P05

October 15, 2023

[1]:

```
import requests
import math
from collections import Counter
from collections import Counter
import nltk
from nltk.corpus import brown
import re
!pip install elotl
import elotl.corpus
!pip install subword-nmt
!pip install spacy
import spacy
spacy.load('en_core_web_sm')
spacy.load('es_core_news_sm')

nltk.download("cess_esp")
from nltk.corpus import cess_esp as cess
def lemmatize(words: list, lang="en") -> list:
    model = "en_core_web_sm" if lang == "en" else "es_core_news_sm"
    nlp = spacy.load(model)
    nlp.max_length = 1500000
    lemmatizer = nlp.get_pipe("lemmatizer")
    return [token.lemma_ for token in nlp(" ".join(words))]


axolotl = elotl.corpus.load("axolotl")
BIBLE_FILE_NAMES = {"spa": "spa-x-bible-reinavaleracontemporanea", "eng":␣
 ↪"eng-x-bible-kingjames"}

def get_bible_corpus(lang: str) -> str:
    file_name = BIBLE_FILE_NAMES[lang]
    r = requests.get(f"https://raw.githubusercontent.com/ximenina/
 ↪theturningpoint/main/Detailed/corpora/corpusPBC/{file_name}.txt.clean.txt")
    return r.text

def write_plain_text_corpus(raw_text: str, file_name: str) -> None:
    with open(f"{file_name}.txt", "w") as f:
        f.write(raw_text)
```

```
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: elotl in /home/xbmu/.local/lib/python3.8/site-
packages (0.0.1.16)
Requirement already satisfied: importlib-resources in
/home/xbmu/.local/lib/python3.8/site-packages (from elotl) (6.1.0)
Requirement already satisfied: future in /home/xbmu/.local/lib/python3.8/site-
packages (from elotl) (0.18.3)
Requirement already satisfied: zipp>=3.1.0 in
/home/xbmu/.local/lib/python3.8/site-packages (from importlib-resources->elotl)
(3.17.0)

[notice] A new release of pip is
available: 23.2.1 -> 23.3
[notice] To update, run:
python3 -m pip install --upgrade pip
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: subword-nmt in
/home/xbmu/.local/lib/python3.8/site-packages (0.3.8)
Requirement already satisfied: mock in /home/xbmu/.local/lib/python3.8/site-
packages (from subword-nmt) (5.1.0)
Requirement already satisfied: tqdm in /home/xbmu/.local/lib/python3.8/site-
packages (from subword-nmt) (4.66.1)

[notice] A new release of pip is
available: 23.2.1 -> 23.3
[notice] To update, run:
python3 -m pip install --upgrade pip
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: spacy in /home/xbmu/.local/lib/python3.8/site-
packages (3.7.1)
Requirement already satisfied: spacy-legacy<3.1.0,>=3.0.11 in
/home/xbmu/.local/lib/python3.8/site-packages (from spacy) (3.0.12)
Requirement already satisfied: spacy-loggers<2.0.0,>=1.0.0 in
/home/xbmu/.local/lib/python3.8/site-packages (from spacy) (1.0.5)
Requirement already satisfied: murmurhash<1.1.0,>=0.28.0 in
/home/xbmu/.local/lib/python3.8/site-packages (from spacy) (1.0.10)
Requirement already satisfied: cymem<2.1.0,>=2.0.2 in
/home/xbmu/.local/lib/python3.8/site-packages (from spacy) (2.0.8)
Requirement already satisfied: preshed<3.1.0,>=3.0.2 in
/home/xbmu/.local/lib/python3.8/site-packages (from spacy) (3.0.9)
Requirement already satisfied: thinc<8.3.0,>=8.1.8 in
/home/xbmu/.local/lib/python3.8/site-packages (from spacy) (8.2.1)
Requirement already satisfied: wasabi<1.2.0,>=0.9.1 in
/home/xbmu/.local/lib/python3.8/site-packages (from spacy) (1.1.2)
Requirement already satisfied: srsly<3.0.0,>=2.4.3 in
```

```
/home/xbmu/.local/lib/python3.8/site-packages (from spacy) (2.4.8)
Requirement already satisfied: catalogue<2.1.0,>=2.0.6 in
