Introduktion til Sprogteknologi

Forår 2020
Lecture 4

Dagens mål

- Mulige løsningsforslag til sammenligning af oratio obliqua @CL-CIT
- Benytte regression test systemet
- Regular Expressions

Løsningsforslag 1/2

For hvert corpus, sammenlign hvor meget @CL-CIT udgør af syntaktiske funktioner, og hvilken retning den mest lægger sig til:

- `cat corpus.txt | wc -w` => 7112
- `cat corpus-analyseret.txt | egrep '@CL-<?CIT>?' | wc -l` => 154
- `cat corpus-analyseret.txt | grep '@CL-<CIT' | wc -l` => 119
- `cat corpus-analyseret.txt | grep '@CL-CIT>' | wc -l` => 35

Så ud af 7112 ord har 154 en @CL-CIT funktion, altså 2.17%. Af de 154 @CL-CIT går 119 (77.3%) til venstre og 35 (22.7%) til højre.

Løsningsforslag

Corporaene giver ca. disse tal:

Corpus	# ord	# @CL-CIT	# @CL-<CIT	# @CL-CIT>
Ataqqinartuaraq	7112	154 (2.17%)	119 (77.3%)	35 (22.7%)
Aviscorpus	3232	74 (2.29%)	36 (48.6%)	38 (51.4%)
UkiutTrettenit	14898	549 (3.69%)	407 (74.0%)	142 (26.0%)
Facebook	2366	20 (0.85%)	14 (70.0%)	6 (30.0%)

Kalaallisut Regression Test 1/2

Findes i mappen ~/langtech/regression/, og indeholder bl.a. filerne:

- analyse.pl Program der kører corpora gennem analysen og checker om den morfologiske analyse har formelle fejl
- compare.pl Program der sammenligner kørslen med det forventede
- input-all.txt Corpus med diverse sætninger
- input-mt.txt Corpus med sætninger der også bliver brugt i Nutserut-projektet

Kalaallisut Regression Test 2/2

Regression test benyttes således:

- cd ~/langtech/regression/
- ./analyse.pl
  Dette tager op til 2 minutter at køre. Vent til den er færdig, og hvis den ikke kommer med nogen ERROR-linier, så gå videre til...
- ./compare.pl
  Starter en web-server og giver en URL man kan åbne fra sin browser for at arbejde videre…
- Når man er færdig i browseren, lukkes web-serveren med CTRL-C i terminalen.

Regular Expression

- https://krijnhoetmer.nl/stuff/regex/cheat-sheet/
- https://www.regular-expressions.info/unicode.html
- https://regex101.com/
- https://regexone.com/
- Eller søg på Google efter: regular expression tutorial

Generelt, Regular Expressions (regex) er case-sensitive - altså, der er forskel på store og små bogstaver.

Regex: Anchors

Begyndelsen og slutningen

- ^ Matcher begyndelsen af inputtet. Under egrep betyder dette starten af linien. I andre kontekster kan det betyde starten af et tag eller ord.
- $ Matcher slutningen af inputtet. Tilsvarende mht. egrep og kontekster, bare slutningen.

Regex: Wildcard

- `.` Et punktum. Matcher 1 vilkårligt tegn.

I nogen sammenhæng er line-break ikke et tegn som `.` matcher, men i de tilfælde kan man enten skrive `[^]` eller tilføje flag `/s`.

Hvad der menes med *tegn* kan også variere - om der menes *byte* eller *Unicode code point* eller *Unicode grapheme cluster* kommer nogen gange an på en prøve.

Regex: Classes & Ranges

- [abc] Tegn i [] betyder at på den plads må der være 1 af de tegn, men vi er ligeglade hvilken en af dem det er. Så [abc] betyder enten a eller b eller c.

- [a-z] Inde i [] betyder - en range, altså en liste af tegn i alfanumerisk rækkefølge. Så [a-z] betyder enten a eller b eller c eller d eller … eller z. Hvilket også betyder at hvis man vil have en literal - til at være et alternativ skal den stå først eller sidst i [], såsom [-ab] eller [ab-]. Man kan også sige [0-9] for at matche alle tal.

- [^abc] Inde i [] betyder ^ hvis den står først at matchet skal inverteres. Altså, [^abc] betyder alle andre tegn end a eller b eller c.

Regex: Repetitions

- ? Det tegn eller gruppe der står lige før ? skal forekomme 0 eller 1 gang. Der er altså ok hvis det slet ikke findes, eller hvis det kun findes én gang. E.g. ab?c matcher ac og abc men ikke abbc.
- + Det der står lige før skal forekomme 1 eller flere gange. Altså mindst én gang. E.g. ab+c matcher abc og abbc og abbbc men ikke ac.
- * Det der står lige før skal forekomme 0 eller flere gang. E.g. ab*c matcher alle af ac, abc, abbc, abbbc, og så videre.

Regex: Repetitions 2/2

- {N} Det tegn eller gruppe der står lige før { skal forekomme nøjagtigt N gange. E.g. ab{2}c matcher abbc men ikke abc eller abbbc.
- {N,} Det der står lige før { skal forekomme mindst N gange.
- {N,M} Det der står lige før { skal forekomme mindst N gange, men højst M gange.

Regex: Groups

- () Grupperer flere tegn så de kan manipuleres som en gruppe. E.g. a(bc)?d matcher ad og abcd men ikke abd eller acd. Tilsvarende med + og * og {}.

Regex: Alternative

- | Matcher enten alt der står før eller alt der står efter |. E.g. abc|def matcher abc eller def, men ikke bcde. Flere | i træk er tilladt, e..g abc|def|ghi.

  Hvis man vil give alternativer midt i et match kan man bruge (). E.g. a(bc|de)f matcher abcf og adef.