**LEXICAL DISTANCE PROJECT SOURCING AND METHODOLOGY**

**Sources**

This section lists all source materials used during the research. Sources are for both documents analyzed, and the Python packages used.

**Documents analyzed:**

This section lists the documents used for analysis during the research.

* Title: UN Resolution 55/100
  + Type: Legal
  + Source: https://opus.nlpl.eu/UN/v20090831/ar-en\_sample.html
  + Resolution 55/100, [on the report of the Third Committee (A/55/602/Add.2 and Corr.1)]
  + Begins with:
    - “Reaffirming that the promotion and protection of all human rights and fundamental freedoms must be considered”
  + Ending with:
    - “… or acting in any other manner inconsistent with the purposes of the United Nations”
  + Citing:
    - [Alexandre Rafalovitch, Robert Dale. 2009. United Nations General Assembly Resolutions: A Six-Language Parallel Corpus. In Proceedings of the MT Summit XII, pages 292-299, Ottawa, Canada, August](http://uncorpora.org/Rafalovitch_Dale_MT_Summit_2009.pdf)
    - J. Tiedemann, 2012, [*Parallel Data, Tools and Interfaces in OPUS.*](http://www.lrec-conf.org/proceedings/lrec2012/pdf/463_Paper.pdf) In Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC 2012)
* Title: UN Resolution 55/122
  + Type: Technical / Scientific
  + Source: <https://opus.nlpl.eu/MultiUN.php>
    - https://undocs.org/pdf?symbol=en/A/74/408
  + A/RES/55/122, International Cooperation in the Peaceful Uses of Outer Space
  + Begins with:
    - “Deeply convinced of the common interest of mankind in promoting and expanding the exploration and use of outer space for peaceful…”
  + Ending with:
    - “…Endorses the recommendation of the Committee that the Legal Subcommittee, at its fortieth session, taking into account the concerns”
  + Citing:
    - [Alexandre Rafalovitch, Robert Dale. 2009. United Nations General Assembly Resolutions: A Six-Language Parallel Corpus. In Proceedings of the MT Summit XII, pages 292-299, Ottawa, Canada, August](http://uncorpora.org/Rafalovitch_Dale_MT_Summit_2009.pdf)
    - J. Tiedemann, 2012, [*Parallel Data, Tools and Interfaces in OPUS.*](http://www.lrec-conf.org/proceedings/lrec2012/pdf/463_Paper.pdf) In Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC 2012)
* Title: UN Resolution 55/101:
  + Type: Legal
  + Source: <https://opus.nlpl.eu/UN/v20090831/ar-en_sample.html>
  + Begins with:
    - “The General Assembly ,Recalling that , in accordance with Article 56 of the Charter of the United Nations , all”
  + Ends with:
    - Reaffirming further the various articles of the Charter setting out the respective powers and functions of the General Assembly , the Security Council and the Economic and Social Council , as the paramount framework for the achievement of the purposes of the United Nations
  + Citing:
    - [Alexandre Rafalovitch, Robert Dale. 2009. United Nations General Assembly Resolutions: A Six-Language Parallel Corpus. In Proceedings of the MT Summit XII, pages 292-299, Ottawa, Canada, August](http://uncorpora.org/Rafalovitch_Dale_MT_Summit_2009.pdf)
    - J. Tiedemann, 2012, [*Parallel Data, Tools and Interfaces in OPUS.*](http://www.lrec-conf.org/proceedings/lrec2012/pdf/463_Paper.pdf) In Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC 2012)
* Title: UN Resolution 70/1
* Source: <https://www.un.org/en/development/desa/population/migration/generalassembly/docs/globalcompact/A_RES_70_1_E.pdf>
* Begins with:
  + “Reaffirming its resolution 70/1 of 25 September 2015, entitled “Transforming our world: the 2030 Agenda for Sustainable Development”, in which it adopted a comprehensive...”
* Ends with:
  + “recognizes that developing countries with fragile mountainous ecosystems are among the countries that are particularly vulnerable to the adverse effects of climate change.”
* Citing:
  + General Assembly Resolution 70/1, Transforming our world: the 2030 Agenda for Sustainable Development, A/Res/70/1 (25 October 2015), available from digitallibrary.un.org/record/3923923.
* Title: en/2005/NPT\_CONF2005\_32.xml.gz
  + Type: Legal
  + Source: <https://opus.nlpl.eu/MultiUN.php>

**Python Packages:**

This section lists all of the Python packages used, along with official documentation if available.

