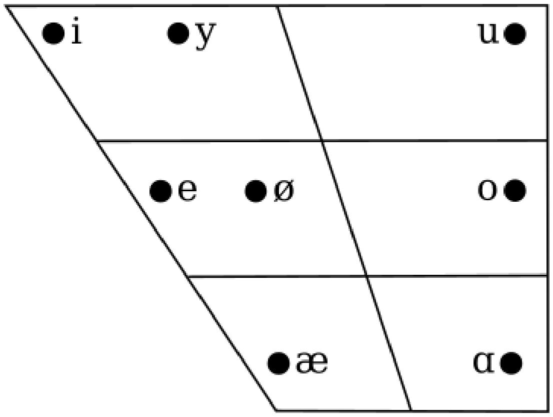
Computational Analysis of Finnish Inflection Patterns

Introduction

Finnish is a Uralic language spoken primarily in Finland, with native speakers in Sweden, Norway, Russia, Estonia, the US, and Australia.[1] In Finland, approximately 4.9 million people speak Finnish as their first language. Standard Finnish is one of the two primary registers, and is the primary form used in writing and official contexts. Standard Finnish is the form used for the purposes of this paper.



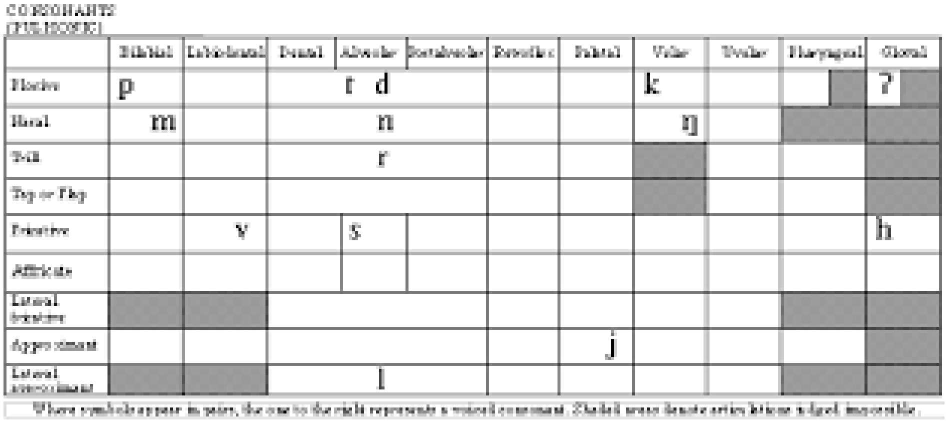
[2]

Vowels

Finnish has 8 distinct monophthongs (single vowels), {/æ/, /e/, /i/, /ø/, /y/, /ɑ/, /o/, /u/}. A doubled form that is phonologically distinct occurs for each monophthong: {/æ:/, /e:/, /i:/, /ø:/, /y:/, /ɑ:/, /o:/, /u:/}. These long vowels are pronounced the same as their short counterparts but are held for a longer duration. The vowels are classified in two ways: the height classes of high {/i/, /y/, /u/}, mid {/e/, /ø/, /o/}, and low {/ɑ/, /æ/}; and the harmonic classes of front harmonic {/y/, /ø/, /æ/}, back harmonic {/u/, /o/, /ɑ/}, and neutral {/i/, /e/}. Harmonic classification is incredibly important as this is a major factor in determining the correct form of inflection for many words.

[1] The Institute for the Languages of Finland. (n.d.). *Languages of finland - institute for the languages of Finland*. Kotimaisten kielten keskus. https://www.kotus.fi/en/on_language/languages_of_finland#Finnish

[2] Suomi, K., Toivanen, J., & Ylitalo, R. (2008). *Finnish sound structure: Phonetics, phonology, phonotactics and Prosody*. University of Oulu. p.20-23



CONSONANTS (PULMONIC)

	Bilabial	Labiodental	Dental	Alveolar	Postalveolar	Retroflex	Palatal	Velar	Uvular	Pharyngeal	Glottal
Plosive	p			t d				k		▨	ʔ ▨
Nasal	m			n				ŋ		▨	▨
Trill				r				▨			▨
Tap or Flap								▨			▨
Fricative		v		s							h
Affricate											
Lateral fricative	▨	▨								▨	▨
Approximant							j				▨
Lateral approximant	▨	▨		l						▨	▨

Where symbols appear in pairs, the one to the right represents a voiced consonant. Shaded areas denote articulations judged impossible.

