Projekt 4: Fínčina

Obsah

Vypracovanie projektu Fínčina **1**
Úvod 1
Metóda 2
Záver 7

Príloha: Jupyter Notebook fínsky preklad slova maslo **8**

Vypracovanie projektu Fínčina

Predpokladaná časová náročnosť: 8hod
Skutočná časová náročnosť: 3hod

Úvod

Cieľom projektu je určiť fínsky preklad slova „maslo“ s využitím paralelných korpusov. Výsledky dosiahneme pomocou aplikácie *KonText*, ktorá je dostupná na stránkach Českého národného korpusu (ČNK), jej API rozhrania, a programovacieho jazyka Python.

Paralelné korpusy sú zbierky rovnakých textov v rôznych jazykoch. Zarovnané korpusy sú paralelné korpusy, ktoré majú navzájom jednoznačne priradené (tie isté, no rôznojazyčné) vety. Takéto korpusy sú vhodným nástrojom na jednoduchú tvorbu slovníkov, nakoľko vieme, že ich texty sú na tú istú tému, sú viac menej zhodné, umožňujú získať významy slov len na základe ich kontextu v zozbieraných textoch. ČNK ponúka k dispozícii širokú škálu zarovnaných korpusov. Medzi nimi je aj sada InterCorp obsahujúca až 61 jazykov, vrátane češtiny a fínčiny. Práve vďaka týmto dvom korpusom dokážeme aspoň približne odhaliť preklad slova maslo.

Prv, než predvedieme naše výsledky, musíme si uvedomiť, že preklad, ku ktorému prídeme, bude skôr len približný a závisí na mnohých predpokladoch. Prvým z predpokladov je, že sú dané dva jazyky úplne preložiteľné. Druhým je, že jednému slovu v jednom jazyku zodpovedá tiež len jediné slovo v druhom jazyku. Ako však vieme, jazyky sú plné perifrastických vyjadrení; nejedná sa iba o viacslovné pomenovania jedného predmetu (napr. anglické *jetski* vs. slovenské *vodný skúter*), ale aj základných gramatických súvislostí (napr. francúzsky slovesný čas *passé recent*). Problematickým momentom pri preklade môže byť aj synonýmia, ktorá môže do značnej miery zmenšovať našu schopnosť detegovať správny ekvivalent. Sem rovnako spadajú prenesené významy, idiomy a metafory, ktoré sú kultúrne závislé a tak pri ich preklade len zriedka dochádza ku prekladu spôsobom „náhrada slova za slovo“. Naprieč jazykmi taktiež existuje rôzna sémantická segmentácia sveta, čo je v lingvistike pomerne dosť známe vďaka hypotéze postulovanej dvojicou Sapir-Whorf (podobné myšlienky demonštruje aj Jakobson so svojím príkladom rozdielov vo význame slov *syr*, *tvorog* (rus.) a *cheese* (ang.)). Ďalším problémom môže byť doménovo špecifický význam, slová v rôznych komunitách môžu mať aj v rámci jediného jazyka rôzne významy. Napriek všetkým spomenutým komplikáciám je ale zarovnaný korpus zvyčajne možné vytvoriť (čo aj demonštruje počet korpusov v sade InterCorp), pri jeho tvorení je len treba všímať si týchto možných komplikácií a uvedomovať si, kde sa vyskytujú častejšie (napr. poézia), a kde sú zriedkavejšie (napr. opisy). Na vytvorenie slovníku z paralelného korpusu teda potrebujeme, aby existoval preklad medzi jazykmi, aby jednému slovu zodpovedalo jedno slovo v druhom jazyku, aby bola zbierka textov dostatočne veľká a diverzifikovaná. Tiež aby boli jednoznačne k sebe priradené vety, ktoré sú svojím prekladom, a aby korpus obsahoval dostatočné množstvo viet s výskytom nášho záujmového slova.

V našom prípade budeme pracovať s dvojicou jazykov čeština a fínčina. Nakoľko fínčina nie je úplne neznámy jazyk, dopredu vieme, čo môžeme od nej očakávať. Má vysokú mieru

flexie, jej slová sa vyskytujú v rôznych tvaroch. Taktiež je aglutinačná – jej pomer afixov k ich funkciám je blízky 1. Každý afix má práve jednu funkciu, čo znamená, že fínske slová sú pomerne dlhé, nakoľko viaceré gramatické vzťahy vyjadruje pomocou viacerých afixov. Naopak čeština napr. pri substantívach vyjadruje gramatickú kategóriu rodu, čísla a pádu jediným afixom (žen-y = singulár, genitív, rod ženský). Fínčina využíva latinku s diakritickou úpravou. Všetky tieto informácie využijeme aj pri našom spracovaní.

Metóda

Náš postup pozostáva z 3 krokov: určenie predpokladaných kandidátov prekladu, analýza frekvencie možných slov a morfémov, reverzné testovanie prekladu.

Na stránkach ČNK pomocou nástroja *KonText*, po zvolení korpusu InterCorp v16 - Czech a paralelného korpusu InterCorp v16 - Finnish, získame dáta vo formáte xlsx po zadaní pokročilého CQL výrazu: [lemma="máslo"& tag="N.{3}1.*"]. Týmto výrazom využívame fakt, že český korpus je lemmatizovaný, a tak môžeme pomocou základného tvaru slova zvoliť všetky jeho rôzne formy. Úplnej volnosti však zamedzujeme tým, že sa sústredíme na výskyty klasifikované ako substantíva v nominatíve. Dôvodom pre to je, že existuje predpoklad, že slová v nominatíve (v páde podmetu) zastupujú podmety (pokiaľ nie je jazyk ergatívny), a tak sú väčsinou v tom najjednoduchšom tvare. Predpokladáme, že preklad slova v nominatíve, čize v najčastejšej pozícii agensu a podmetu, bude vyžadovať rovnakú pozíciu aj vo fínčine. Nakoľko máme tiež predpoklad, že nominatív je čo do stavby najjednoduchší pád, vo výsledku dostaneme najjednoduchšie možné tvary slov. Napriek takémuto zúženiu nášho poľa hypotéz, dostávame nemalé množstvo dát, s ktorými budeme pracovať (944 výskytov).

