Assessing Translatability

An Algorithmic Procedure for Identifying and Scoring Difficult-to-Translate German to English Multi-Noun Compounds by Leveraging Bilingual Corpus Data

Christian Johnson

Table of Contents

Preview of Results

0.8275411579581397	Salzwasser
0.7071156613626228	Jahrhundert
0.4646577380952381	Arkadengangs
0.44820503717850413	Landzunge
0.40021340983894055	Lebensraum
0.35532614425937425	Gegensatz
0.301353143699554	Zugzwang
0.30065816247494626	Feierabend
0.2534613466598761	Schnapsidee
0.2534584945481294	Treppenwitz
0.1970239431014929	Weltschmerz
0.15333555979140517	Geisterfahrer
0.15002381089596242	Fernweh
0.09117387249626349	Torschlusspanik
0.06839805282082635	Ohrwurm

Figure 1: The above list shows the German noun compounds extracted from a short source text[1] and scored according to English 'translatability' using a specialized algorithm. Results are organized from highest to lowest, with a high score indicating that translation into English is unproblematic. From this list we see that scoring confirms many intuitions about the potential difficulties of translating certain terms.

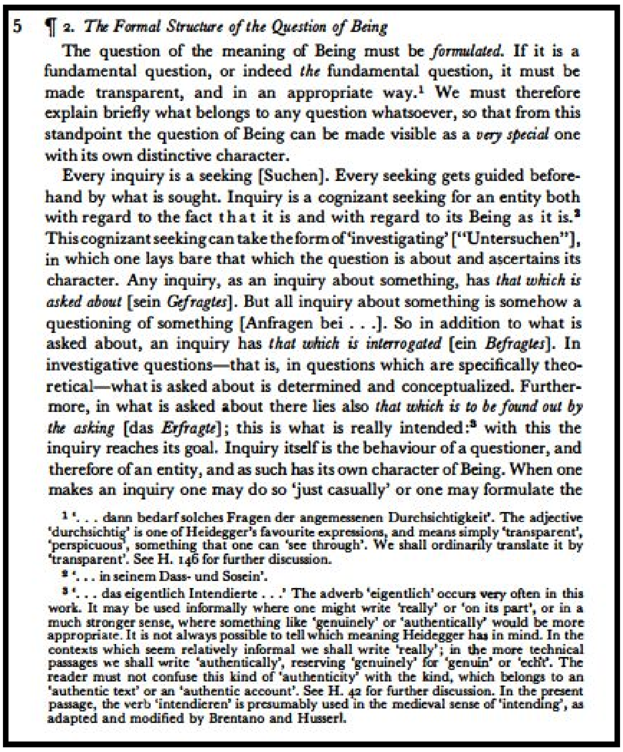
Introduction

Motivation

In many domains it is advantageous to identify those elements of a text which will prove most difficult or in need of inspection when translating into a second language. Although machine translation methods automate with relative success the translation of many texts into other languages, there remains in most cases a need for human intervention to achieve fluent prose translations or appropriate approximation of rare and specialized terms. Thus, a procedure to systematically identify text elements which will pose a challenge to automated translation methods will facilitate semi-automated translation by directing human translators to areas most in need of circumlocution or additional explanation. Figure 2 demonstrates this phenomenon in an English translation of a German philosophical work[2], where words the translator has deemed problematic are afforded footnotes wherein the German source and additional explanations or context regarding the translation are provided.

[1] See Appendices

[2] Heidegger (1962)



5 ¶ 2. *The Formal Structure of the Question of Being*

The question of the meaning of Being must be *formulated*. If it is a fundamental question, or indeed *the* fundamental question, it must be made transparent, and in an appropriate way.[1] We must therefore explain briefly what belongs to any question whatsoever, so that from this standpoint the question of Being can be made visible as a *very special* one with its own distinctive character.

Every inquiry is a seeking [Suchen]. Every seeking gets guided beforehand by what is sought. Inquiry is a cognizant seeking for an entity both with regard to the fact that it is and with regard to its Being as it is.[2] This cognizant seeking can take the form of 'investigating' ["Untersuchen"], in which one lays bare that which the question is about and ascertains its character. Any inquiry, as an inquiry about something, has *that which is asked about* [sein *Gefragtes*]. But all inquiry about something is somehow a questioning of something [Anfragen bei . . .]. So in addition to what is asked about, an inquiry has *that which is interrogated* [ein *Befragtes*]. In investigative questions—that is, in questions which are specifically theoretical—what is asked about is determined and conceptualized. Furthermore, in what is asked about there lies also *that which is to be found out by the asking* [das *Erfragte*]; this is what is really intended:[3] with this the inquiry reaches its goal. Inquiry itself is the behaviour of a questioner, and therefore of an entity, and as such has its own character of Being. When one makes an inquiry one may do so 'just casually' or one may formulate the

[1] '. . . dann bedarf solches Fragen der angemessenen Durchsichtigkeit'. The adjective 'durchsichtig' is one of Heidegger's favourite expressions, and means simply 'transparent', 'perspicuous', something that one can 'see through'. We shall ordinarily translate it by 'transparent'. See H. 146 for further discussion.

[2] '. . . in seinem Dass- und Sosein'.

[3] '. . . das eigentlich Intendierte . . .' The adverb 'eigentlich' occurs very often in this work. It may be used informally where one might write 'really' or 'on its part', or in a much stronger sense, where something like 'genuinely' or 'authentically' would be more appropriate. It is not always possible to tell which meaning Heidegger has in mind. In the contexts which seem relatively informal we shall write 'really'; in the more technical passages we shall write 'authentically', reserving 'genuinely' for 'genuin' or 'echt'. The reader must not confuse this kind of 'authenticity' with the kind, which belongs to an 'authentic text' or an 'authentic account'. See H. 42 for further discussion. In the present passage, the verb 'intendieren' is presumably used in the medieval sense of 'intending', as adapted and modified by Brentano and Husserl.

Figure 2: Influence of relative 'translatability' on translation practices.

Our study aims to implement an algorithmic procedure to identify words of this caliber and score them according to the notion of English 'translatability'; how readily the semantic content of German words can be adequately captured by their translation attempts into English. A low translatability score indicates that further context is required to elaborate the original meaning of this term, or, in the case of machine translation, that specialized techniques are required to retrieve an adequate translation.

Definition of Terms

Processing words according to their 'translatability' requires a notion of a word and a model of translation as a process of semantic exchange. Given the context of our home institution, our study focuses on German to English word translation, a context which simplifies the problem of cross-language word comparison given the orthographic similarities between our source and target languages. In this case, 'words' are defined orthographically as segments separated by spacing conventions, a set which in German includes the subsets of nouns and noun compounds. Since the German orthography reform of 1996[3], nouns and noun compounds are readily identified by first letter capitalization.

We select noun compounds as a suitable test case for translatability scoring given the wealth of German compounds associated with specialized and unique meanings, and the practicality of limiting our computations to a subset of German words for which the act of translation can be reduced to a systematic, assessable model. Because many German noun compounds correspond to English compounds of a similar structure (e.g. "Orangensaft" and "Orange Juice"), the task of translating these words can, in idealized

[3] Duden (1996)

cases, be conceptualized as a process of splitting, segment translation, and rejoining into recognizable English compounds from which the meaning of the original German compound can be retrieved. The challenge to this model is words for which splitting and translating components cannot yield an adequate English compound or phrase which accurately reproduces the full meaning of the original German compound. Words of this order include such specialized and colloquially-employed words as "Weltschmerz," where the English words "world" and "pain" are in themselves not adequate to compositionally reconstruct the specialized feeling intended by the German.

If a German speaker considers the above example he may identify other words of similar translational difficulty. We employ the moniker 'untranslatable' to informally describe these cases we seek to identify through our algorithm. However, this concept may be extended to the construction of compounds for which no English equivalent exists or would be expected. For example: although the valid German compound "Fingertanz" may be composed with various intended meanings, this compound is not a commonly-accepted lexeme and it is unlikely that it corresponds in German to any stable meaning which might be translated into English. Our study concerns only German compounds for which there exists a single, stable meaning. This simplifying assumption allows us to control for meaning in the source language, thus rendering polysemy in the target language a potential indicator of inconsistent translations, ergo 'untranslatability.' However, this assumption poses problems for some polysemous German compounds which will be examined later.

Employing the assumptions of Wierzbicka and Goddard's lexical semantic typology[4], we aim to identify translatability gaps only for source language concepts which could conceivably be derived from existing semantic primes or molecules in the target language, where the source concept "could be expressed", but is not directly tied to a single word form and is perhaps only referenceable through circumlocution or paraphrase. Although the concept of "existential anxiety regarding the state of the world" can be expressed in English, the German concept of "Weltschmerz" cannot be composed in English with recourse to the compound segments alone, thus rendering it of note for our study. If our algorithm is successful, words such as "Weltschmerz" should receive low translatability scores, while prototypical compounds such as "Orangensaft" should receive high scores.

Prior Research

Typically, identification of the types of words we here seek to quantitatively assess is done through informal means such as introspection, resulting in varied word lists which lack standardized methods for collection and validation of their elements.[5] Although many quantitative translation assessment procedures such as BLEU scoring[6] exist, these measures do not assess the correspondence or suitability of source language segments to a target language, but instead focus on the fluency or quality of a final translation itself. Some work has approached formal translatability assessment, such as the cognitive frame analysis of Zakaria[7], but little exists in the form of implemented procedures for identification and scoring. In situations where so-called out-of-vocabulary (OOV) words are already identified,

[4] Wierzbicka and Goddard (2014)

[5] Smith (2018), University of Michigan (2017)

[6] BLEU. Wikipedia.