*All packages used:*

* unidecode: https://pypi.org/project/Unidecode/
* re: https://docs.python.org/3/library/re.html
* ntlk: http://www.nltk.org/
  + stem: https://www.nltk.org/api/nltk.stem.html
    - snowball: https://www.nltk.org/\_modules/nltk/stem/snowball.html
      * SnowballStemmer
  + corpus: https://www.nltk.org/api/nltk.corpus.html
    - stopwords
  + translate: https://www.nltk.org/api/nltk.translate.html
    - Alignment
* numpy: https://numpy.org/
* pandas: https://pandas.pydata.org/
* seaborn: https://seaborn.pydata.org/
* matplotlib: https://matplotlib.org/
  + Pyplot
* transliterate: https://pypi.org/project/transliterate/
  + translit
  + get\_available\_language\_codes
* deep-translator:
  + https://github.com/nidhaloff/deep-translator

*Packages by Python File:*

* Parser.py:
  + unidecode
  + re
  + ntlk
    - stem
      * snowball
        + SnowballStemmer

https://www.nltk.org/\_modules/nltk/stem/snowball.html

* + - corpus
      * stopwords
* Ldistance.py:
  + re
  + matplotlib
  + numpy
  + ntlk
    - stem
      * snowball
        + SnowballStemmer
    - corpus
      * stopwords
* Parser.py:
  + matplotlib
    - pyplot
  + seaborn
  + numpy
  + Pandas
* SheetMaker.py:
  + Deep-translator
  + Numpy
  + pandas
* BagOfConsonants.py:
  + N/A

**Concept**

This section is designed to detail the linguistic concepts behind the undertaking of this project and the methods use.

***Base Reasoning:***

Many current methods of measuring lexical similarity rely on **the use of loanwords and cognates.** This project aims to demonstrate another measure of lexical similarity through word-to-word comparisons between two languages.

In this comparison, words with same meanings will be aligned as a pair. Each word in the pair will be stemmed and then have any vowels and punctuation removed, leaving only the consonants. A Levenshtein distance will be calculated for the two sets of consonants. See Figure 3.1 for an example.

**Figure 3.1: English to German Comparison**

|  |  |  |
| --- | --- | --- |
|  | **English Word** | **German Word** |
| **Step 1 (Alignment):** | **Father** | **Vater (father)** |
| **Step 2 (Remove vowels):** | **Fthr** | **Vtr** |
| **Step 3: (Levenshtein distance):** |  |  |

The reason for this is that consonants remain more stable between languages than vowels do. In the example from figure 3.1, the ‘a’ and ‘e’ sounds will differ between English and German. In contrast, the consonants of both words preserve the similarity between the two.

**Methodology**

This section is designed to describe the methods by which the research was done. The section does not intend to provide high-level explanation of the methods from a linguistic standpoint, but does intend to explain certain choices that were made during research, and to explain how the code used in research was formulated.

***General information:***

*On the source material:*

The source (“parent”) document was taken from OPUS (<https://opus.nlpl.eu/MultiUN.php>). The document is UN Resolution 55/100. The analysis was performed on an excerpt of the document.

This document was selected as an optimal example of text in a legal and political context. Further, the UN Databases provide an ample resource for six major world languages: Arabic, English, French, Mandarin Chinese, Russian, and Spanish.

*On the generation of documents:*

The parent document (Resolution 55/100 excerpt) was in English. The document was then translated to all other languages used in the research via Google Translate. Thus the one English source document was translated into multiple (“child”) documents.