V ďalšom kroku, ktorého realizácia je uvedená v podnadpise Morfémy v prílohe, určíme najčastejšie zoskupenia písmen v slovách. Jedná sa o značné zjednodušenie, no takýmto spôsobom sa snažíme očistiť slová od afixov, a získať najbežnejšie morfémy, medzi ktorými sa môže vyskytovať aj celé fínske slovo pre maslo. Okrem celkovej frekvencie morfémov v texte skúmame aj počet jednotlivých vstupov, viet, v ktorých sa vyskytujú. Predpokladáme totiž, že počet riadkov/viet s českým slovom maslo bude približne rovnaký ako počet riadkov/viet s tým istým slovom po fínsky, očakávame, že sa bude vyskytovať vo väčšine záznamov. Prvých pár najfrekventovanejších zoskupení písmen vidíme na tabuľke 1. Za morfémy považujeme iba tie sekvencie písmen, ktoré sú dlhšie ako 1 písmeno, ale kratšie ako 10. Keďže fínčina používa latinku, budeme za písmená považovať klasické latinské písmená s pridanou diakritikou. Sadu prázdnych znakov uvažujeme totožnú s tou českou.

Obdobný postup aplikujeme aj pri extrakcii slov, textových reťazcov ohraničených medzerami. V tabuľke 2 uvádzame 30 najfrekventovanejších slov zoradených od najčastejšieho po najmenej časté. Na obrázku 1 vidíme 4 rôzne histogramy, v ktorých je vyobrazená frekvencia jednotlivých morfémov a slov zoradených buď podľa frekvencie, alebo podľa počtu riadkov, v ktorých sa nachádzajú. Vidíme, že prvé priečky všetkých štyroch grafov majú viaceré spoločné rysy. Obsahujú podreťazce a reťazce ako: ta, oi, in, voi, vo, voita. Frekvencie výskytu morfémov sú, samozrejme, v tomto prípade závislé od frekvencie výskytu slov, nakoľko tvoria ich základ. No už teraz môžeme povedať, že sa jedná o najčastejšie

reťazce, ktoré sa zároveň vyskytujú v najväčšom počte rôznych textových vzoriek. Pri pohľade na obe tabuľky vidíme aj to, že určité sady písmen (morfémov) sa vyskytujú aj v extrahovaných slovách. Tento postreh nás vedie k zavedeniu ďalšej, syntetickej entity – morfoslovo. V jupyter notebooku dostupnom v prílohe definujeme množinu morfoslov ako prienik množiny slov a morfémov. Predpokladáme, že takéto slová sú význačné, lebo hrajú úlohu jednak pri stavaní ďalších slov, a zároveň sú aj samé slovami. Presne niečo také by sme, myslíme, očakávali aj od nominatívovej formy slova maslo; je samostatná, no zároveň slúži aspoň sčasti k tvoreniu ostatných pádov. K morfoslovám patria nasledujúce tvary: artiklan, asetuksen, ja, komission, on, se, voi, voita. Práve tie sú našimi najväčšími kandidátmi na preklad slova maslo. Práve na tie sa zameriame v poslednom kroku nášho postupu, pri krížovej kontrole.

Kontrolu vykonávame automaticky, pomocou requestov na ČNK API. Návod, ako s nimi pracovať, nájdeme na nasledujúcich dvoch odkazoch: https://wiki.korpus.cz/doku.php/manualy:api, https://github.com/czcorpus/kontext/wiki/HTTP-API. V skratke môžeme celý proces opísať tak, že sa automaticky prihlásime do aplikácie *KonText*, zvolíme si ako hlavný korpus InterCorp pre fínčinu a ako vedľajší zarovnaný už spomínaný InterCrop pre češtinu. Následne máme možnosť vytvoriť si vlastné CQL query tak, aby vyhľadávalo priamo lemma tvary (fínsky korpus je lematizovaný) alebo slová. V prvom prípade dostávame menej výsledkov, a riskujeme stratu prekladu, v druhom prípade zas existuje riziko, že výsledných dát bude príliš veľa a nebude jednoduché sa rozhodnúť, ktorý variant prekladu je správny. My si volíme prvý prístup. Automaticky pomocou requestov potom dosadzujeme do query jednotlivé morfoslová, a získavame vety v češtine, na ktoré aplikujeme rovnaké postupy ako predtým na fínčinu. Po extrahovaní dát získavame tabuľku 3, v ktorej sú riadkové a klasické frekvencie morfoslov vo vetách, ktoré sme získali v úvode práce ako paralelné dáta ku vetám so slovom *maslo*. Taktiež tam sú relatívne zastúpenia slova maslo vo vetách, ktoré sme získali po všetkých dopytoch s morfoslovami, a relatívne zastúpenie riadkov so slovom maslo. Posledný stĺpec tabuľky vyjadruje silu prekladu. Ak prepočítame relatívne frekvencie tak, že ich vydelíme súčtom všetkých relatívnych frekvencií, dostávame silu prekladu. Sila prekladu v tabuľke je aritmetický priemer takto získaných síl testu z klasických a riadkových relatívnych frekvencií.