[7] Zakaria (2017)

there exists a greater variety of proposals for handling these difficult translations, such as the procedure outlined by Paul et al[8]. Our study seeks to complement this field of research by proposing an identification and scoring methodology which can be integrated before machine translation procedures for OOV words. Although the words considered in our study represent only a small subset of OOV words limited to a specific language pair, we anticipate the procedure here outlined as a first step towards more generalized and powerful implementations.

Assumptions

From the terms and scope described above we identify several assumptions upon which our study's algorithm operates.

1. Compounds must be longer than five characters. This assumption is justified by the average length of German nouns necessary to compose a compound.
2. Compounds must be accepted German words.
3. Compound elements must have viable English translation.
4. A word is highly translatable if the source meaning can be reconstructed only through recourse to the source compound constituents.
5. A word is highly translatable if there is little inconsistency in its translations into the target language.
6. Compounds must have stable meanings in the source language.

Methodology

Algorithm Description

Overview

The algorithm described below was designed to implement the extraction and scoring of translation-problematic segments such as those described above: German noun compounds for which English lacks a direct semantic equivalent or corresponding compound which can reliably reconstitute the meaning of the German original. Once candidates are extracted they are scored according to their 'translatability' such that words with a higher score are deemed more translatable and thus less problematic for purposes of machine translation etc. Words receiving low scores should represent a subset of OOV words which will benefit from additional explanation or more elaborate translation procedures.

The algorithm is implemented in Python 3.7 with a variety of supporting libraries documented below and in the appendices. The script can be run uninterrupted when provided a German source text from which the relevant words are extracted and scored. In addition to the relevant dependencies, the algorithm also leverages source and target corpus files which must be accessible to the script.

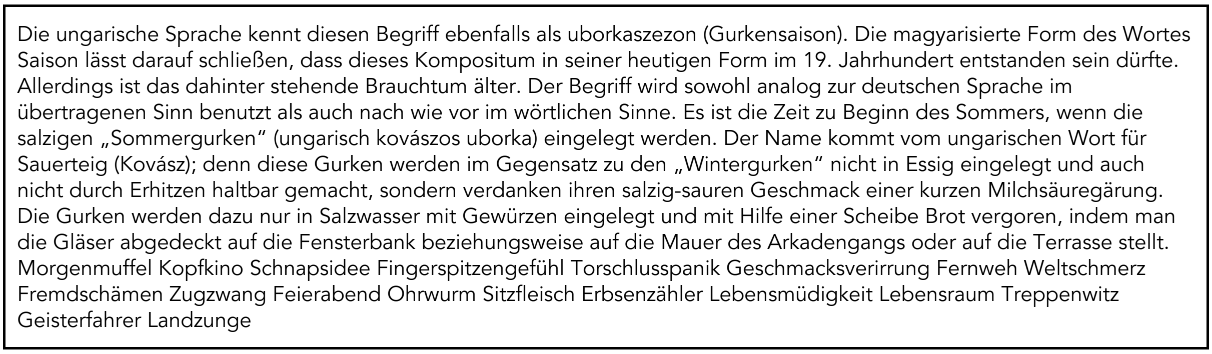
[8] Paul et al (2009)

Data Management

Throughout the subsequent phases of the algorithm, words and associated data necessary for classification and scoring are stored in a Python dictionary which is iteratively updated after each phase of the algorithm. Some data, such as translation segments extracted from target corpora, are stored in separate files to lower memory cost during computation.

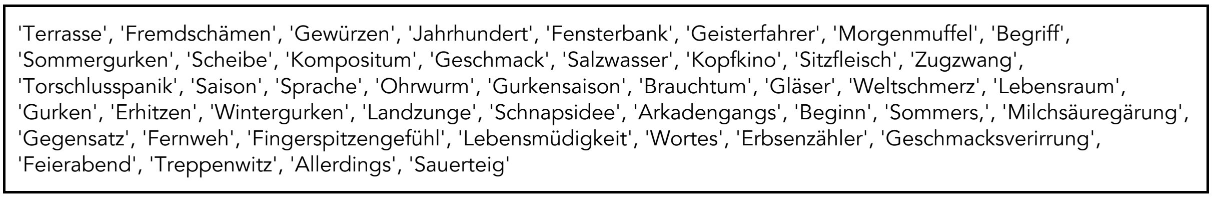
Noun Extraction

Our study considers only German noun compounds, a subclass of German nouns. Thus, the first phase of the algorithm extracts German nouns. As mentioned, the German spelling reform renders noun identification from a source text relatively straightforward. A regular expression was employed to extract the relevant segments into a list which was then controlled for duplicates and divested of segments longer than five characters per Assumption 1. Figure 3 shows the source text from which the words scored in Figure 1 were derived. The text is drawn from a Wikipedia article[9] to which additional words deemed important to the evaluation of the procedure were added. After noun extraction and controlling, the noun list in Figure 4 is obtained.



Die ungarische Sprache kennt diesen Begriff ebenfalls als uborkaszezon (Gurkensaison). Die magyarisierte Form des Wortes Saison lässt darauf schließen, dass dieses Kompositum in seiner heutigen Form im 19. Jahrhundert entstanden sein dürfte. Allerdings ist das dahinter stehende Brauchtum älter. Der Begriff wird sowohl analog zur deutschen Sprache im übertragenen Sinn benutzt als auch nach wie vor im wörtlichen Sinne. Es ist die Zeit zu Beginn des Sommers, wenn die salzigen „Sommergurken" (ungarisch kovászos uborka) eingelegt werden. Der Name kommt vom ungarischen Wort für Sauerteig (Kovász); denn diese Gurken werden im Gegensatz zu den „Wintergurken" nicht in Essig eingelegt und auch nicht durch Erhitzen haltbar gemacht, sondern verdanken ihren salzig-sauren Geschmack einer kurzen Milchsäuregärung. Die Gurken werden dazu nur in Salzwasser mit Gewürzen eingelegt und mit Hilfe einer Scheibe Brot vergoren, indem man die Gläser abgedeckt auf die Fensterbank beziehungsweise auf die Mauer des Arkadengangs oder auf die Terrasse stellt. Morgenmuffel Kopfkino Schnapsidee Fingerspitzengefühl Torschlusspanik Geschmacksverirrung Fernweh Weltschmerz Fremdschämen Zugzwang Feierabend Ohrwurm Sitzfleisch Erbsenzähler Lebensmüdigkeit Lebensraum Treppenwitz Geisterfahrer Landzunge

Figure 3: Original Text



'Terrasse', 'Fremdschämen', 'Gewürzen', 'Jahrhundert', 'Fensterbank', 'Geisterfahrer', 'Morgenmuffel', 'Begriff', 'Sommergurken', 'Scheibe', 'Kompositum', 'Geschmack', 'Salzwasser', 'Kopfkino', 'Sitzfleisch', 'Zugzwang', 'Torschlusspanik', 'Saison', 'Sprache', 'Ohrwurm', 'Gurkensaison', 'Brauchtum', 'Gläser', 'Weltschmerz', 'Lebensraum', 'Gurken', 'Erhitzen', 'Wintergurken', 'Landzunge', 'Schnapsidee', 'Arkadengangs', 'Beginn', 'Sommers,', 'Milchsäuregärung', 'Gegensatz', 'Fernweh', 'Fingerspitzengefühl', 'Lebensmüdigkeit', 'Wortes', 'Erbsenzähler', 'Geschmacksverirrung', 'Feierabend', 'Treppenwitz', 'Allerdings', 'Sauerteig'

Figure 4: Extracted Noun List

Compound Splitting and Extraction

From the subset of German nouns must be extracted the relevant subset of noun compounds. Identifying this word class is a research level task in its own right and several algorithms implemented in Python libraries exist to facilitate this identification. Ultimately, a modified version of the CharSplit algorithm[10] was selected given its demonstrated accuracy

[9] Sauregurkenzeit. Wikipedia.

[10] Tuggener (2016)

of 95% on a German test corpus and successful implementation in studies such as Hätty et al[11]. This splitting algorithm uses n-gram based probabilities trained on a Wikipedia corpus to identify the most likely location for a split in a given word. The algorithm also provides a split confidence metric between zero and one, later used to filter compound noun candidates. The algorithm was modified to score splits based on an average of beginning, middle, and word-ending slice probabilities rather than an equation in which beginning slice probabilities were subtracted from the sum of middle and ending slice probabilities. This minor alteration is seen in lines 76 and 77 of the source code included in the GitHub repository referenced in the Appendices. This change yielded more accurate confidence scores than the original algorithm. Otherwise, the algorithm was employed as intended.