[3]

Consonants

There are 11 broadly accepted consonant phonemes in Finnish, with 6 others having various degrees of use or distinction among speakers. {/p/, /t/, /k/, /s/, /h/, /m/, /n/, /l/, /r/, /ʋ/, /j/} are the 11 broadly accepted consonant phonemes; /ŋ/ and /d/ occur natively but are exclusively due to consonant gradation; /f/, /ʃ/, /b/, and /g/ are non-native and occur in loan words.

For the most part, Finnish is a phonetic language, which is to say that Finnish words are pronounced very closely to how they are spelled. This fact makes it relatively straightforward to analyze the phonemic composition of Finnish words as the dictionary spelling could be used as the phonemic transcription with very few exceptions.

Vowel Harmony

Like several other Uralic languages, a major phenomenon in Finnish vowel phonotactics is vowel harmony. As stated previously, Finnish vowels can be put into three classes according to the distinctive feature [back]: front harmonic, back harmonic, or harmonically neutral.[4] The front vowels have [-back] while the back vowels have [+back]. Each vowel in these groups can be paired up according to the other features [high], [low], and [round]:

[-back]	[+back]	
/y/	/u/	[+high][-low][+round]
/ø/	/o/	[-high][-low][+round]
/æ/	/ɑ/	[-high][+low][-/+round]

The neutral vowels both have the feature [-back], but lack a counterpart in the other distinctive features, resulting in them having a group of their own.

[3] Lyovin, A., & Raimo, I. (n.d.). *Native Phonetic Inventory: finnish*. Speech accent archive: Browse. https://accent.gmu.edu/browse_native.php?function=detail&languageid=20

[4] Suomi, K., Toivanen, J., & Ylitalo, R. (2008). *Finnish sound structure: Phonetics, phonology, phonotactics and Prosody*. University of Oulu. p.51,52

The vowel harmony rule governs which vowels can be used within a word. If a word is uncompounded, vowels from the front and back classes may not co-occur. Neutral vowels may be mixed with front or back vowels in a word, or may be present on their own.[5]

Vowel harmony becomes very important when determining word inflection. As a suffixal agglutinative language, word stems may be inflected by adding suffixes to alter the meaning. Any suffix containing a vowel has two forms: one front and one back. To adhere to vowel harmony, the suffix used to inflect a word must match the vowel class of the stem. For instance:

/kulkijɑ/ "wanderer" → */kulkijɑnɑ/* "as a wanderer"

or */seinæ/* "wall" → */seinænæ/* "as a wall"

In this example, */kulkijɑ/* has back harmony, so the back form of the essive suffix /-*nɑ*/ is used. In contrast, */seinæ/* has front harmony so the front form of the essive suffix /-*næ*/ is used. There are words that have only neutral vowels. In this case, the front form would be used:

/kehite/ "developer" -> */kehitenæ/* "in development"

Another abnormal case is in compound words and loanwords that are relatively new to the Finnish language, or even some rare cases among native words. These words may be "disharmonic", or contain vowels of both front and back classes. When this occurs, the final vowel preceding the suffix is typically used to determine the harmonic class used. For instance:

/mæntu/ "pine tree" -> */mæntunɑ/* "as a pine tree"

Since the final vowel in */mæntu/* is a back vowel, the back suffix is used.

Topic of Research

Vowel harmony is a major factor in determining the inflected form of many words, and scrutinizing the vowels in a word stem is a concrete method for determining the harmonic class of the suffixes to be attached. However, many cognitive processes operate using heuristics formed from trial and error to produce outputs (motor reactions, thoughts, speech) quicker, with the tradeoff of less precise results. The goal of this paper is to explore possible heuristics that may be used to determine the harmonic class of suffixes used to inflect different Finnish nouns. Specifically, I will be looking at word initial phonemes and if there are any associations between specific initial phonemes and either of the vowel harmony classes. I will perform statistical analysis of the phonemic transcriptions of Finnish words to uncover any potential associations. This will be done using Python, an open source programming language that is increasingly popular in scientific and engineering applications because of its many user-developed libraries

[5] Suomi, K., Toivanen, J., & Ylitalo, R. (2008). *Finnish sound structure: Phonetics, phonology, phonotactics and Prosody*. University of Oulu. p.51,52

that provide high-level programming language functionality implemented with low-level compiled language performance.