Morfém	Počet riadkov	Frekvencia	Morfém	Počet riadkov	Frekvencia
ta	675	2156	te	401	1259
oi	789	1769	tu	354	1216
in	643	1631	vo	688	1095
en	405	1622	an	421	1064
is	437	1419	voi	685	1062
it	637	1414	aa	558	988
tt	432	1307	va	384	901
st	424	1300	tä	399	872
et	370	1267	ai	379	847

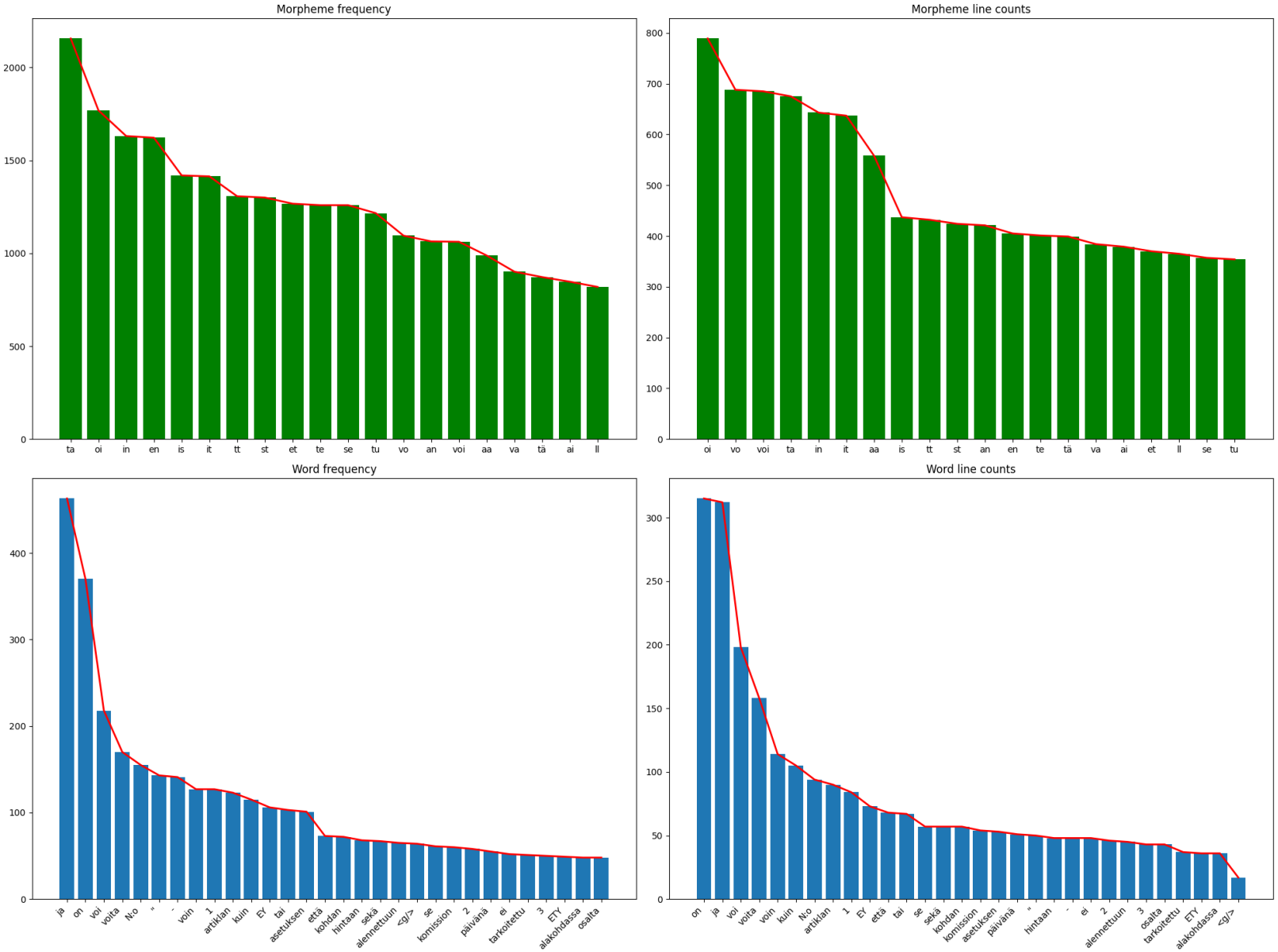
Tabuľka 1: Prvých pár najfrekventovanejších možných morfémov.

Slovo	Počet riadkov	Frekvencia	Slovo	Počet riadkov	Frekvencia
ja	312	463	kohdan	57	72
on	315	370	hintaan	48	68
voi	198	218	sekä	57	67
voita	158	170	alennettuun	45	65
N:o	94	155	<g/>	17	64
"	50	143	se	57	61
-	48	141	komission	54	60
voin	114	127	2	46	58
1	84	127	päivänä	51	55
artiklan	90	123	ei	48	52
kuin	105	115	tarkoitettu	37	51
EY	73	106	3	43	50
tai	67	103	ETY	36	49
asetuksen	53	101	alakohdassa	36	48
että	68	73	osalta	43	48

Tabuľka 2: Prvých pár najfrekventovanejších možných slov.

Morfoslovo	Počet riadkov	Frekvencia	Relatívny počet riadkov slova maslo	Relatívna frekvencia slova maslo	Miera dôvery v preklad
se	357	1259	0,00	0,00	0,00
on	315	370	0,00	0,00	0,00
artiklan	90	123	0,00	0,00	0,00
asetuksen	53	101	0,00	0,00	0,00
voi	685	1062	0,04	0,07	0,28
ja	312	463	0,00	0,00	0,01
komission	54	60	0,00	0,00	0,00
voita	158	170	0,13	0,16	0,71

Tabuľka 3: Morfoslová s ich frekvenciou vo vetách priradených slovu maslo a frekvencia slova maslo vo vetách priradených morfoslovám.



Obr. 1: Zoradené frekvencie najčastejších slov a morfémov.

Záver

Podľa našich výsledkov sa maslo po fínsky povie *voi* alebo *voita*. Zadaná úloha je, pokiaľ sú dodržané predpoklady, riešiteľná úplne.