*On the use of Google Translate:*

The research involved the use of Google Translate for generation of documents rather than using the source documents of different languages (i.e.: using the source document created from a human translator in Spanish or any other language) because Google Translate provides a translation that focuses more on linear relationships between words, and would be more apt to provide a word that is closer in meaning than a human translator may, since the human translator may be more prone to change wording to suite tone or other subtleties.

***Parsing Information:***

*On the removal of common words:*

The method by which common words were removed involved the use of the Python package ntlk. From ntlk, package corpus’s stopwords was used. stopwords is essentially a list of the most common words in each language. Thus, the removal involved comparing the words found in the documents and removing those that also appeared in the stopwords list.

*On the use of the stemmer:*

The Snowball Stemmer from the Python package ntlk was used to stem words. The general process used was to create an instance of the stemmer, for example stemmer = SnowballStemmer(“english”). Then, the stemmer was used to stem each word found in the analyzed documents. For example, for a word such as “running”, the stemmer returns “run”.

*On the use of unidecode:*

The Python package unidecode was used to change characters with accents to their respective non-accented characters (ie: ‘é’ would be changed to ‘e’).

*On the removal of punctuation and numbers:*

To remove punctuation and numbers from the documents, the Python package re was used. A regular expression was created that included all the punctuation and number characters that were to be removed, and this expression was then called to remove the appropriate characters from the documents. Note that this was done after using unidecode.

*On the removal of vowels*

To remove the vowels from the documents, the Python package re was used. A regular expression that contained only consonants was used to extract the consonants from a string without changing the order of characters. Note that this was done after using unidecode.

***Calculation Information:***

*On the calculation of the Levenshtein Distance:*

The Levenshtein distance was calculated using numpy. The Levenshtein distance was calculated using a distance matrix outlined by Gad in “Measuring Text Similarity Using the Levenshtein Distance” (<https://blog.paperspace.com/measuring-text-similarity-using-levenshtein-distance/>). The primary use of numpywas for populating a matrix easily.

Additionally, a procedure was introduced to the calculations to account for word length. This procedure is **taken** from a calculation described in [Schepens (2008),](https://www.cambridge.org/core/journals/bilingualism-language-and-cognition/article/abs/distributions-of-cognates-in-europe-as-based-on-levenshtein-distance/9B0B8913C6A5F39984B11A4063F55FDB) using the longer word length. See Figure 4.1 for the calculation. This created a scale of potential distances between 0 to 1, **in which a distance of 1 would signal identical pairs and a score of 0 would indicate a maximum amount of edits was required.**

**Figure 4.1**

*On the Alignment Process:*

To properly generate Levenshtein distances between many pairs of words, alignments were used. An alignment is a relationship between one language’s word or words to another language’s word or words. Note that alignments are based on typical indexing, ie: 0 is the first index, 1 is the second, et cetera. For example, the tuple (0,2) between English and French means that the first word in the English document corresponds to the third word in the French document. This represents a one-one relationship. A dictionary having integer keys and tuple or list values, such as { … 0:[0,1], ...} would be a one-many relationship, and a dictionary having tuple keys and tuple or list values, such as { … (0,1):(1,2), … } would be a many-many relationship.

Depending on the relationship between a word or words in one language to a word or words in another language (ie: one-one, one-many, or many-many), the calculation method adjusted to the relationship.

When calculating a distance that involves a one-many relationships, the multiple words that mapped to one word were concatenated into one string. This string was then compared to the one-word string, and an Levenshtein distance was taken. Similarly, when calculating a distance that involves a many-many relationship, both languages need to have the concatenation done on their respective words.

*On the measuring of samples:*

To obtain an average Levenshtein distance between two samples, alignments of each word from one language to another language were taken using the methods described in “*On the Alignment Process*”. Each word found in the sample after removing common words was paired to another word in the other language’s sample (also after removing common words). Then each alignment yielded a Levenshtein distance, which were then averaged over all distances. This average was recorded as the distance between one language to another.

*On the generation of graphs:*

The correlation heatmap was generated using seaborn’s heatmap function.

**Use of Each Python Package:**

This section goes into further detail on how each Python package was used in the research.