Data

The subject of this project is a dataset sourced from Kielipankki - The language bank of Finland.[6] It is a file contained within the FinnWordNet data package, containing the index information for the FinnWordNet dictionary. Organized according to the Princeton WordNet Database (PWD)[7] format, this dataset contains an alphabetized list of Finnish words with several fields of data:

```
a n 6 7 @ ~ #m #s #p %p ; 6 0 15224595 14842042 13755137 06874656 06874656 05437099
aakkonen n 1 4 @ ~ #m #p 1 0 06872314
```

However, the only data being used in the scope of this project is the word stem. Before use, the dataset was preprocessed with a Python script to remove unnecessary information and preserve the words used in analysis. The preprocessing included removing the first 29 lines of text which contained copyright information, then recording the word stem from each entry. This was done by reading each line, splitting the strings on whitespace characters and then writing the first string into its own line in another file:

```
for line in file_in:
   # Split the line into words
   words = line.split()
   # If the first word is a single character, skip the line
   if len(words[0]) == 1:
       continue
   word_count += 1
   # Otherwise, write the first word to the output file
   file_out.write(words[0] + '\n')
```

The remaining word list contained 107,662 Finnish nouns. Of these words, 41,717 were later discarded in the main program. Reasons for discarding were either that the word contained no vowels (i.e. it was an abbreviation) or it contained invalid characters/phonemes, such as punctuation marks or phonemes not recognized in Finnish phonology. After this preprocessing, 65,945 valid strings remained for analysis.

[6] The Language Bank of Finland. (2019, September 16). *FinnWordNet – the Finnish wordnet*. Kielipankki. https://www.kielipankki.fi/corpora/finnwordnet/

[7] The Trustees of Princeton University. (2023). *WNDB(5WN)* | *wordnet*. Princeton University. https://wordnet.princeton.edu/documentation/wndb5wn

Analysis

All of the analysis performed on the noun index dataset was programmed in Python 3. The standard library was used, with the addition of the NumPy, Pandas, and SciPy libraries, all of which are open source projects available for free use by the public. NumPy provides tools for creating, manipulating, and performing operations on arrays and matrices.[8] Pandas provides a number of high-performance data structures and tools for analyzing data, such as the DataFrame structure which was a major component of the data analysis in this project.[9] SciPy provides tools for scientific computation and engineering applications.[10] It was primarily used for the statistical analysis functions provided in the stats sublibrary.

Primary Analysis

First, the index.noun file from the FinnWordNet-2.0 library was preprocessed in Python as described in the previous section. The resulting file is passed to the main program for analysis. After opening the file for reading, the lines are scanned in and organized into different groups for further organization. Each word is given a set of boolean flags as it is read in: **front**, **back**, **neutral**, and **discard**. The default state for each flag is ***False***. The program then looks at each individual character in the string and runs it through a four-way conditional filter which determines which flag to mark as ***True***:

```
for letter in word:
   if letter in front_vowels:
       front = True
       back = False
   elif letter in mid_vowels:
       neutral = True
   elif letter in back_vowels:
       back = True
       front = False
   elif (letter not in consonants) & (letter not in all_vowels):
       discard = True
```

First, if the character is included in the set of front vowels `{'ö', 'ä', 'y'}` then **front** would be ***True*** and **back** would be ***False***. Else, if the character is in the mid vowels `{'e', 'i'}` then **neutral** would be set to ***True***. Else, if the character is in the back vowels `{'a', 'o', 'u'}` then **back** would become ***True*** and **front** would become ***False***. Else, if the character is not included in the list of consonants `['d', 'g', 'h', 'j', 'k', 'l', 'm', 'n', 'p', 'r', 's', 't', 'v', 'š', 'f', 'b']` or the set of all vowels, then **discard** would be set to ***True***. Cases one and three are implemented in a way that assures that the final non-neutral vowel will be used to determine the harmonic class of the entire word. After running each character through

[8] *NumPy documentation#*. NumPy documentation - NumPy v1.24 Manual. (2022). https://numpy.org/doc/stable/

[9] *Pandas documentation#*. pandas documentation - pandas 2.0.1 documentation. (2023). https://pandas.pydata.org/docs/index.html