Treba však podotknúť, že počet všetkých predpokladov, s ktorými sme v tejto práci narábali, nie je malý. Naviac, naše hypotézy sú veľmi umelé a len ťažko odrážajú skutočnú prirodzenosť jazyka. Získať správny preklad je zložitý proces, ktorý nie je obecne takto jednoducho riešiteľný, napriek tomu, že v prípade niektorých slov funguje. Naša metóda by napríklad nedokázala získať preklad slova maslo v prípade, že by ku nemu bolo vo fínskych textoch odkazované prevažne zámenami alebo ak by bolo maslo vo fínčine viacslovné pomenovanie. Podobne by nás mohol zmiasť aj kultúrne zaťažený kontext (v extrémnom hypotetickom príklade preložíme slovo *pes* do angličtiny ako *Snoopy*), pragmatika alebo väčšia miera viazanosti niektorých slov (preložíme anglické *butter* ako slovenské *chlieb* pre častú frekvenciu spojenia *bread and butter*). Využili sme aj znalosti o fínčine a jej jednoducho dešifrovateľné písmo, čo v prípade neznámych jazykov nemusí byť možné; našu metódu treba preto upraviť smerom k väčšej obecnosti a aplikovateľnosti nezávisle od jazyka. Tieto a iné zjednodušenia je treba brať do úvahy v prípade, že sa rozhodneme túto heuristickú metódu aplikovať na nové, odlišné dáta.

Príloha: Jupyter Notebook fínsky preklad slova maslo

Projekt LIA: Preklad do fínčiny

Fínčina

- veľký počet pádov
- na pomedzí flektívnych a aglutinačných jazykov
- používa latinku s diakritikou

Metóda:

1. Určenie predpokladaných kandidátov na preklad:
2. Analýza frekvencií možných slov a morfémov
3. Reverzné testovanie prekladu
4. Výsledky

Nástroje:

- ČNK (CQL query: [lemma="máslo" & tag="N.{3}1.*"])
- ČNK API (https://github.com/czcorpus/kontext/wiki/HTTP-API)
- Python

Stanovenie konštánt

[101]:
```
TOP_K = 30
WORD_LIMIT = 20
WHITE_SPACE_SYMBOLS = " .,!?;()"
```

Morfémy

Určenie prekladu jedného slova vo fínčine je zložité pre veľké množstvo morfémov; budeme sa teda snažiť nájsť priesečník medzi morfémami a slovami.

Predpoklady

1. morfém - krátky ngram zo znakov
2. slovo - reťazec znakov vymedzený medzerou alebo white space symbolom

Nahratie dát

Dáta sú získané zo zarovnaných korpusov dostupných na ČNK: InterCorp v16 - Finnish, InterCorp v16 Czech

Stiahnuté dáta vo formáte xlsx nahráme pomocou pandas

[102]:
```
import pandas as pd
corpus_data = pd.read_excel("korpus.xlsx", header=None)
corpus_data[0] = corpus_data.iloc[:, 1:5].astype(str).apply(' '.join, axis=1)
corpus_data = corpus_data.drop(columns=[1,2,3,4,5])
corpus_data.columns = ["cz", "fn"]
corpus_data.head()
```

[102]:
```
                                                   cz  \
0  Co je Arašídový strom a Arašídové máslo ? _SUB...
1                        Arašídové máslo . _SUBTITLES
2                  Řezník byl jako máslo . _SUBTITLES
3  d ) popřípadě chladírenský sklad , v němž je m...
4  A když se tě zeptají , jestli je to máslo prav...

                                                   fn
0  Sitten riennän koulunäytelmiin- runoiltoihin j...
1                                  Maapähkinävoita .
2                  Teurastaja oli kuin sulaa voita .
3  d ) tarvittaessa kylmävarasto , jossa voita sä...
4  Jos he kysyvät , onko se oikeaa voita , mitä v...
```

Extrakcia morfémov

Extrahujeme všetky potenciálne morfémy do prednastavenej dĺžky

Načítanie všetkých textov

[103]:
```
finnish_text = " ".join(corpus_data["fn"])
czech_text = " ".join(corpus_data["cz"])
```

