# Appendix 2. Used Regex queries (Regular Expressions)

# Claes Pauline. Master of Linguistics (Digital Text Analysis). Academic Year 2021-2022. Master's Thesis.

This files contains all Regular Expressions queries used for retrieving concordances. All regular expressions used were adopted from existing regular expressions previously stored in CosyCat (Arévalo and Petré 2017) by Prof. Petré in the contexts of one of his MA courses and will be added as Appendix 2.

## Progressive [BE Ving]+[GO to V]

For the texts included in the EMMA corpus, the following Regex was provided by Cosycat (Arévalo and Petré 2017):

```
".*'[sſmr]e?|am|ar[te]?|i[sſ]|wa[sſ]t?|wer[et]?|be+n?e?|be+[yi]nge?" [!word="\.|;|,|from|in|of|at|after|like|for|my|a|-"]{0,2} "a"? "-"? "a?goe?ing|_oing|g_ing" [!word="to|\."]{0,5} "to" [!word="\.|;|,|\!|\?|:"]
```

This Regex searches for all instances of be going to + INF that meet the following conditions:

- any form of \textit{to be}, including contracted forms and historical spellings (am, 'm, are, 're, is, 's, art, wert, ...);
- typical forms and spellings of going (going, a going, agoing, a-going);
- allowing for maximally two words between the copula and the form of going;
- allowing for maximally five words between the form of going and to.

For the texts collected from EEBO, a regular expression was used that searches for all progressive instances of the verb go:

```
"a?goe?ing|_oing|g_ing"
```

## Non-progressive [GO to V]

Both the texts collected from EMMA as from EEBO were queried for non-progressive instances of [GO to V] using the following regular expression provided by Cosycat (Arévalo and Petré 2017):

```
"(go+e?[ts]?h?e?|gaes?|wente?s?t?|vvente?s?t?)" "to"
```

AND:

(go+e?[tſ]?h?e?|gaeſ?|wente?ſ?t?|vvente?ſ?t?)’‘to’ (including the historical long 's').