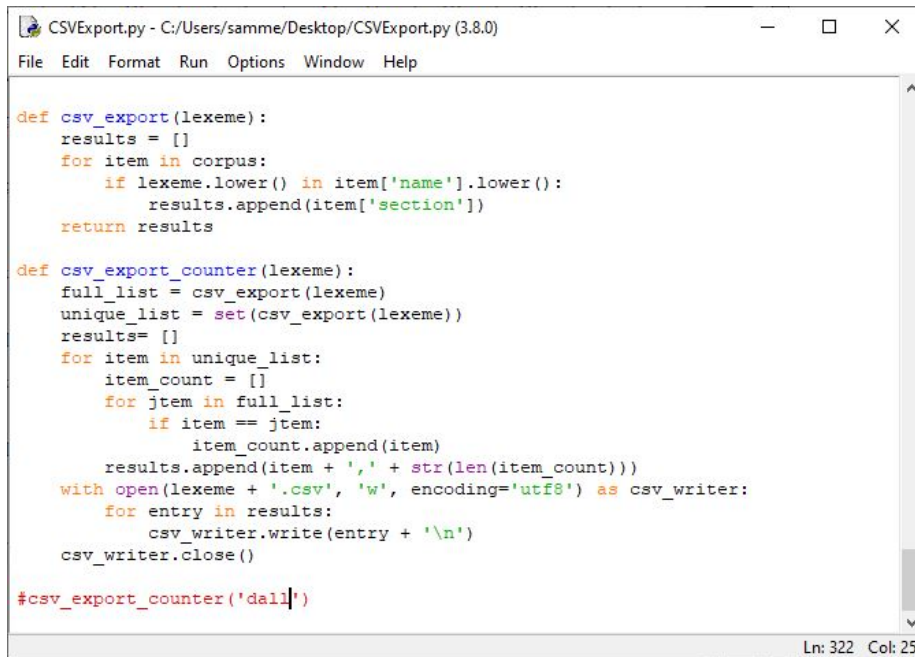


Example

The following is a very simple example for the query of an element (see fig. 1), that is then exported as a CSV-file (see fig. 2), which is in turn used to generate a map (see 3). A query was made using the single lexeme *Dall* ‘valley’ in this exact graphematic rendering.

The image shows a screenshot of a Python IDLE editor window. The title bar reads "CSVExport.py - C:/Users/samme/Desktop/CSVExport.py (3.8.0)". The menu bar includes "File", "Edit", "Format", "Run", "Options", "Window", and "Help". The code is written in Python and is color-coded. It defines two functions: `csv_export` and `csv_export_counter`. `csv_export` takes a `lexeme` as input, iterates through a `corpus`, and appends matching items to a `results` list. `csv_export_counter` calls `csv_export`, creates a `unique_list` from the results, and then iterates through it to count occurrences. Finally, it uses `with open` to write the results to a CSV file. The last line of code is a comment: `#csv_export_counter('dall:')`. The status bar at the bottom right indicates "Ln: 322 Col: 25".

```
def csv_export(lexeme):
    results = []
    for item in corpus:
        if lexeme.lower() in item['name'].lower():
            results.append(item['section'])
    return results

def csv_export_counter(lexeme):
    full_list = csv_export(lexeme)
    unique_list = set(csv_export(lexeme))
    results= []
    for item in unique_list:
        item_count = []
        for jtem in full_list:
            if item == jtem:
                item_count.append(item)
        results.append(item + ',' + str(len(item_count)))
    with open(lexeme + '.csv', 'w', encoding='utf8') as csv_writer:
        for entry in results:
            csv_writer.write(entry + '\n')
    csv_writer.close()

#csv_export_counter('dall:')
```