[10] *Scipy documentation#*. SciPy documentation - SciPy v1.10.1 Manual. (2023). https://docs.scipy.org/doc/scipy/

the conditional filter, the flag values are considered. The string is assigned a categorical label based on which flags it has set to ***True***:

```
if discard:
   category = 'Unknown'
elif front:
   category = 'Front'
elif back:
   category = 'Back'
elif neutral:
   category = 'Front'
else:
   category = 'Unknown'
```

If **discard** is ***True*** then the string is given the label "Unknown". Else, if **front** is ***True*** then the label is "Front", and "Back" for **back** respectively. If the only flag set to ***True*** is **neutral**, then the word is given the "Front" label, as ruled by the vowel harmony rules. If no label is set to true, then the word is given the "Unknown" label.[11] The order of these conditionals is important: it assures that invalid strings will be flagged before being mixed with valid strings, and it assures that front and back vowels will be given a higher priority when it comes to determining harmonic class. The string and its label are then appended to the ends of two lists, **words** for holding words and **categories** for holding categorical labels. Since they are appended at the same time, it can be assumed that their indexes will be the same, i.e. the words will align with their label as if they were two columns in a table.

A new file is opened for writing, and any word with the "Unknown" label is written into the file[12]. A DataFrame object, **db**, is initialized with headers "Word" and "Vowel Harmony", and the **words** and **categories** lists are assigned to these headers respectively. The back and front words are then split into two separate lists by **db** and their lengths are reported

```
df = pd.DataFrame({'Word': words, 'Vowel Harmony': categories})

front_harmony_words = df[df['Vowel Harmony'] == 'Front']['Word'].tolist()
back_harmony_words = df[df['Vowel Harmony'] == 'Back']['Word'].tolist()
```

There are 65,945 total usable words, 12,523 in the front harmony list and 53,422 in the back harmony list.

Another DataFrame is initialized, **initial_phoneme_contingency**, with the purpose of holding the frequencies of the initial phoneme of each word in the front and back harmony groups respectively:

[11] This condition may be redundant as it is expected that if none of the other flags have been applied, then the discard flag would have been applied in the previous conditional filter.

[12] For the purpose of manual inspection of words.

```
initial_phoneme_contingency = pd.DataFrame(index=['Front', 'Back'],
columns=consonants + ['vowel'], dtype=int).fillna(0)
```

The words in each list are iterated over and the cell corresponding with the harmony group and the initial character is incremented. If the initial character is a vowel, then a cell corresponding to a column labeled “Vowel” is incremented:

```
for word in front_harmony_words:
   if word[0] in consonants:
       for consonant in set(consonants):
           if consonant == word[0]:
               initial_phoneme_contingency.at['Front', consonant] += 1
   else:
       initial_phoneme_contingency.at['Front', 'vowel'] += 1
```

This is done to prevent the inherent difference between the back harmony group not containing any of the front vowels and vice versa, but still allows for the opportunity to see a difference in the occurrence of vowel-initial words between the groups.

At this point, all of the data necessary for this stage of analysis has been prepared and is ready for statistical analysis. A chi-squared test for independence was chosen, as this test will show whether two categorical variables are independent or whether they are dependent. Then, if the variables are found to be dependent, the contributions of each category can be scrutinized to determine which ones contribute the most to the dependence. This information combined with whether the observed frequencies are greater than or less than the expected frequencies can be used to find any potential patterns that could indicate whether a phoneme initial is more associated with one harmony group or the other.

Accordingly, a chi-squared test for independence was performed on the observed frequencies contingency table **initial_phoneme_contingency.** This was performed using the function chi2_contingency() from the scipy.stats library:

```
chi2_stat, p_val, dof, expected = chi2_contingency(initial_phoneme_contingency,
correction=True)
```

The function takes a contingency table as input and returns a chi-squared test statistic, p-value, degrees of freedom, and a table of the expected values. To determine if the test rejected the null hypothesis or not, the p-value was compared against an alpha of 0.05. The results of this initial test are the following:

Chi-square statistic: 132.1881249611558
p-value: 2.411015091798216e-20

Degrees of freedom: 16

These results suggest that since $0.05 < 2.41 * 10^{-20}$, the null hypothesis that the variables are independent should be rejected. Additionally, the corresponding chi-squared critical value 26.296 is less than the test statistic 132.19, providing the same conclusion.