To exclude from the candidate list shown in Figure 4 nouns which are not compounds, we exclude words which do not yield from among their potential splits a maximum split confidence of above 0.2. Tests revealed this cutoff sufficient to exclude the majority of irrelevant nouns while maintaining relevant compounds, although as seen in Figure 5, many splits produce segments which are not viable German nouns. This filtering error is a result of the n-gram based method which permits segments resembling German nouns or morphemes. Thus, post-split segments were themselves evaluated as an additional filter.

```
[[0.30261903623110475, 'Ter', 'Rasse'], [0.2627755300154281, 'Terr', 'Asse'], [0.45223013987859595, 'Fremd', 'Schämen'],
[0.28004018914257595, 'Jahr', 'Hundert'], [0.33541435142280873, 'Fenster', 'Bank'], [0.31251828894504025, 'Geister',
'Fahrer'], [0.6553763440860215, 'Morgen', 'Muffel'], [0.22631043274205218, 'Begr', 'Iff'], [0.34284027490610336,
'Sommer', 'Gurken'], [0.40419759781150844, 'Sche', 'Ibe'], [0.21773017483912183, 'Sch', 'Eibe'], [0.27627696846763783,
'Geschm', 'Ack'], [0.3155046301647851, 'Salz', 'Wasser'], [0.3416215799044943, 'Kopf', 'Kino'], [0.330379327964933, 'Sitz',
'Fleisch'], [0.38193774028571603, 'Zug', 'Zwang'], [0.6473419572385599, 'Torschluss', 'Panik'], [0.3108512773553841,
'Sai', 'Son'], [0.3195864122528014, 'Ohr', 'Wurm'], [0.3971189050110809, 'Gurken', 'Saison'], [0.31502154796986487,
'Brauch', 'Tum'], [0.33848108324992443, 'Welt', 'Schmerz'], [0.27875584121217767, 'Weltsch', 'Merz'],
[0.32375219433872676, 'Lebens', 'Raum'], [0.26986542759763377, 'Gur', 'Ken'], [0.33968737876445415, 'Winter',
'Gurken'], [0.2986115516722687, 'Land', 'Zunge'], [0.3380868046056452, 'Schnaps', 'Idee'], [0.3026685189198815,
'Arkaden', 'Gangs'], [0.22712418300653595, 'Beg', 'Inn'], [0.44280653266828385, 'Milchsäure', 'Gärung'],
[0.3554954013796291, 'Milch', 'Säuregärung'], [0.2648251464379348, 'Gegen', 'Satz'], [0.21627683037430603, 'Gegens',
'Atz'], [0.40370197264152397, 'Fern', 'Weh'], [0.3366954851104707, 'Fingerspitzen', 'Gefühl'], [0.4561609700445278,
'Lebens', 'Müdigkeit'], [0.32994946230240346, 'Erbsen', 'Zähler'], [0.6113890944419594, 'Geschmacks', 'Verirrung'],
[0.3225394651540662, 'Feier', 'Abend'], [0.23721778854858702, 'Treppen', 'Witz'], [0.3413953541282734, 'All', 'Erdings'],
[0.2943121087824102, 'Aller', 'Dings'], [0.3282415092082421, 'Sauer', 'Teig']]
```

Figure 5: Compounds Extracted Based on CharSplit Split Scores

Several external resources were introduced to filter the split compound candidates. The Python SpellChecker library[12] supports several languages including German, but is trained on a relatively limited set of German nouns. To capture a larger set of valid German nouns, a web dictionary API was introduced based on the Duden CLI module by Bosák[13] and combined into an or-condition which would capture from the aforementioned list only those compounds for which both segments were either validated as correctly-spelled German words by the SpellChecker or as commonly-accepted German nouns in the Duden Dictionary. Neither of these conditions could be used in isolation as tests showed they each alone excluded some relevant segments. The Duden API combined with SpellChecker represents the implementation of Assumption 2, which stipulates that compounds must be

[11] Hätty et al (2018)
[12] Barrus (2019)
[13] Bosák (2018)

accepted German lexemes with a relatively stable meaning.We thereby interpret inclusion in the Dude Dictionary as a verification of German lexicon inclusion.

The above conditions implementing Assumption 2 remove many invalid compound candidates, as seen in Figure 6, but permit rare cases for which both segments are technically valid nouns which nevertheless do not produce a true compound, such as the word "Allerdings," discussed in greater detail in the Evaluation section.

```
[['Ter', 'Rasse'], ['Jahr', 'Hundert'], ['Fenster', 'Bank'], ['Geister', 'Fahrer'], ['Morgen', 'Muffel'], ['Sommer',
'Gurken'], ['Sche', 'Ibe'], ['Salz', 'Wasser'], ['Kopf', 'Kino'], ['Sitz', 'Fleisch'], ['Zug', 'Zwang'], ['Torschluss',
'Panik'], ['Ohr', 'Wurm'], ['Gurken', 'Saison'], ['Brauch', 'Tum'], ['Welt', 'Schmerz'], ['Lebens', 'Raum'], ['Gur',
'Ken'], ['Winter', 'Gurken'], ['Land', 'Zunge'], ['Schnaps', 'Idee'], ['Arkaden', 'Gangs'], ['Beg', 'Inn'],
['Milchsäure', 'Gärung'], ['Gegen', 'Satz'], ['Fern', 'Weh'], ['Fingerspitzen', 'Gefühl'], ['Lebens', 'Müdigkeit'],
['Erbsen', 'Zähler'], ['Geschmacks', 'Verirrung'], ['Feier', 'Abend'], ['Treppen', 'Witz'], ['All', 'Erdings'], ['Aller',
'Dings'], ['Sauer', 'Teig']]
```

Figure 6: The list of compound candidates after Duden and SpellChecker condition.

To control for cases of proper or rare nouns irrelevant to our study, we introduced a further condition which translated each of the segments into English and immediately removed those candidates for which both segments could not produce a distinct English translation. This condition is based on Assumption 3, which stipulates that German compound components must have direct English translations, a requirement important for translatability scoring based on Assumption 4.

We employed GoogleTranslate API[14] for translation and validation of compound candidate segments. Segments fed into the translator would return either a translated value or a false value if the translation was identical to the source token, an indication that the translator API could find no suitable translation and that the segment was likely not a valid German word or was a proper noun. Based on these returned values, it could be judged whether candidate segmentations were indicative of actual word combinations or merely errors resulting from the compound splitting algorithm.

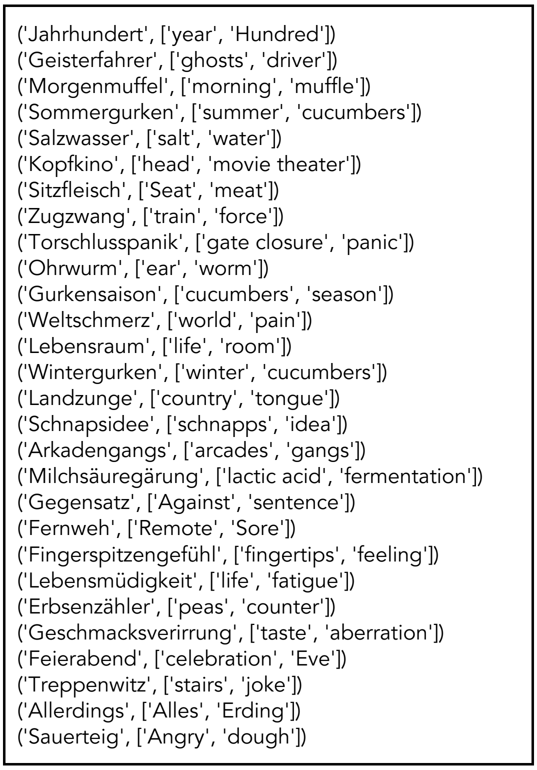
This important translation phase yields the segments which facilitate assessment of semantic reproducibility as modelled after the assumptions of Wierzbicka's concept of 'semantic primes.' Thus, we implement the ability to translate constituents into English as indication that the constituent concepts themselves are present in English and potentially re-composable into the source compound meaning. In other words, the translations here obtained begin the implementation of Assumption 4, fulfilled during later calculations.

An unfortunate weakness of this translation condition is an erroneous sensitivity to cognates, such as the German "Arm" which result in the same word form in an English translation. However, because the condition only excludes words for which both segments return a false translation value, and cases where a German compound is composed of two direct English cognates are extremely rare, this condition remains, overall, relatively unproblematic.

After these translation-based filtering conditions, we arrive at the final list of compound candidates. Subsequent evaluations do not produce any additional information regarding word-internal meanings or structure. Thus, this phase of the algorithm marks a shift from the incorporation of lexical to contextual meaning, both of which are ultimately juxtaposed in the final scoring. From Figure 7 we observe that the finalized list is relatively

[14] Google (2019).

cleaned of irrelevant segments, but still retains some problematic anomalies. In theory, the list should at this phase consist only of valid German compound nouns.



```
('Jahrhundert', ['year', 'Hundred'])
('Geisterfahrer', ['ghosts', 'driver'])
('Morgenmuffel', ['morning', 'muffle'])
('Sommergurken', ['summer', 'cucumbers'])
('Salzwasser', ['salt', 'water'])
('Kopfkino', ['head', 'movie theater'])
('Sitzfleisch', ['Seat', 'meat'])
('Zugzwang', ['train', 'force'])
('Torschlusspanik', ['gate closure', 'panic'])
('Ohrwurm', ['ear', 'worm'])
('Gurkensaison', ['cucumbers', 'season'])
('Weltschmerz', ['world', 'pain'])
('Lebensraum', ['life', 'room'])
('Wintergurken', ['winter', 'cucumbers'])
('Landzunge', ['country', 'tongue'])
('Schnapsidee', ['schnapps', 'idea'])
('Arkadengangs', ['arcades', 'gangs'])
('Milchsäuregärung', ['lactic acid', 'fermentation'])
('Gegensatz', ['Against', 'sentence'])
('Fernweh', ['Remote', 'Sore'])
('Fingerspitzengefühl', ['fingertips', 'feeling'])
('Lebensmüdigkeit', ['life', 'fatigue'])
('Erbsenzähler', ['peas', 'counter'])
('Geschmacksverirrung', ['taste', 'aberration'])
('Feierabend', ['celebration', 'Eve'])
('Treppenwitz', ['stairs', 'joke'])
('Allerdings', ['Alles', 'Erding'])
('Sauerteig', ['Angry', 'dough'])
```

Figure 7: The final list of compounds after constituent translation.

Corpus Segment Extraction

With the relevant compounds extracted, the next phase of the algorithm concerns scoring the resultant words based on their 'translatability.' As mentioned, this notion is rooted in Assumption 4, which regards translatability as the semantic distance between translated components of a source compound and translation attempts in the target language. This comparison is facilitated through the introduction of corpus translation segments.