/home/xbmu/.local/lib/python3.8/site-packages (from spacy) (2.0.10)
Requirement already satisfied: weasel<0.4.0,>=0.1.0 in
/home/xbmu/.local/lib/python3.8/site-packages (from spacy) (0.3.2)
Requirement already satisfied: typer<0.10.0,>=0.3.0 in
/home/xbmu/.local/lib/python3.8/site-packages (from spacy) (0.9.0)
Requirement already satisfied: pathy>=0.10.0 in
/home/xbmu/.local/lib/python3.8/site-packages (from spacy) (0.10.2)
Requirement already satisfied: smart-open<7.0.0,>=5.2.1 in
/home/xbmu/.local/lib/python3.8/site-packages (from spacy) (6.4.0)
Requirement already satisfied: tqdm<5.0.0,>=4.38.0 in
/home/xbmu/.local/lib/python3.8/site-packages (from spacy) (4.66.1)
Requirement already satisfied: requests<3.0.0,>=2.13.0 in
/home/xbmu/.local/lib/python3.8/site-packages (from spacy) (2.31.0)
Requirement already satisfied: pydantic!=1.8,!=1.8.1,<3.0.0,>=1.7.4 in
/home/xbmu/.local/lib/python3.8/site-packages (from spacy) (2.4.2)
Requirement already satisfied: jinja2 in /home/xbmu/.local/lib/python3.8/site-
packages (from spacy) (3.1.2)
Requirement already satisfied: setuptools in
/home/xbmu/.local/lib/python3.8/site-packages (from spacy) (68.2.2)
Requirement already satisfied: packaging>=20.0 in
/home/xbmu/.local/lib/python3.8/site-packages (from spacy) (23.1)
Requirement already satisfied: langcodes<4.0.0,>=3.2.0 in
/home/xbmu/.local/lib/python3.8/site-packages (from spacy) (3.3.0)
Requirement already satisfied: numpy>=1.15.0 in
/home/xbmu/.local/lib/python3.8/site-packages (from spacy) (1.24.4)
Requirement already satisfied: annotated-types>=0.4.0 in
/home/xbmu/.local/lib/python3.8/site-packages (from
pydantic!=1.8,!=1.8.1,<3.0.0,>=1.7.4->spacy) (0.6.0)
Requirement already satisfied: pydantic-core==2.10.1 in
/home/xbmu/.local/lib/python3.8/site-packages (from
pydantic!=1.8,!=1.8.1,<3.0.0,>=1.7.4->spacy) (2.10.1)
Requirement already satisfied: typing-extensions>=4.6.1 in
/home/xbmu/.local/lib/python3.8/site-packages (from
pydantic!=1.8,!=1.8.1,<3.0.0,>=1.7.4->spacy) (4.7.1)
Requirement already satisfied: charset-normalizer<4,>=2 in
/home/xbmu/.local/lib/python3.8/site-packages (from
requests<3.0.0,>=2.13.0->spacy) (3.2.0)
Requirement already satisfied: idna<4,>=2.5 in /usr/lib/python3/dist-packages
(from requests<3.0.0,>=2.13.0->spacy) (2.8)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/lib/python3/dist-
packages (from requests<3.0.0,>=2.13.0->spacy) (1.25.8)
Requirement already satisfied: certifi>=2017.4.17 in
/home/xbmu/.local/lib/python3.8/site-packages (from
requests<3.0.0,>=2.13.0->spacy) (2023.5.7)
Requirement already satisfied: blis<0.8.0,>=0.7.8 in
/home/xbmu/.local/lib/python3.8/site-packages (from thinc<8.3.0,>=8.1.8->spacy)
```

```
(0.7.11)
Requirement already satisfied: confection<1.0.0,>=0.0.1 in
/home/xbmu/.local/lib/python3.8/site-packages (from thinc<8.3.0,>=8.1.8->spacy)
(0.1.3)
Requirement already satisfied: click<9.0.0,>=7.1.1 in
/home/xbmu/.local/lib/python3.8/site-packages (from typer<0.10.0,>=0.3.0->spacy)
(8.1.7)
Requirement already satisfied: cloudpathlib<0.16.0,>=0.7.0 in
/home/xbmu/.local/lib/python3.8/site-packages (from weasel<0.4.0,>=0.1.0->spacy)
(0.15.1)
Requirement already satisfied: MarkupSafe>=2.0 in
/home/xbmu/.local/lib/python3.8/site-packages (from jinja2->spacy) (2.1.3)

[notice] A new release of pip is
available: 23.2.1 -> 23.3
[notice] To update, run:
python3 -m pip install --upgrade pip

[nltk_data] Downloading package cess_esp to /home/xbmu/nltk_data…
[nltk_data]   Package cess_esp is already up-to-date!
```