Extrakcia morfémov

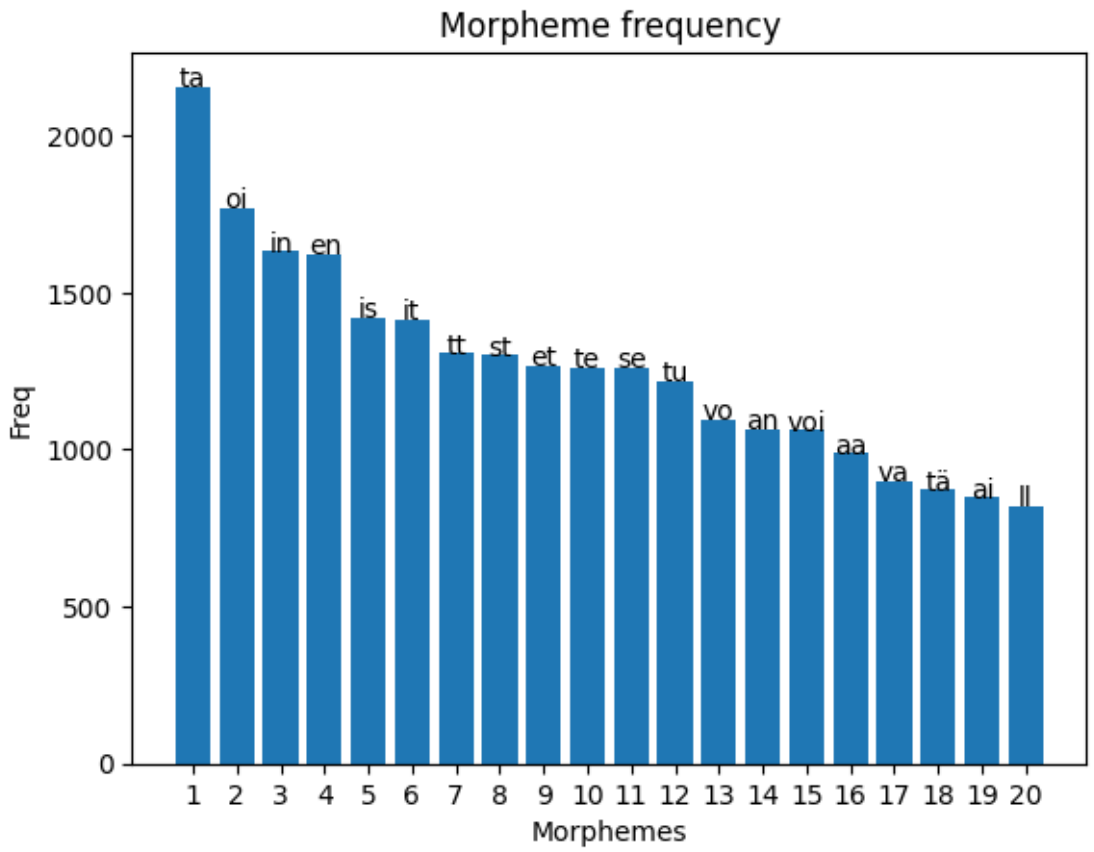
[104]:
```
from collections import Counter
from functools import reduce

def find_white_space_symbols(string):
  found = []
  for white_space in WHITE_SPACE_SYMBOLS:
    found.append(string.find(white_space) == -1)
  return not reduce(lambda x, y: x*y, found)

def find_ngrams(text, n):
    ngrams = [text[i:i+n] for i in range(len(text)-n+1) if not␣
 ↪find_white_space_symbols(text[i:i+n])]
    return ngrams

def most_frequent_ngrams(text, n, top_k=5):
    ngrams = find_ngrams(text, n)
    ngram_counts = Counter(ngrams)
    return ngram_counts.most_common(top_k)

morphemes = dict()
shortest_morpheme = 2
longest_morpheme = 10
for morpheme_length in range(shortest_morpheme, longest_morpheme):
  for word, freq in most_frequent_ngrams(finnish_text, morpheme_length, TOP_K):
```

```
    word = word.lower().strip()
    if word in morphemes:
      morphemes[word] += freq
    else:
      morphemes[word] = freq
possible_morpheme_translations = sorted(morphemes.items(), key=lambda item:␣
 ↪item[1], reverse=True)
limited_possible_morpheme_translations = possible_morpheme_translations[:
 ↪WORD_LIMIT]
```

Ukážka extrahovaných dát

[105]:
```
morphs = [data_entry[0] for data_entry in possible_morpheme_translations]
freqs = [data_entry[1] for data_entry in possible_morpheme_translations]
```

[106]:
```
import matplotlib.pyplot as plt
plt.bar([str(index + 1) for index in range(WORD_LIMIT)], freqs[:WORD_LIMIT])
plt.xlabel('Morphemes')
plt.ylabel('Freq')
plt.title('Morpheme frequency')
for index, (word, freq) in enumerate(possible_morpheme_translations[:
 ↪WORD_LIMIT]):
    plt.text(index, freq + .2, word, ha='center')
plt.show()
```



[107]:
```
def get_num_hits(corpus_data, string, corpus_data_key="fn"):
  return corpus_data[corpus_data_key].apply(lambda sent: sent.find(string) >= 0).
 ↪sum()

morphemes_df = dict()
for morpheme, freq in limited_possible_morpheme_translations:
  morphemes_df[morpheme] = get_num_hits(corpus_data, morpheme), freq

morphemes_df = pd.DataFrame.from_dict(morphemes_df).transpose()
morphemes_df.columns = ["num_of_lines", "freq"]
morphemes_df.sort_values(by="freq", ascending=False).head(TOP_K)
```

[107]:

	num_of_lines	freq
ta	675	2156
oi	789	1769
in	643	1631
en	405	1622
is	437	1419
it	637	1414
tt	432	1307

```
st            424  1300
et            370  1267
se            357  1259
te            401  1259
tu            354  1216
vo            688  1095
an            421  1064
voi           685  1062
aa            558   988
va            384   901
tä            399   872
ai            379   847
ll            365   819
```

Slová

Extrakcia slov

[108]:
```
def get_words_freqs(text):
    words = [word for word in text.split() if word not in WHITE_SPACE_SYMBOLS]
    words_counter = Counter(words)
    return words_counter

word_frequencies = get_words_freqs(finnish_text)
possible_word_translations = word_frequencies.most_common(TOP_K)
words_df = dict()
for word, freq in possible_word_translations:
  words_df[word] = get_num_hits(corpus_data, " " + word + " "), freq

words_df = pd.DataFrame.from_dict(words_df).transpose()
words_df.columns = ["num_of_lines", "freq"]
words_df.sort_values(by="freq", ascending=False).head(TOP_K)
```

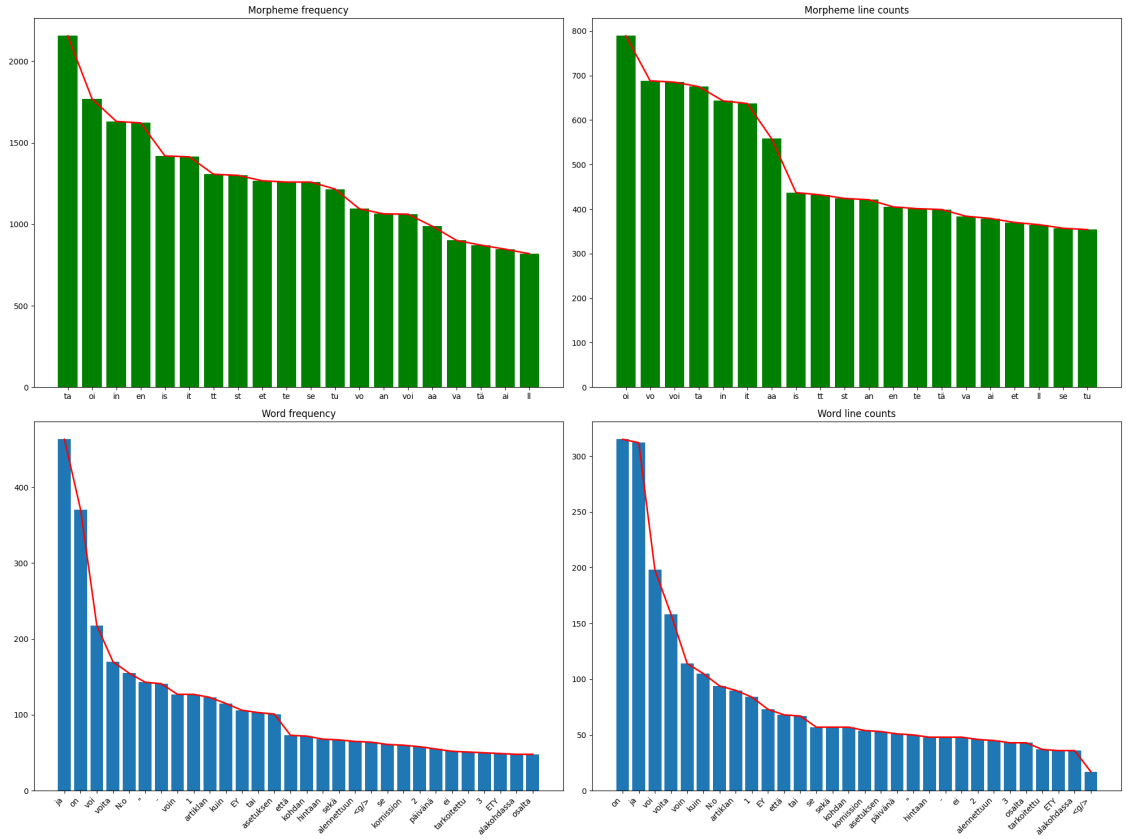
[108]:

	num_of_lines	freq
ja	312	463
on	315	370
voi	198	218
voita	158	170
N:o	94	155
"	50	143
-	48	141
voin	114	127
1	84	127
artiklan	90	123
kuin	105	115
EY	73	106
tai	67	103

```
asetuksen               53   101
että                    68    73
kohdan                  57    72
hintaan                 48    68
sekä                    57    67
alennettuun             45    65
<g/>                    17    64
se                      57    61
komission               54    60
2                       46    58
päivänä                 51    55
ei                      48    52
tarkoitettu             37    51
3                       43    50
ETY                     36    49
alakohdassa             36    48
osalta                  43    48
```

[109]:
```
def create_histos(axes, morphemes_df, words_df):
  x, y = list(morphemes_df["freq"].keys()), list(morphemes_df["freq"].values)
  axes[0][0].bar(x, y, color="green")
  axes[0][0].set_title('Morpheme frequency')
  axes[0][0].plot(x, y, color='red', linewidth=2)
  x = list(morphemes_df.sort_values(by="num_of_lines",␣
↪ascending=False)["num_of_lines"].keys())
  y = list(morphemes_df.sort_values(by="num_of_lines",␣
↪ascending=False)["num_of_lines"].values)
  axes[0][1].bar(x, y, color="green")
  axes[0][1].set_title('Morpheme line counts')
  axes[0][1].plot(x, y, color='red', linewidth=2)
  x, y = list(words_df["freq"].keys()), list(words_df["freq"].values)
  axes[1][0].bar(x, y)
  axes[1][0].set_xticks(x)
  axes[1][0].set_xticklabels(x, rotation=45, ha='right')
  axes[1][0].set_title('Word frequency')
  axes[1][0].plot(x, y, color='red', linewidth=2)
  x = list(words_df.sort_values(by="num_of_lines",␣
↪ascending=False)["num_of_lines"].keys())
  y = list(words_df.sort_values(by="num_of_lines",␣
↪ascending=False)["num_of_lines"].values)
  axes[1][1].bar(x, y)
  axes[1][1].set_xticks(x)
  axes[1][1].set_xticklabels(x, rotation=45, ha='right')
  axes[1][1].set_title('Word line counts')
  axes[1][1].plot(x, y, color='red', linewidth=2)
  return
```

```
fig, axes = plt.subplots(nrows=2, ncols=2, figsize=(20, 15))
create_histos(axes, morphemes_df, words_df)
plt.tight_layout()
plt.show()
```



Morfoslová

Morfoslovo - prienik množín morfémov a slov

[110]:
```
morph_words = set.intersection(set([morph[0] for morph in␣
↪possible_morpheme_translations]),
                               set([word[0] for word in␣
↪possible_word_translations]))
morph_words
```

[110]:
```
{'artiklan', 'asetuksen', 'ja', 'komission', 'on', 'se', 'voi', 'voita'}
```

[111]:
```
butter_line_freq = corpus_data.shape[0]
butter_freq = get_words_freqs(czech_text)["máslo"]
```

[112]:
```
morph_words_df = dict()
for morph_word in morph_words:
  try:
    info = morphemes_df.loc[morph_word]
  except:
    info =  words_df.loc[morph_word]
  morph_words_df[morph_word] = {"num_of_lines": info[0],
                                "num_of_lines_relative": info[0]/
 ↪butter_line_freq,
                                "freq": info[1],
                                "freq_relative": info[1] / butter_freq}

morph_words_df
```

[112]:
```
{'se': {'num_of_lines': 357,
  'num_of_lines_relative': 0.3781779661016949,
  'freq': 1259,
  'freq_relative': 1.2223300970873787},
 'on': {'num_of_lines': 315,
  'num_of_lines_relative': 0.3336864406779661,
  'freq': 370,
  'freq_relative': 0.3592233009708738},
 'artiklan': {'num_of_lines': 90,
  'num_of_lines_relative': 0.09533898305084745,
  'freq': 123,
  'freq_relative': 0.11941747572815534},
 'asetuksen': {'num_of_lines': 53,
  'num_of_lines_relative': 0.05614406779661017,
  'freq': 101,
  'freq_relative': 0.09805825242718447},
 'voi': {'num_of_lines': 685,
  'num_of_lines_relative': 0.725635593220339,
  'freq': 1062,
  'freq_relative': 1.0310679611650486},
 'ja': {'num_of_lines': 312,
  'num_of_lines_relative': 0.3305084745762712,
  'freq': 463,
  'freq_relative': 0.4495145631067961},
 'komission': {'num_of_lines': 54,
  'num_of_lines_relative': 0.057203389830508475,
  'freq': 60,
  'freq_relative': 0.05825242718446602},
 'voita': {'num_of_lines': 158,
  'num_of_lines_relative': 0.1673728813559322,
  'freq': 170,
  'freq_relative': 0.1650485436893204}}
```