Assumption 5 states that a source compound which yields consistent translation attempts in the target language is of high translatability. This important assumption balances scoring by rewarding words such as "Wissenschaft" or "Landzunge" with higher scores. Although the components of these words may not in themselves be sufficient to reconstitute the intended meaning of the word, there does not exist the semantic word gap which would render such words difficult-to-translate; a direct translation is easily found.

Based on Assumptions 4 and 5, the obvious method for assessing attempted translations of words in-context is integration of bilingual corpora. In our study, three German-English corpora were employed to provide ample translation data for implementing the assessment of semantic distance between source compound components and target translation attempts. By extracting and comparing target segments which attempt to translate the source compound, one may assess both the degree to which these attempts settle upon a 'stable' translation attempt (an indication that the selection of adequate translations is, in practice, unproblematic) and the degree to which these attempts can consummately capture the semantics of the compound as the sum of its constituents. This assessment procedes naturally from the first condition to the second, meaning that first the 'stability' of translation attempts is assessed and then these results are used to influence the semantic distance scoring.

Already one senses that the described implementation of the semantic concerns at stake during translation carries several limitations and reductions with respect to the matter's full scope, leaving the algorithm unduly sensitive to certain phenomena and insensitive to others which 'slip through the cracks.' One such case is words for which there is a high semantic distance (low numerical score) between source constituents and translation attempts, but a relatively stable translation attempt (high numerical score). Words of this pattern include "Augenblick" or "Landzunge," where the stable English equivalent is metaphorically linked to the consummate meaning of the German constituents. The algorithm is insensitive to such metaphorical relationships, and, as a result of the two-fold interpretation of translatability here implemented, struggles to properly handle words for which the stability and semantic distance scores are at odds, despite the intuitively high translatability of such words. The two-fold implementation again leads to erroneous calculations of the opposite case, where a low numerical stability is at odds with a high semantic distance score. Further problematic cases will be discussed during the relevant stages of the algorithm.

Corpora were selected based on availability and domain, with aims to include as much corpus data realistic for computations and as much domain variety as possible to increase the likelihood of finding sufficient segments for words which are used only in specialized and rare contexts. Thankfully, many German-English corpora are freely available online. All corpora were downloaded from the OPUS Corpus Collection.[15]

The three selected corpora were the Europarl Corpus[16], Wikipedia Corpus[17], and OpenSubtitles2018 Corpus[18]. Each corpus for this language pair contains at least one million sentence-aligned segments. The EuroParl Corpus contains European Parliament transcriptions and was chosen for its representation of formal, political language. The Wikipedia Corpus contains a subset of parallel sentences extracted from Wikipedia articles. This represents a more general language domain, but because of Wikipedia's structure is also influenced by the topical overlaps between the German and English subsections of Wikipedia. The OpenSubtitles2018 Corpus contains translated movie subtitles from the OpenSubtitles website.[19] The OpenSubtitles2018 Corpus is the largest of the three, containing around twenty-two million aligned German-English fragments. The Wikipedia Corpus contains around two million fragments and the EuroParl around one million. Of course, the exact choice and number of corpora used here is a matter open to criticism and represents one area of improvement for the algorithm.

Although these corpora can be queried online through the OPUS query interface, implementing a query through Python was deemed difficult and unnecessarily time-intensive given the number of queries necessary to extract all relevant segments for all

[15] Tiedemann (2012)
[16] Tiedemann (2012)
[17] Tiedemann (2012)
[18] Lison and Tiedemann (2016)
[19] www.opensubtitles.org

words. Given that the raw corpus data was freely downloadable, we opted to host the corpora locally and query them using Python scripts.

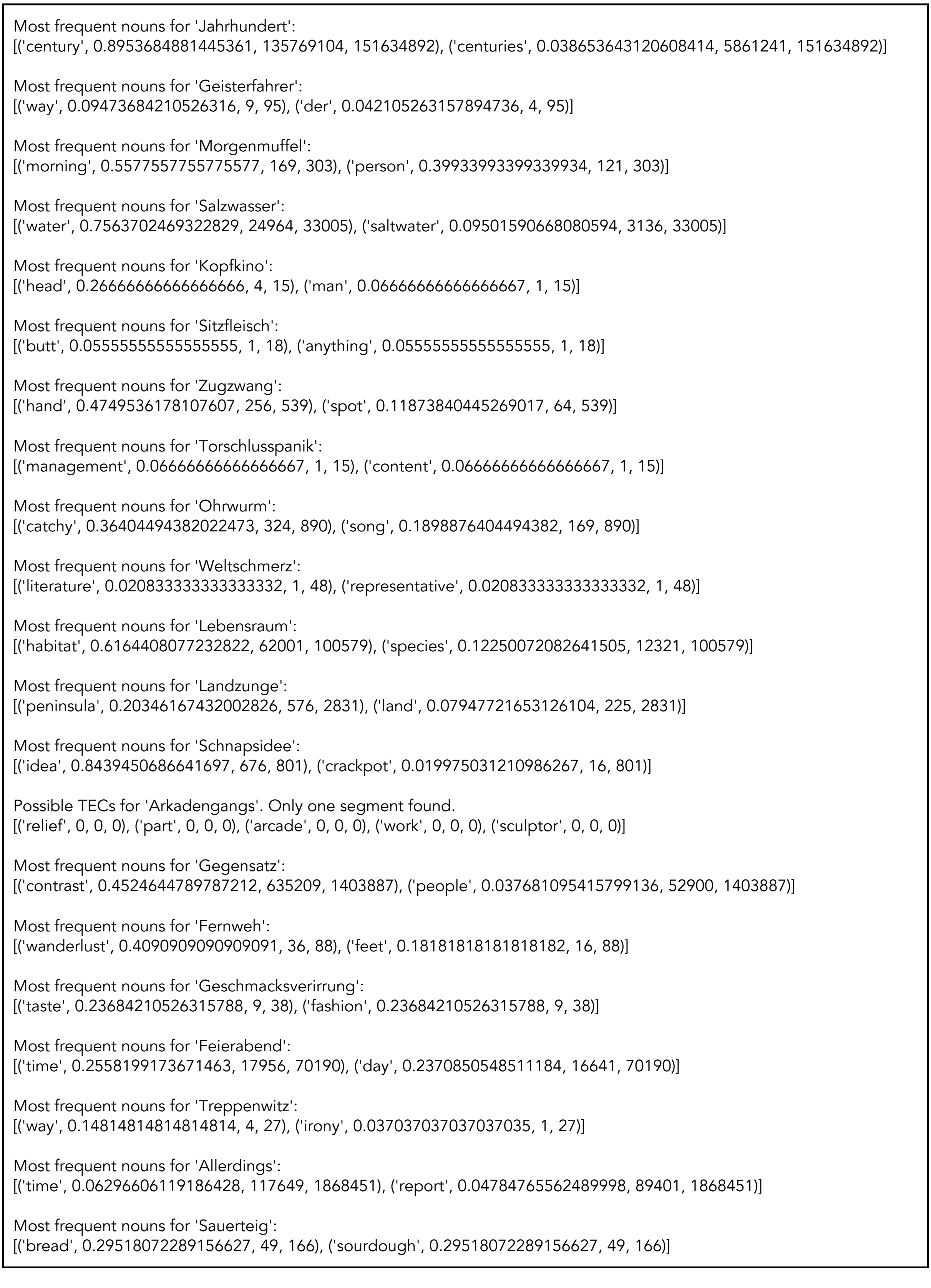
Corpus pre-processing was relatively uninvolved given that the corpora are provided in aligned TMX format. However, it was found that processing the relatively large XML documents using existing Python libraries posed a challenge to memory constraints. Thus, it was decided to convert the original documents to plain text files, which were more easily processed. TMX files were converted to plain text using Heartsome TMX Editor 8.

For each word in the resultant compound list shown in Figure 7, an iterator of each of the two bilingual-aligned corpus files was queried for the source word. The corresponding target segments were subsequently extracted into a separate document. After iterating through all words in the list for all three corpora, the algorithm produced a directory of files containing, for each word, all of the translation segments corresponding to that word.

Translation Equivalent Extraction and Translation Stability Scoring

The extracted segments were used to calculate the relative 'stability' of translation attempts based on the most frequent translation equivalence candidates (TECs) found in these segments. Inspired by the BASE algorithm employed by Tufiş et al[20] for the extraction of multilingual lexicons, our algorithm iterates, for each segment, across all other segments in a given word's segment corpus and identifies the set intersection of the nouns contained in that segment and the others. These noun intersections are then extracted into a list containing all TECs associated with the source word. We then calculate the frequency of each noun in the list relative to the size of its TEC list, extracting the two most-frequent candidates as the most-likely target translation equivalents. Figure 8 shows for each word the two most-frequent TECs, scored according to their relative frequency within the extracted TEC corpus for that word. This frequency score becomes the translation 'stability' score described in Assumption 5, with a more consistent translation hopefully correlated with a higher relative frequency and vice-versa.