Since the null hypothesis was rejected, it was necessary to determine where the dependance most likely occurred. To do this, the contributions were determined from the observed values DataFrame and the expected values output from the chi2_contingency() function.

Secondary Analysis

To complement the results of the analysis performed on the whole set, another round of scrutiny was performed on the dataset. This time, the approach was to take random samples of the dataset, treating the dictionary as a population of Finnish words. These samples were taken using the Pandas sample() function on the separated lists of front and back harmony word groups:

```
front_sample = df[df['VowelHarmony']=='Front']['Word'].sample(n=6000).tolist()
back_sample = df[df['Vowel Harmony'] =='Back']['Word'].sample(n=6000).tolist()
```

The primary motivation for taking samples from within the dataset was to see if smaller sets of words would provide strong enough evidence for initial phoneme-vowel class associations and if multiple samples would point to one or more of the strongest associations present (if any). The sample sizes were chosen to be equal to each other, and about one-tenth of the total number of words. After sampling, the data was analyzed the same as before. This process was repeated for a total of 10 sample runs, and the results were compared. Like the population data analysis, each sample test rejected the null hypothesis that the variables were independent, so the contributions of each character to the test statistic were calculated and used to discover the greatest contributors.

Results

For each set of chi-squared test statistic contributions, the top five contributors were selected and the harmony class it occurred more frequently with was determined. The number of samples each one occurred in and the number of times each was ranked as the greatest contributor were recorded. The results show the following:

Front	Sets	Top Contr.
ʋ	7	3
n	6	1
h	5	1

Back		
d	8	4
vowel	6	1
g	5	0

Additionally, the ranking of the top contributors for the overall population are /d/, /ʊ/, /h/, /n/, /s/. /d/ was more commonly found in back harmony words, while the rest were found more frequently in front harmony words.

Conclusions

Overall, the data point to a few potential associations between initial phonemes and the harmonic class a word belongs to. The strongest associations made were front harmonic words beginning with /ʊ, n, h/, and back harmonic words beginning with /d, g/ or a vowel[13]. These associations should be considered lightly, as more research must be done to conclusively point to reliable patterns that can be used to determine word harmony aside from the vowels contained within. There are several ways that these associations could be more rigorously scrutinized. First, rather than a dictionary of unique words, a corpus containing more naturally generated words such as a transcription from television could be used. This would increase the weight that certain words have on the associations based on their frequency, which would more accurately uncover the cognitive heuristics generated from repeated exposure to certain words more than others. Another way to test for these associations is by running an experiment such as a Wug test. This would present the opportunity for experimenters to more closely control the combinations of phonemes, and then record the time it takes for a speaker to produce the inflected form of the given words. This information would provide evidence for phoneme patterns that allow a speaker to more quickly determine the inflected forms of certain words.

[13] Note that further analysis would be needed to find which vowels may be responsible for this.

References

The Institute for the Languages of Finland. (n.d.). *Languages of finland - institute for the languages of Finland*. Kotimaisten kielten keskus. https://www.kotus.fi/en/on_language/languages_of_finland#Finnish

The Language Bank of Finland. (2019, September 16). *FinnWordNet – the Finnish wordnet*. Kielipankki. https://www.kielipankki.fi/corpora/finnwordnet/

Lyovin, A., & Raimo, I. (n.d.). *Native Phonetic Inventory: finnish*. Speech accent archive: Browse. https://accent.gmu.edu/browse_native.php?function=detail&languageid=20

NumPy documentation#. NumPy documentation - NumPy v1.24 Manual. (2022). https://numpy.org/doc/stable/

Pandas documentation#. pandas documentation - pandas 2.0.1 documentation. (2023). https://pandas.pydata.org/docs/index.html

Scipy documentation#. SciPy documentation - SciPy v1.10.1 Manual. (2023). https://docs.scipy.org/doc/scipy/

Suomi, K., Toivanen, J., & Ylitalo, R. (2008). *Finnish sound structure: Phonetics, phonology, phonotactics and Prosody*. University of Oulu.

The Trustees of Princeton University. (2023). *WNDB(5WN)* | *wordnet*. Princeton University. https://wordnet.princeton.edu/documentation/wndb5wn