Kontrola výsledkov

ČNK API request pred vykonaním tohto kroku je potrebné si zdarma aktivovať vlastný prístupový kľúč k API podľa návodu: https://wiki.korpus.cz/doku.php/manualy:api

[113]:
```
personal_access_token = "YOUR_VERY_SPECIAL_API_KEY"
```

Nastavenie korpusov

[114]:
```
czech_corpus_name= "intercorp_v16_cs"
finnish_corpus_name="intercorp_v16_fi"
original_query = "[lemma=\"máslo\" & tag=\"N.{3}1.*\"]"
MAX_NUM_RESULTS = 4*corpus_data.shape[0]
FIRST_N_PAGES = 1
```

Funkcia na posielanie API requestov

[115]:
```
import pickle, requests
cookies_file = 'cookies.pickle'

def load_cookies(s):
  try:
      with open(cookies_file, 'rb') as f:
          s.cookies.update(pickle.load(f))
  except FileNotFoundError:
      pass
  return s

def generate_request_body(corpusA, corpusB, query, fromp=0):
  request_body = {
        "type": "concQueryArgs",
        "maincorp": corpusA,
        "usesubcorp": None,
        "viewmode": "align",
        "pagesize": 100,
        "attrs": "word",
        "attr_vmode": "visible-kwic",
        "base_viewattr": "word",
        "ctxattrs": [],
        "structs": ["text","p","g"],
        "refs": [],
        "fromp": fromp,
        "shuffle": 1, #premiesaj riadky
        "queries": [
            {
                "qtype": "advanced",
                "corpname": corpusA,
                "query": query,
                "pcq_pos_neg": "pos",
                "include_empty": False,
```

```
                "default_attr": "word"
            },{
                "qtype": "simple",
                "corpname": corpusB,
                "query": "",
                "pcq_pos_neg": "pos",
                "include_empty": False,
                "default_attr": "word"
            }
        ],
        "text_types": {},
        "context":
        {
            "fc_lemword_wsize": [-5, 5],
            "fc_lemword": "",
            "fc_lemword_type": "all",
            "fc_pos_wsize": [-5, 5],
            "fc_pos": [],
            "fc_pos_type": "all"
        },
        "async": False
    }
  return request_body


def get_corpus_data(corpusA, corpusB, query, fromp=0):
  with requests.Session() as s:
    s = load_cookies(s)
    r = s.post('https://korpus.cz/login', data={'personal_access_token':␣
 ↪personal_access_token})
    request_body = generate_request_body(corpusA, corpusB, query, fromp)
    r = s.post('https://korpus.cz/kontext-api/v0.17/query_submit',␣
 ↪params={'format': 'json'}, json=request_body)
    response_json = r.json()
    conc_persistence_op_id = response_json['conc_persistence_op_id']
    r = s.get('https://korpus.cz/kontext-api/v0.17/view',
              params={'format': 'json', 'q': '~' +conc_persistence_op_id,
                      'pagesize': MAX_NUM_RESULTS, 'viewmode':'align'})

    with open(cookies_file, 'wb') as f: #save cookie
      pickle.dump(s.cookies, f)
    return r
```

[116]:
```
maslo_responses = [get_corpus_data(czech_corpus_name,finnish_corpus_name,␣
 ↪original_query, i) for i in range(FIRST_N_PAGES)]
def get_align_sents(rs):
  aligned_sents = []
```

```
  for r in rs:
    try:
      parsed_response = r.json()
      for line in parsed_response["Lines"]:
        aligned_sents.append(line["Align"][0]["Kwic"][0]["str"])
    except:
      pass
  return aligned_sents

maslo_aligned = get_align_sents(maslo_responses)
```

[117]:
```
print("Počet fínskych paralelných viet: ", len(maslo_aligned))
print("Ukážka:\n" + "\n".join(maslo_aligned[:4]))
```

```
Počet fínskych paralelných viet:  944
Ukážka:
Sitten riennän koulunäytelmiin- runoiltoihin ja esitelmiin .
Maapähkinävoita .
Teurastaja oli kuin sulaa voita .
d ) tarvittaessa kylmävarasto , jossa voita säilytetään , ja mahdollisesti
korvaava varasto ;
```

Krížova kontrola prekladu pomocou API request

[118]:
```
QUERY_TYPE = "lemma"
translations = dict()
morph_words = list(morph_words)
```

[119]:
```
for morph_word in morph_words:
  translation_query = f"[{QUERY_TYPE}=\"{morph_word}\"]"
  print(f"Querying corpus with query: {translation_query}")
  translations_corpus = [get_corpus_data(finnish_corpus_name, czech_corpus_name,␣
 ↪translation_query) for i in range(FIRST_N_PAGES)]
  translations_aligned = list(set(get_align_sents(translations_corpus)))
  print(f"{len(translations_aligned)} results retrieved")
  translations[morph_word] = translations_aligned
```

```
Querying corpus with query: [lemma="se"]
3754 results retrieved
Querying corpus with query: [lemma="on"]
2682 results retrieved
Querying corpus with query: [lemma="artiklan"]
0 results retrieved
Querying corpus with query: [lemma="asetuksen"]
0 results retrieved
Querying corpus with query: [lemma="voi"]
2783 results retrieved
Querying corpus with query: [lemma="ja"]
3766 results retrieved
```

```
Querying corpus with query: [lemma="komission"]
5 results retrieved
Querying corpus with query: [lemma="voita"]
391 results retrieved
```