[20] Tufiş et al (2004)



```
Most frequent nouns for 'Jahrhundert':
[('century', 0.8953684881445361, 135769104, 151634892), ('centuries', 0.038653643120608414, 5861241, 151634892)]

Most frequent nouns for 'Geisterfahrer':
[('way', 0.09473684210526316, 9, 95), ('der', 0.042105263157894736, 4, 95)]

Most frequent nouns for 'Morgenmuffel':
[('morning', 0.5577557755775577, 169, 303), ('person', 0.39933993399339934, 121, 303)]

Most frequent nouns for 'Salzwasser':
[('water', 0.7563702469322829, 24964, 33005), ('saltwater', 0.09501590668080594, 3136, 33005)]

Most frequent nouns for 'Kopfkino':
[('head', 0.26666666666666666, 4, 15), ('man', 0.06666666666666667, 1, 15)]

Most frequent nouns for 'Sitzfleisch':
[('butt', 0.05555555555555555, 1, 18), ('anything', 0.05555555555555555, 1, 18)]

Most frequent nouns for 'Zugzwang':
[('hand', 0.4749536178107607, 256, 539), ('spot', 0.11873840445269017, 64, 539)]

Most frequent nouns for 'Torschlusspanik':
[('management', 0.06666666666666667, 1, 15), ('content', 0.06666666666666667, 1, 15)]

Most frequent nouns for 'Ohrwurm':
[('catchy', 0.36404494382022473, 324, 890), ('song', 0.1898876404494382, 169, 890)]

Most frequent nouns for 'Weltschmerz':
[('literature', 0.020833333333333332, 1, 48), ('representative', 0.020833333333333332, 1, 48)]

Most frequent nouns for 'Lebensraum':
[('habitat', 0.6164408077232822, 62001, 100579), ('species', 0.12250072082641505, 12321, 100579)]

Most frequent nouns for 'Landzunge':
[('peninsula', 0.20346167432002826, 576, 2831), ('land', 0.07947721653126104, 225, 2831)]

Most frequent nouns for 'Schnapsidee':
[('idea', 0.8439450686641697, 676, 801), ('crackpot', 0.019975031210986267, 16, 801)]

Possible TECs for 'Arkadengangs'. Only one segment found.
[('relief', 0, 0, 0), ('part', 0, 0, 0), ('arcade', 0, 0, 0), ('work', 0, 0, 0), ('sculptor', 0, 0, 0)]

Most frequent nouns for 'Gegensatz':
[('contrast', 0.4524644789787212, 635209, 1403887), ('people', 0.037681095415799136, 52900, 1403887)]

Most frequent nouns for 'Fernweh':
[('wanderlust', 0.4090909090909091, 36, 88), ('feet', 0.18181818181818182, 16, 88)]

Most frequent nouns for 'Geschmacksverirrung':
[('taste', 0.23684210526315788, 9, 38), ('fashion', 0.23684210526315788, 9, 38)]

Most frequent nouns for 'Feierabend':
[('time', 0.2558199173671463, 17956, 70190), ('day', 0.2370850548511184, 16641, 70190)]

Most frequent nouns for 'Treppenwitz':
[('way', 0.14814814814814814, 4, 27), ('irony', 0.037037037037037035, 1, 27)]

Most frequent nouns for 'Allerdings':
[('time', 0.06296606119186428, 117649, 1868451), ('report', 0.04784765562489998, 89401, 1868451)]

Most frequent nouns for 'Sauerteig':
[('bread', 0.29518072289156627, 49, 166), ('sourdough', 0.29518072289156627, 49, 166)]
```

Figure 8: The most frequent TECs extracted for each word including relative frequencies.

One limitation of this extraction procedure relevant to our test case, seen in the case of "Salzwasser" is the extraction of nouns in isolation which may be together components of a compound corresponding to the German source. The algorithm may be improved by introducing regular expressions to capture full English equivalents of German compounds and not only isolated words.

As this phase of the algorithm was inspired by the work of Tufiş et al, it shares many of the assumptions and weaknesses inherent to the BASE algorithm. As mentioned, each segment produced only the set of nouns as TECs, which implements the assumption of Tufiş et al, relevant to our study, that translation equivalents must be of the same part-of-speech. As our study concerns only German noun compounds, it is reasonable to assume that most English correlates will also be nouns. However, this is a limitation of the algorithm as it is conceivable that a German noun corresponds more readily to a different part-of-speech in English, as is the case with the German "Schnapsidee," sometimes translated as a "crazy idea," indicating that the most effective method of encapsulating the first half of the German compound is through an English modifier. Noun identification was facilitated through the part-of-speech tagger of the Python Natural Language Toolkit[21] library.

This frequency-based algorithm is also sensitive to what Melamed[22] has termed 'indirect associations,' or cases where a word irrelevant to direct translation commonly co-occurs with translations, as is the case with collocations. Tufiş et al were able to combat these indirect associations in their study by imposing an 'occurrence threshold' which limited TEC extraction to only those candidates which occurred in the segment corpus more than three times. However, such a threshold was inappropriate for our study as many of the words we seek to assess are so rare that they only result in a single target segment for assessment. Thus, an occurrence threshold would have caused the algorithm to overlook too many relevant segments.

As many of the segments we wish to assess produce, due to their rarity, only a single target segment for evaluation, it was necessary to impose additional conditions to score words based on the number of segments that could be extracted for them. Words which produced no target segments were immediately removed from consideration as without translation attempts it is impossible to assess the semantic distance between the word-internal constituents and translations of the word in-usage. This represents a potential weakness of the algorithm, but is an unfortunate side effect of deliberately considering rare words. It is not possible, at this stage, to implement a representation of this rarity which would allow such words to be scored without compromising the conditions of scoring across all candidates or introducing an unnecessary conflation of rarity with untranslatability.

Words which produced only a single target segment were controlled to render their results comparable to those with multiple segments. From the set of nouns appearing in the target segment, a preliminary distance analysis was performed to select the two most likely TECs from the set of nouns, identified as the two TECs with the lowest semantic distance from the source constituent translations. However, as it is rare to find only a single target

[21] www.nltk.org

[22] Melamed (2001)

segment across more than twenty-five million aligned segments in multiple language domains, this condition represents a small catch for very rare exceptions.

Inspection of the resultant frequency scores suggests that the relative frequencies may be used to classify words into 'low' or 'high' stability and thereby exaggerate one condition of the bipartite scoring technique by rendering these continuous frequency scores into a binary classification. Comparison of several tests identified a relative frequency of 0.2 as sufficient threshold to split the words into those whose most frequent translation was relatively frequent and those which was rare and likely chance or indirect association. This threshold finalizes our implementation of the notion of translation 'stability' as the first step in the two-fold translatability scoring procedure. This threshold rewards words above 0.2 with a score of one while those below receive a zero stability score. Naturally, words associated with only a single segment yielded no relative frequency scores and thus were also classified with zero stability. Figure 9 shows the preliminary stability scores associated with each word in the list.

```
'Jahrhundert': 'score': [1]
'Geisterfahrer': 'score': [0]
'Morgenmuffel': 'score': [1]
'Salzwasser': 'score': [1]
'Kopfkino': 'score': [1]
'Sitzfleisch': 'score': [0]
'Zugzwang': 'score': [1]
'Torschlusspanik': 'score': [0]
'Ohrwurm': 'score': [1]
'Weltschmerz': 'score': [0]
'Lebensraum': 'score': [1]
'Landzunge': 'score': [1]
'Schnapsidee': 'score': [1]
'Arkadengangs': 'score': [0]
'Gegensatz': 'score': [1]
'Fernweh': 'score': [1]
'Geschmacksverirrung':  'score': [1]
'Feierabend': 'score': [1
'Treppenwitz': 'score': [0
'Allerdings': 'score': [0
'Sauerteig': 'score': [1]
```

Figure 9: Translation 'stability' scores based on relative frequency of TECs.

This binary scoring based on number of returned segments and relative frequency of TECs is an obvious reduction of the continuous notion of stability proposed earlier. An improvement to the algorithm at this stage may involve a more sophisticated integration of returned relative frequency scores to more-accurately represent the degree to which segments commonly contained a given TEC. In a sense, the decision to reduce scores to a binary is an unnecessary simplification intended to increase the score spread between words of high and low stability.

Distance Measure Scoring

The second scoring component is the semantic distance between extracted TECs and source compound constituent translations. The NLTK WordNet Interface[23] allows for semantic distance scoring based on WordNet[24] path distances. As mentioned, a high semantic distance is represented by a low numerical score, with entirely antithetical meanings producing a score of zero, and the same word sense resulting in a score of one when compared with itself. Three semantic distance scores were calculated for all permutations of source translations and target TECs. These permutation scores were averaged for each measure, and then integrated into the final translatability score for each word.

The chosen semantic distance scores were the "Path Similarity," which identifies the shortest path through the WordNet taxonomy connecting two word senses, the "Wu-Palmer Similarity," which returns a score based on the depth of the compared word senses in the taxonomy and their least common subsumer, and the "Lin Similarity," which scores based on the information content of the least common subsumer and the two compared synsets. Information content required to calculate the Lin Similarity is gathered from the SemCor semantically-annotated English corpus.[25] Full details on these similarity scores and their calculations can be found in the appendices. All scores return a value between zero and one.

As with the choice of bilingual corpora, the choice of distance measures represents a clearly debatable and somewhat arbitrary aspect of the algorithm. The algorithm may be improved through different combinations of distance measures or different semantically-annotated corpora used to gather information content.

Because a word can have any number of meanings, all contained within the "synset" used to calculate distance scores between all possible senses of the compared words, we decided to reduce this multifarious scoring to a single integer by simply selecting the highest distance score from among the possible synset combination scores. This simplification introduces the potential for error, as the sense in which a word is used in the source compound may be different from that which receives the highest distance score as compared with target translations.

These average distance measures implement the second phase of the two-fold translatability scoring procedure, namely calculating the degree to which TECs adequately capture the meaning of the source constituents in isolation. Clearly this implementation leaves several gaps by which additional meaning remains unconsidered. For example, there exists the potential that combinations of words systematically introduce a meaning that cannot be inferred from the constituents in isolation, but is nevertheless conventionally expected when such words are combined. Such patterns have been well-documented by Meibauer[26] and reveal a weakness of this algorithm. Overall, this phase of calculation is improved by more sophisticated manipulation of distance scores to more accurately encapsulate the meanings of source and target tokens and how these do or do not transfer the relevant semantic information during translation.