0.1 Corpus Nahuatl (axolotl)

[2]:

```
train_rows_count = len(axolotl) - round(len(axolotl)*.3)
axolotl_train = axolotl[:train_rows_count]
axolotl_test = axolotl[train_rows_count:]
axolotl_words_vanilla_train = [word for row in axolotl_train for word in row[1].
 ↪lower().split()]
write_plain_text_corpus(" ".join(axolotl_words_vanilla_train),␣
 ↪"axolotl_plain_vanilla")
!subword-nmt learn-bpe -s 300 < axolotl_plain_vanilla.txt > axolotl_vanilla.
 ↪model
axolotl_test_words = [word for row in axolotl_test for word in row[1].lower().
 ↪split()]
axolotl_test_types = Counter(axolotl_test_words)
axolotl_singletons = [singleton for singleton in axolotl_test_types.items() if␣
 ↪singleton[1] == 1]
write_plain_text_corpus(" ".join(axolotl_test_words), "axolotl_plain_test")
!subword-nmt apply-bpe -c axolotl_vanilla.model < axolotl_plain_test.txt >␣
 ↪axolotl_vanilla_tokenized.txt
with open("axolotl_vanilla_tokenized.txt") as f:
    axolotl_test_tokenized = f.read().split()
axolotl_test_tokenized_types = Counter(axolotl_test_tokenized)
axolotl_singletons_tokenized = [singleton for singleton in␣
 ↪axolotl_test_tokenized_types.items() if singleton[1] == 1]
```

```
100%|########################################| 300/300 [00:02<00:00, 143.38it/s]
```

```
[3]: print("Axolotl Information")
     print("Tokens:", len(axolotl_test_words))
     print("Types (vanilla):", len(axolotl_test_types))
     print("Types (native BPE):", len(axolotl_test_tokenized_types))
     print("TTR (Vanilla):", len(axolotl_test_types)/len(axolotl_test_words))
     print("TTR (BPE):", len(axolotl_test_tokenized_types)/
       ↪len(axolotl_test_tokenized))
     print("Singletons:", len(axolotl_singletons))
     print("Singletons (Tokenized):", len(axolotl_singletons_tokenized))
```

```
Axolotl Information
Tokens: 86604
Types (vanilla): 25714
Types (native BPE): 451
TTR (Vanilla): 0.2969146921620249
TTR (BPE): 0.0017565178105453385
Singletons: 19553
Singletons (Tokenized): 16
```

0.2 Corpus Biblia en español

```
[4]: cess_sents = cess.sents()
     cess_words = cess.words()
     cess_plain_text = " ".join([" ".join(sentence) for sentence in cess_sents])
     cess_plain_text = re.sub(r"[-|_]", " ", cess_plain_text)
     with open("cess_plain.txt", "w") as f:
         f.write(cess_plain_text)
     !subword-nmt learn-bpe -s 300 < cess_plain.txt > cess.model
     spa_bible_plain_text = get_bible_corpus('spa')
     spa_bible_words = spa_bible_plain_text.replace("\n", " ").split()
     spa_bible_types = Counter(spa_bible_words)
     spa_bible_lemmas_types = Counter(lemmatize(spa_bible_words, lang="es"))
     write_plain_text_corpus(spa_bible_plain_text, "spa-bible")
     !subword-nmt apply-bpe -c cess.model < spa-bible.txt > spa_bible_tokenized.txt
     with open("spa_bible_tokenized.txt", "r") as f:
         tokenized_text = f.read()
     spa_bible_tokenized = tokenized_text.split()
     spa_bible_tokenized_types = Counter(spa_bible_tokenized)
```

```
100%|########################################| 300/300 [00:00<00:00, 396.91it/s]
```

```
[5]: print("Bible Spanish Information")
     print("Tokens:", len(spa_bible_words))
     print("Types (vanilla):", len(spa_bible_types))
     print("Types (lemmatized)", len(spa_bible_lemmas_types))
     print("Types (native BPE):", len(spa_bible_tokenized_types))
     print("TTR (Vanilla):", len(spa_bible_types)/len(spa_bible_words))
print("TTR (BPE):", len(spa_bible_tokenized_types)/len(spa_bible_tokenized))
```

```
Bible Spanish Information
Tokens: 30073
Types (vanilla): 3568
Types (lemmatized) 2313
Types (native BPE): 392
TTR (Vanilla): 0.11864463139693412
TTR (BPE): 0.006288904575498942
```