Kontrola frekvencie výskytu slova maslo pre jednotlivých kandidátov prekladu

[120]:
```
WORD_SUFFIX_LEN = 1

def get_target_word_freq(translations, word):
  freqs = dict()
  word_stem = word[:len(word) - WORD_SUFFIX_LEN]
  print("Základný tvar slova: "  + word)
  print("Umelý kmeň slova: " + word_stem)
  for morphword, sents in translations.items():
    lc = sum([word_stem in sent  for sent in sents])
    sents_str = " ".join(sents).strip().lower()
    wf = get_words_freqs(sents_str)[word]
    mf = Counter(find_ngrams(sents_str, len(word_stem)))[word_stem]
    rlc = lc/len(sents) if len(sents) else 0
    rmf = mf/len(sents) if len(sents) else 0
    freqs[morphword] = (wf, mf, lc, rmf, rlc)
    print("Počet výskytov v paralelných vetách pre morfoslovo "+ morphword)
    print("Kmeň ako morfém:",mf,"Slovo ako slovo:", wf, sep="\t")
    print(f"Relatívne zastúpenie slova {word} ", rmf, sep="\t")
    print(f"Počet riadkov so slovom/kmeňom: {word}/{word_stem}", lc, sep="\t")
    print(f"Relatívny počet riadkov: ", rlc, sep="\t")
    print()
  return freqs
translation_frequencies = get_target_word_freq(translations, "máslo")
```

```
Základný tvar slova: máslo
Umelý kmeň slova: másl
Počet výskytov v paralelných vetách pre morfoslovo se
Kmeň ako morfém:        1       Slovo ako slovo:        0
Relatívne zastúpenie slova máslo        0.0002663825253063399
Počet riadkov so slovom/kmeňom: máslo/másl      1
Relatívny počet riadkov:        0.0002663825253063399

Počet výskytov v paralelných vetách pre morfoslovo on
Kmeň ako morfém:        0       Slovo ako slovo:        0
Relatívne zastúpenie slova máslo        0.0
Počet riadkov so slovom/kmeňom: máslo/másl      0
Relatívny počet riadkov:        0.0

Počet výskytov v paralelných vetách pre morfoslovo artiklan
Kmeň ako morfém:        0       Slovo ako slovo:        0
Relatívne zastúpenie slova máslo        0
Počet riadkov so slovom/kmeňom: máslo/másl      0
```

```
Relatívny počet riadkov:        0

Počet výskytov v paralelných vetách pre morfoslovo asetuksen
Kmeň ako morfém:        0       Slovo ako slovo:        0
Relatívne zastúpenie slova máslo        0
Počet riadkov so slovom/kmeňom: máslo/másl      0
Relatívny počet riadkov:        0

Počet výskytov v paralelných vetách pre morfoslovo voi
Kmeň ako morfém:        202     Slovo ako slovo:        95
Relatívne zastúpenie slova máslo        0.07258354293927416
Počet riadkov so slovom/kmeňom: máslo/másl      120
Relatívny počet riadkov:        0.04311893639956881

Počet výskytov v paralelných vetách pre morfoslovo ja
Kmeň ako morfém:        8       Slovo ako slovo:        1
Relatívne zastúpenie slova máslo        0.002124269782262347
Počet riadkov so slovom/kmeňom: máslo/másl      2
Relatívny počet riadkov:        0.0005310674455655868

Počet výskytov v paralelných vetách pre morfoslovo komission
Kmeň ako morfém:        0       Slovo ako slovo:        0
Relatívne zastúpenie slova máslo        0.0
Počet riadkov so slovom/kmeňom: máslo/másl      0
Relatívny počet riadkov:        0.0

Počet výskytov v paralelných vetách pre morfoslovo voita
Kmeň ako morfém:        62      Slovo ako slovo:        31
Relatívne zastúpenie slova máslo        0.1585677749360614
Počet riadkov so slovom/kmeňom: máslo/másl      51
Relatívny počet riadkov:        0.13043478260869565
```

[121]:
```
results_df = pd.DataFrame.from_dict(translation_frequencies).transpose().drop(0,␣
↪axis=1)
results_df.columns = ["TWfreq", "TWline_freq", "cross_rel_freq",␣
↪"cross_rel_line_freq"]
```

[122]:
```
results_df["translation_validityLine"] = results_df["cross_rel_freq"] /␣
↪results_df["cross_rel_freq"].sum()
results_df["translation_validityFreq"] = results_df["cross_rel_line_freq"] /␣
↪results_df["cross_rel_line_freq"].sum()
```

[123]:
```
results_df.sort_values(by="translation_validityLine", ascending=False)
```

[123]:

	TWfreq	TWline_freq	cross_rel_freq	cross_rel_line_freq	\
voita	62.0	51.0	0.158568	0.130435	

```
voi          202.0        120.0        0.072584             0.043119
ja             8.0          2.0        0.002124             0.000531
se             1.0          1.0        0.000266             0.000266
on             0.0          0.0        0.000000             0.000000
artiklan       0.0          0.0        0.000000             0.000000
asetuksen      0.0          0.0        0.000000             0.000000
komission      0.0          0.0        0.000000             0.000000

           translation_validityLine  translation_validityFreq
voita                      0.678969                  0.748115
voi                        0.310794                  0.247311
ja                         0.009096                  0.003046
se                         0.001141                  0.001528
on                         0.000000                  0.000000
artiklan                   0.000000                  0.000000
asetuksen                  0.000000                  0.000000
komission                  0.000000                  0.000000
```

[124]:
```
((results_df["translation_validityLine"] +
  results_df["translation_validityFreq"])/2).round(2) # translation_test
```

[124]:
```
se           0.00
on           0.00
artiklan     0.00
asetuksen    0.00
voi          0.28
ja           0.01
komission    0.00
voita        0.71
dtype: float64
```