[23] www.nltk.org/howto/wordnet.html

[24] wordnet.princeton.edu

[25] Landes (1998)

[26] Meibauer (2013)

Translatability Scoring

After the average distance scores are obtained, these scores are in turn averaged with the binary stability classification to obtain the final translatability score. These scores along with further details of the results are exported to a separate directory. Figure 10 reproduces the final translatability scores associated with each word from Figure 1. Scores are sorted from highest to lowest, with the highest scores showing the words with the most consistent TECs and lowest semantic distance between source constituents and target TECs.

```
0.8275411579581397     Salzwasser
0.7071156613626228     Jahrhundert
0.4646577380952381     Arkadengangs
0.44820503717850413    Landzunge
0.40021340983894055    Lebensraum
0.35532614425937425    Gegensatz
0.301353143699554      Zugzwang
0.30065816247494626    Feierabend
0.2534613466598761     Schnapsidee
0.2534584945481294     Treppenwitz
0.1970239431014929     Weltschmerz
0.15333555979140517    Geisterfahrer
0.15002381089596242    Fernweh
0.09117387249626349    Torschlusspanik
0.06839805282082635    Ohrwurm
```

Figure 10: Final list of scored compounds.

Algorithm Summary and Reflection

Comments on Final Implementation

The algorithm outlined above is both an imperfect implementation of the notion of translatability originally intended and an idealized scoring procedure which nevertheless permits erroneous classifications running counter to the intentions of the algorithm. As implemented, our algorithm represents 'translatability' as the quadripartite average of translation stability and source-target semantic distances, which are, respectively, the binary scoring of most-frequent TEC relative frequencies relative to a threshold of 0.2, and a threefold set of semantic distance scores calculated across all permutations of source constituent translations and most-frequent target TECs. In short, translatability is a combination of the approximated degree to which the meaning contained in a source compound corresponds to frequent translations, and how consistent these translations are across various contexts. The final translatability score is a value between zero and one, with a high translatability indicating that a word is both consistently translated, and that its constituents correspond closely to this translation. Further details of algorithm successes and weaknesses will be discussed in conjunction with specific examples in the Evaluation section.

Diagram of Filtering and Scoring Procedure

Figure 11 shows how the original source text is successively filtered for relevant compounds and how these compounds are subsequently scored. At each phase of the algorithm, new resources are integrated to leverage their data towards accurate classification and scoring.

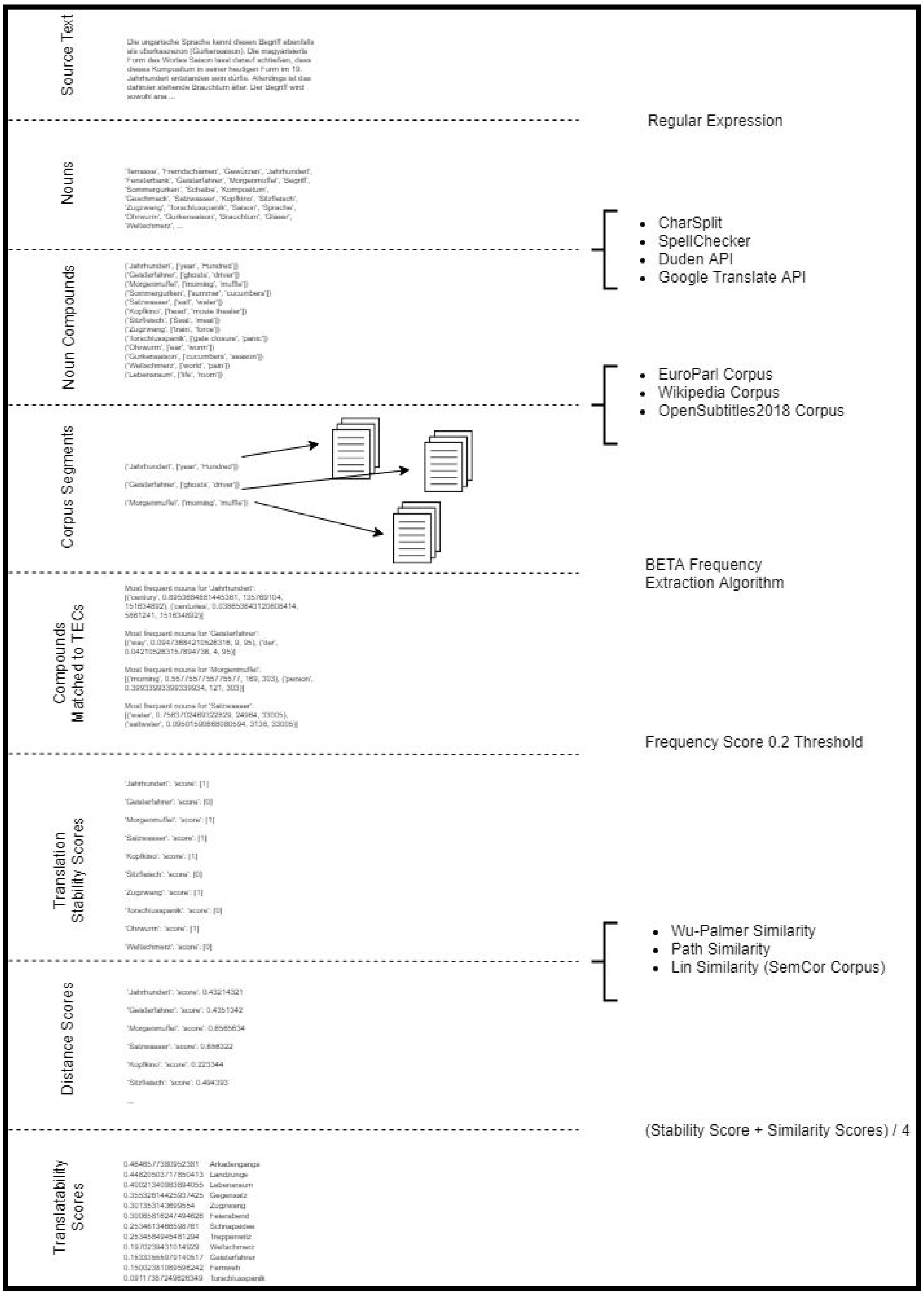

Source Text
Die ungarische Sprache kennt diesen Begriff ebenfalls als uborkaszezon (Gurkensaison). Die magyarisierte Form des Wortes Saison lässt darauf schließen, dass dieses Kompositum in seiner heutigen Form im 19. Jahrhundert entstanden sein dürfte. Allerdings ist das dahinter stehende Brauchtum älter. Der Begriff wird sowohl ans ...
Regular Expression
Nouns
'Terrasse', 'Fremdschämen', 'Gewürzen', 'Jahrhundert', 'Fensterbank', 'Geisterfahrer', 'Morgenmuffel', 'Begriff', 'Sommergurken', 'Scheibe', 'Kompositum', 'Geschmack', 'Salzwasser', 'Kopfkino', 'Sitzfleisch', 'Zugzwang', 'Torschlusspanik', 'Saison', 'Sprache', 'Ohrwurm', 'Gurkensaison', 'Brauchtum', 'Gläser', 'Weltschmerz', ...
CharSplit
SpellChecker
Duden API
Google Translate API
Noun Compounds
('Jahrhundert', ['year', 'Hundred'])
('Geisterfahrer', ['ghosts', 'driver'])
('Morgenmuffel', ['morning', 'muffle'])
('Sommergurken', ['summer', 'cucumbers'])
('Salzwasser', ['salt', 'water'])
('Kopfkino', ['head', 'movie theater'])
('Sitzfleisch', ['Seat', 'meat'])
('Zugzwang', ['train', 'force'])
('Torschlusspanik', ['gate closure', 'panic'])
('Ohrwurm', ['ear', 'worm'])
('Gurkensaison', ['cucumbers', 'season'])
('Weltschmerz', ['world', 'pain'])
('Lebensraum', ['life', 'room'])
EuroParl Corpus
Wikipedia Corpus
OpenSubtitles2018 Corpus
Corpus Segments
('Jahrhundert', ['year', 'Hundred'])
('Geisterfahrer', ['ghosts', 'driver'])
('Morgenmuffel', ['morning', 'muffle'])
BETA Frequency
Extraction Algorithm
Compounds
Matched to TECs
Most frequent nouns for 'Jahrhundert':
[('century', 0.8953884881446381, 135789104, 151634892), ('centuries', 0.03885384312080841, 5881241, 151634892)]
Most frequent nouns for 'Geisterfahrer':
[('way', 0.09473684210526316, 9, 95), ('der', 0.042105263157894736, 4, 95)]
Most frequent nouns for 'Morgenmuffel':
[('morning', 0.5577557755775577, 169, 303), ('person', 0.39933993399339934, 121, 303)]
Most frequent nouns for 'Salzwasser':
[('water', 0.7583702484932282, 24984, 33005), ('saltwater', 0.09501590668080594, 3136, 33005)]
Frequency Score 0.2 Threshold
Translation
Stability Scores
'Jahrhundert': 'score': [1]
'Geisterfahrer': 'score': [0]
'Morgenmuffel': 'score': [1]
'Salzwasser': 'score': [1]
'Kopfkino': 'score': [1]
'Sitzfleisch': 'score': [0]
'Zugzwang': 'score': [1]
'Torschlusspanik': 'score': [0]
'Ohrwurm': 'score': [1]
'Weltschmerz': 'score': [0]
Wu-Palmer Similarity
Path Similarity
Lin Similarity (SemCor Corpus)
Distance Scores
'Jahrhundert': 'score': 0.43214321
'Geisterfahrer': 'score': 0.4351342
'Morgenmuffel': 'score': 0.8585834
'Salzwasser': 'score': 0.858322
'Kopfkino': 'score': 0.223344
'Sitzfleisch': 'score': 0.494393
...
(Stability Score + Similarity Scores) / 4
Translatability
Scores
0.4848577380952381 Arkadengangs
0.44828583717850413 Landzunge
0.40021340983894055 Lebensraum
0.35532814425937425 Gegensatz
0.301353143895654 Zugzwang
0.30086818247494828 Feierabend
0.25346134885598781 Schnapsidee
0.25345849456481294 Treppenwitz
0.19702394310140929 Weltschmerz
0.15333565979140517 Geisterfahrer
0.15002381089596242 Fernweh
0.09117387249828349 Torschlusspanik

Figure 11: Summary of the algorithm's filtering and scoring sequence, with the integrated resources shown at right and status of the processed text at left.

Weaknesses, Potential Improvements

Given the above implementation, it remains unclear exactly what subclasses of words are scored highly or lowly according to our constructed translatability measure. It may be the case that these conditions identify patterns across lexemes and contexts which are in fact not representative of the notion of 'translatability' we intend, and instead describe co-occuring patterns such as word frequency or metaphorical meaning.

Overall, one of the algorithm's strengths is also a weakness; although we are able to filter and analyze many aspects of word meaning and distribution due to the variety and scope of leveraged resources such as translation API's, web dictionaries, and bilingual corpora, this resource variety also complicates the algorithm to the point of overfitting. It is difficult to imagine the applications of this algorithm outside of this specific language pair for this particular class of words. At the same time, the imposed complexity introduces several lacunae wherethrough undesired candidates spill. These gaps result from assumptions motivated by the practical demands of computation and simplification of the scoring problem. Several anomalous subsets of compounds with unique relationships between constituents and translations perform counter to the original intent of the study, and a crucial consideration for improving this algorithm would be to attempt to simplify and generalize the procedure that it may be applied to more language pairs.

Evaluation

Overview

Our evaluation is necessarily qualitative given the nature of our study. We examine three texts which demonstrate the algorithm's performance across a range of textual domains. All texts were gathered from Wikipedia articles and are included in the appendices. The first text regards a generic subject and also includes an appended list of words which we deemed potentially challenging to the algorithm. The second text is the longest of the three and contains specialized vocabulary. The third text describes a historical event and contains some unique words specific to the context. For each evaluation text we will provide an overview of the results of the algorithm, details on the computation, and discussion of the results.

Hardware Specifications

All computations were run on an Intel Core i5-3317U CPU at 1.70 GHz with 3.89 GB RAM.

Results

Text One: Generic Text with 'Untranslatable Words'[27]

```
(0.7139908155580019, 'Salzwasser', ['salt', 'water'],
[('water', 0.7563702469322829, 24964, 33005), ('saltwater', 0.09501590668080594, 3136, 33005)])

(0.7071156613626228, 'Jahrhundert', ['year', 'hundred'],
[('century', 0.8953684881445361, 135769104, 151634892), ('century', 0.038653643120608414, 5861241, 151634892)])

(0.6477580214413711, 'Geschmacksverirrung', ['taste', 'aberration'],
[('taste', 0.23684210526315788, 9, 38), ('fashion', 0.23684210526315788, 9, 38)])

(0.6117190378041573, 'Landzunge', ['country', 'tongue'],
[('peninsula', 0.20346167432002826, 576, 2831), ('land', 0.07947721653126104, 225, 2831)])

(0.5911525202302318, 'Feierabend', ['celebration', 'eve'],
[('time', 0.2558199173671463, 17956, 70190), ('day', 0.2370850548511184, 16641, 70190)])

(0.5623658388184533, 'Zugzwang', ['train', 'force'],
[('hand', 0.4749536178107607, 256, 539), ('spot', 0.11873840445269017, 64, 539)])

(0.5348783827145683, 'Kopfkino', ['head', 'movie theater'],
[('head', 0.26666666666666666, 4, 15), ('man', 0.06666666666666667, 1, 15)])

(0.5143744735214919, 'Morgenmuffel', ['morning', 'muffle'],
[('morning', 0.5577557755775577, 169, 303), ('person', 0.39933993399339934, 121, 303)])

(0.48495251174479115, 'Schnapsidee', ['schnapps', 'idea'],
[('idea', 0.8439450686641697, 676, 801), ('crackpot', 0.019975031210986267, 16, 801)])

(0.4646577380952381, 'Arkadengangs', ['arcade', 'gang'],
[('arcade', 0, 0, 0), ('arcade', 0, 0, 0)])

(0.44290117752851577, 'Sauerteig', ['angry', 'dough'],
[('bread', 0.29518072289156627, 49, 166), ('sourdough', 0.29518072289156627, 49, 166)])

(0.40021340983894055, 'Lebensraum', ['life', 'room'],
[('habitat', 0.6164408077232822, 62001, 100579), ('species', 0.12250072082641505, 12321, 100579)])

(0.36310165002317984, 'Gegensatz', ['against', 'sentence'],
[('contrast', 0.4524644789787212, 635209, 1403887), ('people', 0.037681095415799136, 52900, 1403887)])

(0.3590230528208263, 'Ohrwurm', ['ear', 'worm'],
[('catchy', 0.36404494382022473, 324, 890), ('song', 0.1898876404494382, 169, 890)])

(0.34590693026342884, 'Fernweh', ['remote', 'sore'],
[('wanderlust', 0.4090909090909091, 36, 88), ('foot', 0.18181818181818182, 16, 88)])

(0.27137596311982004, 'Weltschmerz', ['world', 'pain'],
[('representative', 0.020833333333333332, 1, 48), ('literature', 0.020833333333333332, 1, 48)])

(0.21941395626836635, 'Treppenwitz', ['stair', 'joke'],
[('way', 0.14814814814814814, 4, 27), ('newspaper', 0.037037037037037035, 1, 27)])

(0.19887857663532663, 'Sitzfleisch', ['seat', 'meat'],
[('butt', 0.05555555555555555, 1, 18), ('anything', 0.05555555555555555, 1, 18)])
```

[27] Sauregurkenzeit. Wikipedia.

```
(0.15333555979140517, 'Geisterfahrer', ['ghost', 'driver'],
[('way', 0.09473684210526316, 9, 95), ('der', 0.042105263157894736, 4, 95)])

(0.02535439481990153, 'Torschlusspanik', ['gate closure', 'panic'],
[('strategy', 0.06666666666666667, 1, 15), ("'end-of-term-itis", 0.06666666666666667, 1, 15)])

(0.0, 'Allerdings', ['alles', 'erding'],
[('time', 0.06296606119186428, 117649, 1868451), ('report', 0.04784765562489998, 89401, 1868451)])
```

Computation

The computation lasted 00:44:34.

Discussion

As discussed throughout the Methodology section, these results are a promising indication that the scoring algorithm can somewhat capture intuitions about the relative translatability of German compounds. Furthermore, the relatively simple TEC extraction algorithm seems in many cases to extract relevant translations. Of admirable note is "Ohrwurm." However, it is clear from the demonstrated scores as well as the inclusion of certain words that the algorithm is open to considerable error.

Noting the most highly scored words, including "Salzwasser," "Jahrhundert," and "Landzunge," we note that the corresponding scores, although correct in identifying the words as unproblematic for translation, do not capture our intuitions about the degree of translatability we might associate with these words. Few would argue that one word is necessarily 'more translatable' than the others, and the scores, all below 0.7, would seem to indicate that these words are well-below an ideal of translatability.

There exist two obvious interpretations for these results. Firstly, because it is arguably the case that an ideal of 'perfect translatability,' assumed to be represented by a score of one, would only exist for translations which yield a semantic distance of zero, our cross-linguistic comparisons must, implicitly, always result in scores well-below one as a translation can, according to popular opinion, 'never fully translate' the meaning of the source word.

The second, more likely interpretation is that the intuitively low scores reflect undesirable noise introduced during the distance measuring phrase, for which the complicated sequence of synset comparisons and averagings likely incorporates values irrelevant to the comparisons we intend. The 0.7 'cap' is then a side-effect of the averaging procedure, intended originally to remove noise from unwanted word-sense comparisons. In theory, the averaging both normalizes the results and simultaneously renders them unintuitive. If we take this second interpretation, the algorithm may be modified to again normalize final scores according to this perceived threshold, theoretically re-aligning the values according to a more intuitive score range.

The low scores may also be suboptimal comparisons resulting from the structure of the data compared, suggested in the case of "Jahrhundert," where a distance comparison of "year" and "hundred" with "century," although not producing a wholly inaccurate score, is not the ideal comparison one imagines for assessing the distance between the source and target languages. More optimally, the German compound may be translated as a whole, with this token being weighed against most frequent TECs.

Further exploration of the results shows some misidentifications and miscalculations which are continued 'bugs' of the algorithm. The non-compound "allerdings," which was erroneously identified as a noun because capitalized at the start of a sentence, was continually able to surpass compound filtering because its segments, as split by the CharSplit algorithm, are both a valid German noun ("das All") and a proper noun (Erding is a town in Southern Bavaria). Filtering segments for proper nouns would conceivably remedy this type of error, potentially implemented through a dictionary of German proper nouns. Nevertheless, this adverbial is never matched to noun segments because no common TEC can be expected from the set of nouns. Thus, the word is scored extremely low by the algorithm, which is, in some sense, an indication of its misidentification.