0.3 Entropía de un texto

La entropía de un texto es una medida que nos permite evaluar cuán impredecible o caótico es un conjunto de datos textual. Se utiliza ampliamente en teoría de la información y procesamiento de lenguaje natural para comprender la complejidad de un texto y su contenido informativo. Esta métrica se basa en la probabilidad de ocurrencia de símbolos individuales en el texto, lo que nos permite cuantificar cuánta información o incertidumbre hay en el texto.

La fórmula para calcular la entropía de un texto se presenta como:

$H(X) = -\sum_{i=1}^{n} p(x_i) \log_2(p(x_i))$

Donde $(H(X))$ es la entropía del texto, (n) es el número de símbolos únicos en el texto, $(p(x_i))$ es la probabilidad de que el símbolo (x_i) aparezca en el texto y (log_2) es el logaritmo en base 2. Cuanto mayor sea la entropía, más impredecible es el texto, mientras que una entropía baja indica un texto más predecible.

0.3.1 Entropía Corpus biblia español

[6]:

```
word_probabilities_spa_types = {word: count / len(spa_bible_words) for word,␣
 ↪count in spa_bible_types.items()}
entropy_spa_types = -sum(prob * math.log(prob, 2) for prob in␣
 ↪word_probabilities_spa_types.values())
```

[7]:

```
#debe sumar 1
total = sum(value for value in word_probabilities_spa_types.values())
total
```

[7]: 0.9999999999999347

[8]:

```
entropy_spa_types
```

[8]: 8.553905984312227

0.3.2 Entropía Corpus biblia español (tokenizado)

[9]:
```
word_probabilities_spa_tokenized_types = {word: count / len(spa_bible_words) for word, count in spa_bible_tokenized_types.items()}
entropy_spa_tokenized_types = -sum(prob * math.log(prob, 2) for prob in word_probabilities_spa_tokenized_types.values())
```

[10]:
```
entropy_spa_tokenized_types
```

[10]:
```
13.54100128153741
```

0.3.3 Entropía Corpus Axolotl Nahuatl

[11]:
```
word_probabilities_nah_types = {word: count / len(axolotl_test_words) for word, count in axolotl_test_types.items()}
entropy_nah_types = -sum(prob * math.log(prob, 2) for prob in word_probabilities_nah_types.values())
```

[12]:
```
#debe sumar 1
total = sum(value for value in word_probabilities_nah_types.values())
total
```

[12]:
```
1.0000000000001217
```

[13]:
```
entropy_nah_types
```

[13]:
```
11.415959291241606
```

0.3.4 Entropía Corpus Axolotl Nahuatl (tokenizado)

[14]:
```
word_probabilities_nah_tokenized_types = {word: count / len(axolotl_test_words) for word, count in axolotl_test_tokenized_types.items()}
entropy_nah_tokenized_types = -sum(prob * math.log(prob, 2) for prob in word_probabilities_nah_tokenized_types.values())
```

[15]:
```
entropy_nah_tokenized_types
```

[15]:
```
18.66060699433508
```

0.4 Preguntas

0.4.1 ¿Aumento o disminuyó la entropia para los corpus?

En los dos corpus aumentó la entropía

0.4.2 ¿Qué significa que la entropia aumente o disminuya en un texto?

Cuando aumenta la entropía, significa que dada una palabra es más difícil predecir la siguiente, es decir que el texto es más aleatorio.

0.4.3 ¿Como influye la tokenizacion en la entropía de un texto?

Se pierde información y hace más difícil predecir un texto, sin embargo se reduce la cantidad de palabras de una sola ocurrencia en el corpus (singletons) lo cual hace más fácil el análisis del texto.

[ ]: