
123	23	2	NOUN, ADJ, PRON, NUM	n23
124	24	2	NOUN, ADJ, PRON, NUM	n24
125	25	2	NOUN, ADJ, PRON, NUM	n25

126	26	2	NOUN, ADJ, PRON, NUM	n26
127	27	2	NOUN, ADJ, PRON, NUM	n27
128	28	2	NOUN, ADJ, PRON, NUM	n28
129	29	2	NOUN, ADJ, PRON, NUM	n29
130	30	2	NOUN, ADJ, PRON, NUM	n30
131	31	2	NOUN, ADJ, PRON, NUM	n31
132	32	2	NOUN, ADJ, PRON, NUM	n32
133	33	2	NOUN, ADJ, PRON, NUM	n33
134	34	2	NOUN, ADJ, PRON, NUM	n34
135	35	2	NOUN, ADJ, PRON, NUM	n35
136	36	2	NOUN, ADJ, PRON, NUM	n36
137	20	2	VERB	v20
138	21	2	VERB	v21
139	22	2	VERB	v22
140	23	2	VERB	v23
141	24	2	VERB	v24
142	25	2	VERB	v25
143	26	2	VERB	v26

144	27	2	VERB	v27
145	28	2	VERB	v28
146	29	2	VERB	v29
147	30	2	VERB	v30
148	31	2	VERB	v31
149	32	2	VERB	v32
150	33	2	VERB	v33
151	34	2	VERB	v34
152	35	2	VERB	v35
153	36	2	VERB	v36

Appendix D List of User Roles and Permissions

Action	Admin	Editor	Teacher	Guest	Player
View and use online dictionary interface	✓	✓	✓	✓	✓
View and use language learning app	✓	✓	✓	✓	✓
View user profile	✓	✓	✓	✓	✓
View and use dictionary writing system	✓	✓	✓	✓	
View list of relations and entities to be validated	✓	✓	✓	✓	
Search among entities and relations to be validated	✓	✓	✓	✓	
Search among validated entities and relations	✓	✓	✓	✓	
Filter validated entities and relations according to status, language, part of speech and type	✓	✓	✓	✓	
Search among all the entities and relations in the database	✓	✓	✓		
View entities and relations in detail	✓	✓	✓		
Add remarks to entities and relations	✓	✓	✓		
Validate sentence analyses	✓	✓	✓		
Edit entities and relations	✓	✓			
Merge entities	✓	✓			
Approve validated entities and relations	✓				
Modify the role of users	✓				
Delete users	✓				

Magyar nyelvű összefoglaló

A disszertáció két finnugor nyelvet vizsgált lexikográfiai szempontból. A finnugor nyelvek elsajátítása nagy kihívást jelent, még azok számára is, akiknek anyanyelve ugyanehhez a nyelvcsaládhoz tartozik. A lexikai erőforrások kézi összeállítása drága és időigényes tevékenység. Nem meglepő tehát, hogy a szabadon elérhető online szótárak többsége vagy alacsony minőségű, vagy túlságosan kis fedéssel rendelkezik. Annak ellenére, hogy a finn és a magyar a két legtöbb beszélővel rendelkező finnugor nyelv, nem létezik olyan jó minőségű online erőforrás, melyet a fenti nyelvek elsajátításának megkönnyítése céljából hoztak volna létre. A kutatás során egy összetett keretrendszer kiépítése történt meg. A rendszer négy fő komponensből áll (adatbázis, szótáríró rendszer, online kétnyelvű szótár és nyelvtanuló alkalmazás), és célja, hogy felületet biztosítson azok számára, akik ezen két finnugor nyelv egyikét próbálják elsajátítani.

Automatikus szótárépítési módszerek alkalmazásával kétnyelvű fordításjelölteket és más lexikai információkat (példamondákat, definíciókat stb.) sikerült kinyerni különböző forrásokból. Mivel a finn és a magyar gazdag morfológiával rendelkezik, azok a hagyományos módszerek, melyeket az angolra és egyéb nagyobb nyelvekre fejlesztettek ki, nem biztosítanak megbízható eredményeket. Ebből fakadóan a disszertáció három új, alternatív módszert mutatott be, amelyek segítségével fordítási párok százezrei, valamint további lexikai információk generálhatók finnre és magyarra. Kimutatható továbbá, hogy ezen nyelvek természetesnyelv-feldolgozó eszközei és eljárásai (például lemmatizálás) pozitívan befolyásolják a kétnyelvű szótárak minőségét.