Figure 1: Code and query for CSV Export in Python's IDLE editor

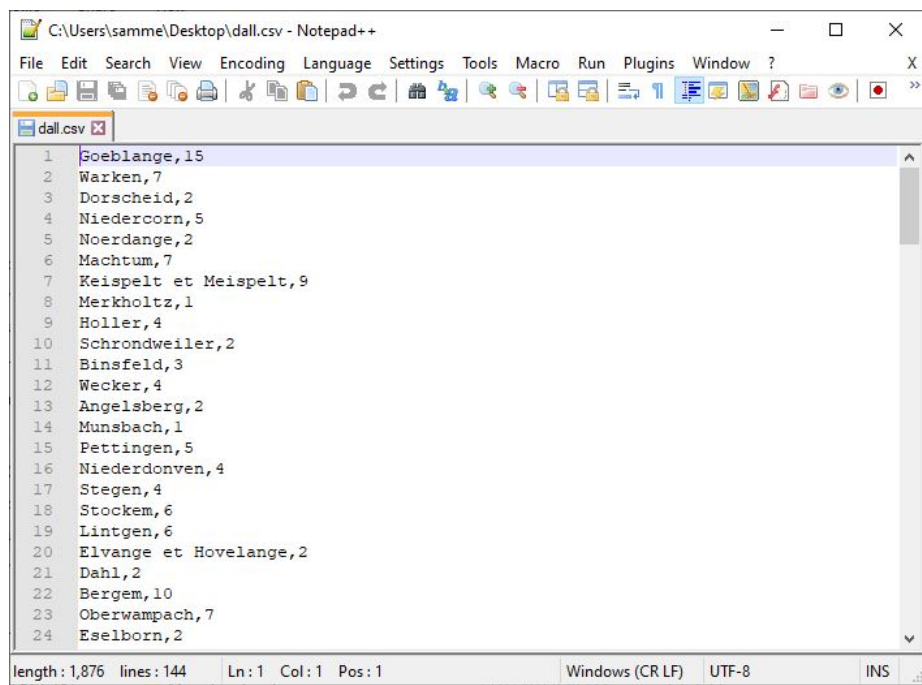


Figure 2: Exported CSV file

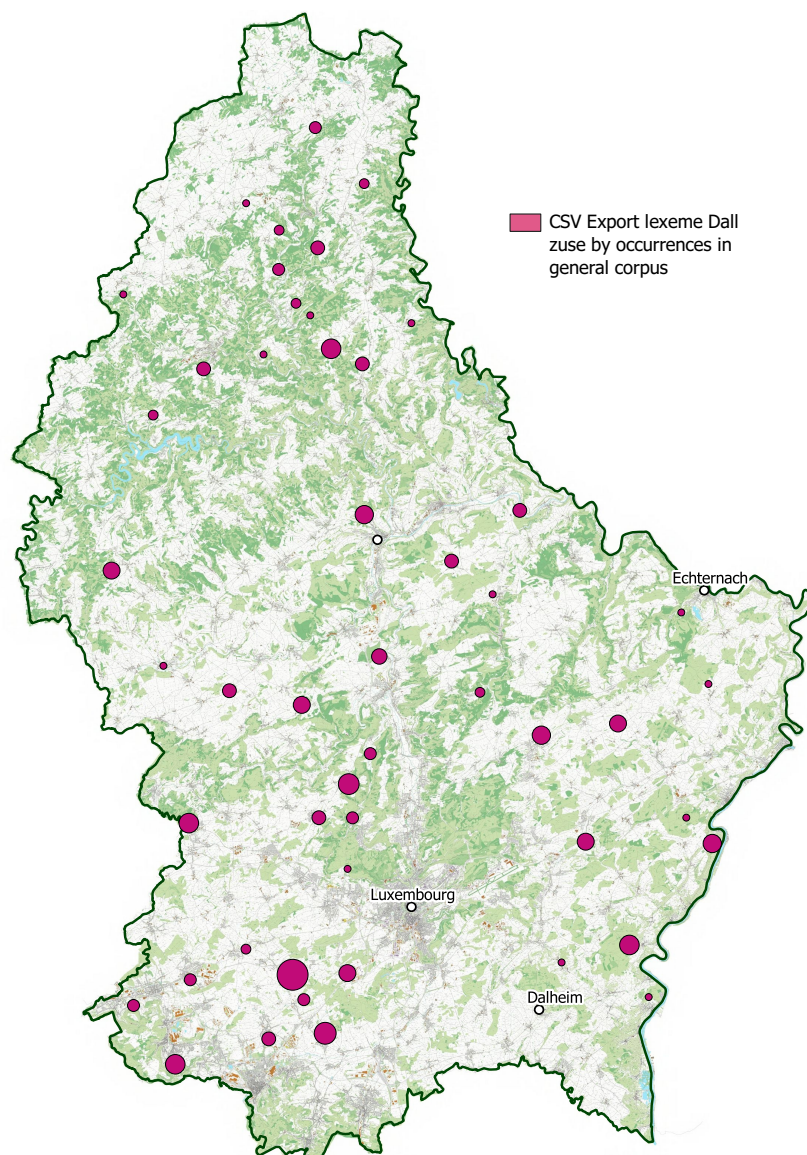


Figure 3: Map generated with CSV Export