A fenti módszerekkel kinyert adathalmaz eltárolása egy relációs adatbázis segítségével történik. Ez az adatbázis nyelvfüggetlen módon képes tárolni a lexikai adatokat. A hagyományos lexikográfiában a szótárak alapvető egysége a szócikk, azonban a javasolt struktúrában a lexikai entitás (lemma, több szavas kifejezés vagy mondat) alkotja a szótár legkisebb egységét. Ez az újítás elősegítette egy kétnyelvű szótár automatikus létrehozását, melynek során nem volt szükség a szócikkek kézzel történő átdolgozására a két iránynak (finn–magyar és magyar–finn) megfelelően.

A szótárépítési módszerek kézi kiértékelése egy új szótáríró rendszerben történt, amelynek egyik célja megkönnyíteni a lexikográfiai munkát, illetve lehetővé tenni az entítások, valamint az entítások közötti kapcsolatok ellenőrzését, módosítását és törlését olyan szerkesztők számára is, akik nem rendelkeznek magas szintű számítástechnikai ismerettel.

Ennek a hibrid szótárépítési metodológiának a segítségével sikerült lecsökkenteni a szótárkészítési munka költségeit és a folyamat időtartamát, míg a szótárban szereplő adatok továbbra is jó minőségűek maradtak.

A szótár célközönsége számára elkészült egy további, nyelvtanulást elősegítő komponens is. Az adatbázisban szereplő, kézi validáláson átesett adatokat digitális szókérdőív alakítva biztosítható a nyelvtanulók szókincsének fejlesztése. A kinyert példamondatok segítségével behelyettesítési feladatok is generálhatók. A nyelvtanuló alkalmazás célja egy olyan platform létrehozása, amelyen a nyelvtanulók a finn és a magyar nyelv, illetve nyelvtan legnehezebb egységeit gyakorolhatják. A disszertáció egy erre kidolgozott, automatikus folyamatot mutat be, amely a nyelvtanárokat hivatott mentesíteni a feladatok kézi összeállítása és ellenőrzése alól.

Summary in English

This dissertation provided a detailed analysis of two Finno-Ugric languages from a lexicographic perspective. Learning a Finno-Ugric language is a great challenge, even for those whose native language belongs to the same language family. Manually constructed lexical resources are expensive and time-consuming. As a consequence, the majority of dictionaries available online usually have low quality or their coverage is insufficiently small. Even though Finnish and Hungarian are the two most spoken Finno-Ugric languages, they also lack high quality online resources. Therefore, a complex framework was proposed in this thesis. It consists of four main components (a database, a dictionary writing system, an online bilingual dictionary, and a language learning application), and aims to provide a platform for learners of these languages.

Automatic dictionary building methods were applied to Finnish and Hungarian in order to extract bilingual translation candidates and other lexical information. Since these languages are rich in inflectional morphology, traditional methods that are generally developed for English and other widely spoken languages do not provide reliable results in this case. For this reason, three alternative methods were proposed which led to hundreds of thousands of translation pairs, as well as additional lexicographic information (such as example sentences and definitions). It was found that language processing methods, such as lemmatization, have a positive impact on the resulting proto-dictionaries in the case of these Finno-Ugric languages.

In order to store the obtained data set, a relational database was constructed. This database stores lexical data in a language-independent way. In traditional lexicography, the central unit of a lexicon is a dictionary entry, however, in the proposed structure the basic unit was defined as a lexical entity (a lemma, a multi-word expression, or a sentence). This innovation makes it possible to compile automatically reversible dictionaries from the previously extracted data set. The schema of the database facilitated the automatic compilation of a bidirectional learner's dictionary for Finnish and Hungarian.

The evaluation of the dictionary building methods was carried out in a new dictionary writing system, which aids lexicographic work and enables the editors to correct, modify and delete entities, as well as relations between the entities, without advanced IT knowledge.

With the help of this hybrid dictionary building methodology, it was possible to reduce the cost and required time of dictionary compilation, while maintaining the high quality of the data present in the dictionary.

It had been observed that a language learning interface for the same target audience could be of great value. For this reason, a further component was developed with the help of the same database. A great number of digital flashcards could be automatically created to facilitate vocabulary acquisition in Finnish and Hungarian. Furthermore, thousands of example sentences were converted into fill-in-the-blank exercises. The aim of this language learning application is to provide high-quality examples which can help learners practice the most challenging areas of Finnish and Hungarian without requiring language instructors to manually construct these tasks.