When inspecting the TECs resulting from queries of "Schnapsidee," we find that the identified tokens together seem to well-encapsulate the meaning of the German original, a "crackpot idea." However, because these translations, particularly "crackpot" are semantically distant from the german constituent correlating to the same aspect of the compound meaning (Schnapps), the word ultimately receives a relatively low score, although it seems that a stable translation exists. Here again, we find that metaphorical or narrative meanings, such as "an idea one has whilst drinking Schnaps," cannot be assessed by the algorithm, which blindly compares denotational meanings. It is difficult to propose an implementation which could compare such connotational or metaphorical meanings, and perhaps the algorithm as it stands correctly handles such cases as they represent, in a sense, the essence of what makes some translations difficult.

"Sauerteig" is incorrectly scored because the initial segment is erroneously translated to an unintended meaning. This type of error might be overcome by accounting for multiple meanings and correlating this with the WordNet synset results, or perhaps complicating the translation phase to more actively identify the most likely translation.

In all, the results from this text indicate that the algorithm identifies and splits compounds effectively, but that scores do not perfectly align with our intuitions regarding the notion of translatability proposed. The algorithm seems to have appropriately handled the 'untranslatable' words artificially included in the text, but it is unclear whether scoring is an accurate reflection of the notion we intend to assess.

Text Two: Biographical Text with Specialized Vocabulary[28]

```
(0.7413132625081347, 'Blickwinkel', ['view', 'corner'],
[('view', 0.3082882446667755, 54756, 177613), ('point', 0.201117035352142, 35721, 177613)])

(0.7340971560964124, 'Lebensbejahung', ['life', 'affirmation'],
[('life', 0.3076923076923077, 4, 13), ('affirmation', 0.3076923076923077, 4, 13)])

(0.7192203705505291, 'Weltbild', ['world', 'image'],
[('world', 0.5149495334222053, 2704, 5251), ('view', 0.1601599695296134, 841, 5251)])

(0.6922485344884015, 'Kunstwerk', ['art', 'plant'],
[('art', 0.7170157551800879, 446224, 622335), ('work', 0.12063599186933083, 75076, 622335)])

(0.6029826251712376, 'Hauptwerk', ['head', 'plant'],
[('work', 0.6161766295535618, 22801, 37004), ('work', 0.08174791914387634, 3025, 37004)])

(0.5321632119390639, 'Fortschritt', ['fort', 'step'],
[('progress', 0.9070562682743314, 216472369, 238653738), ('report', 0.007991330938214762, 1907161, 238653738)])

(0.5270511268210136, 'Vielzahl', ['a lot of', 'number'],
[('number', 0.40450872838333385, 356409, 881091), ('variety', 0.1673561527696912, 147456, 881091)])

(0.4922987820685692, 'Grundlage', ['reason', 'location'],
[('basis', 0.7720602140649508, 72216004, 93536751), ('report', 0.01729784264155166, 1617984, 93536751)])

(0.48681804584305755, 'Moralphilosophie', ['moral', 'philosophy'],
[('philosophy', 0.6030769230769231, 196, 325), ('professor', 0.04923076923076923, 16, 325)])

(0.4692203705505291, 'Weltbildes', ['world', 'image'],
[('view', 0.06060606060606061, 4, 66), ('world', 0.06060606060606061, 4, 66)])

(0.4332834727876646, 'Wissenschaft', ['knowledge', 'shaft'], [('science', 0.552013090173098, 18181696, 32937074),
('scientist', 0.18628032957633092, 6135529, 32937074)])

(0.4210754396499743, 'Aufzeichnungen', ['on', 'drawing'],
[('record', 0.7984182302212136, 688900, 862831), ('record', 0.10292166136821695, 88804, 862831)])

(0.4160518698634438, 'Herrschaft', ['sir', 'shaft'],
[('rule', 0.3792556393429984, 767376, 2023374), ('reign', 0.14305017263244463, 289444, 2023374)])

(0.41128287406314734, 'Gesellschaft', ['fellow', 'shaft'],
[('society', 0.7726805506838174, 130919364, 169435304), ('company', 0.09580554121117521, 16232841, 169435304)])

(0.4041928568720496, 'Herrenmensch', ["men's", 'human'],
[('race', 0.2647058823529412, 9, 34), ('master', 0.2647058823529412, 9, 34)])

(0.3958362740515719, 'Aufgabe', ['on', 'gift'],
[('task', 0.45722672045039425, 32684089, 71483331), ('job', 0.14495632555231652, 10361961, 71483331)])

(0.3930012884499046, 'Menschheit', ['human', 'ness'],
[('humanity', 0.5052725669258218, 2114116, 4184110), ('mankind', 0.282914168126555, 1183744, 4184110)])

(0.3923090021760258, 'Zusammenspiel', ['together', 'game'],
[('interaction', 0.2935919957229054, 3844, 13093), ('interplay', 0.12220270373482013, 1600, 13093)])
```

[28] Friedrich Nietzsche. Wikipedia.

```
(0.3665394273829911, 'Mittelpunkt', ['medium', 'point'],
[('heart', 0.16241350708853072, 263169, 1620364), ('centre', 0.1294548632282623, 209764, 1620364)])

(0.36310165002317984, 'Gegensatz', ['against', 'sentence'],
[('contrast', 0.4524644789787212, 635209, 1403887), ('people', 0.037681095415799136, 52900, 1403887)])

(0.36159267685627344, 'Einheit', ['on', 'ness'],
[('unit', 0.38954650334672725, 5929225, 15220840), ('unit', 0.36560932248154504, 5564881, 15220840)])

(0.3567321469178893, 'Erkenntnisentwicklung', ['knowledge', 'development'],
[('progress', 0, 0, 0), ('progress', 0, 0, 0)])

(0.31872682650250705, 'Widerstreit', ['contrary', 'dispute'],
[('conflict', 0.14578587699316628, 64, 439), ('time', 0.08200455580865604, 36, 439)])

(0.31613674315880197, 'Methode', ['meth', 'ode'],
[('method', 0.4452925482314409, 8392609, 18847405), ('method', 0.40651877539640074, 7661824, 18847405)])

(0.30489912301815153, 'Einfluss', ['on', 'river'], [('influence', 0.871429369818835, 19740249, 22652724), ('impact',
0.024173693194690406, 547600, 22652724)])

(0.29145894173938214, 'Weltanschauung', ['world', 'view'],
[('religion', 0.1678435474965856, 5776, 34413), ('age', 0.12277337052857931, 4225, 34413)])

(0.2833333333333333, 'Wolfgang', ['wolf', 'gear'],
[('*', 0.3264930888108498, 55225, 169146), ('der', 0.14203705674387807, 24025, 169146)])

(0.2719988489799808, 'Machterweiterung', ['makes', 'extension'],
[('power', 0.125, 4, 32), ('system', 0.125, 4, 32)])

(0.194897610080579, 'Herrschafts', ['sir', 'shaft'],
[('territory', 0.08440717676798816, 1369, 16219), ('part', 0.06313582834946667, 1024, 16219)])

(0.16973251539460066, 'Widersinn', ['contrary', 'sense'],
[('paradox', 0.09191176470588236, 25, 272), ('absurdity', 0.09191176470588236, 25, 272)])

(0.029981323320508676, 'Seienden', ['be', 'the'],
 [('everything', 0.03333333333333333, 4, 120), ('reality', 0.03333333333333333, 4, 120)])
```

Computation

The computation lasted 01:34:22.

Discussion

The results of this longer text show in greater detail the errors our algorithm is prone to make and the types of words that receive high and low scores. Again the maximum score caps at around 0.7, suggesting that this is a ceiling artificially imposed by the conditions of the calculation. The example of "Lebensbejahung" essentially confirms this interpretation, given that the source translations and target TECs are identical and theoretically result in a perfect score by our. Thus, our calculations clearly require normalization or restructuring.

From this larger pool of words we can see in more detail the types of tokens that 'slip through the cracks' of the algorithm's successive layers. Notably, the proper noun "Wolfgang," is identified as a compound. Again, filtering for proper nouns would avoid this issue. This word also falsely passes the high translation stability threshold of 0.2, as this proper noun is indirectly associated with the formatting symbol "*", used in the OpenSubtitles2018 corpus to identify character names in movie scripts. Again, indirect associations displace the scores. This problem may be remedied by increased attention to TEC filtering phases. Somehow this character has been classified as a noun, which is an unfortunate error of the part-of-speech tagger remedied by further filtering.

In the results we find several examples of substantives ending with the morpheme "-schaft," which are not true noun compounds although their meaning is also compositional in nature. The translations corresponding to this second constituent reveal the classification error, as this morpheme cannot be translated as a full word, with the incorrect translation "shaft" disrupting the intended operation of the algorithm. This translation error, setting aside the fact that the words are technically irrelevant to our study of noun compounds, accounts for the relatively low scores of these words, which when treated as true compounds do not yield segments which compositionally align with target TECs. A similar case is found with the words "Menschheit" and "Einheit," which when falsely considered among the true compounds cannot yield source translations which expectedly align with the target TECs, and although, interestingly, the GoogleTranslate API has accurately translated the German suffix "-heit" to its closest English equivalent, this correlation is unconsidered by the WordNet distance measures which are indifferent to morphemes.

Additional non-compounds have also eluded the algorithm due to errors similar to those found to result from Text One. "Einfluss" and "Aufgabe" both should have been excluded during the compound splitting phase given their initial constituents are prepositions rather than nouns. This error results from the leniency of the constituent assessment condition, which in order to capture as many potential segments as possible, combined the Duden API, which rules out non-nouns, with the SpellCheck library, which permits them. Thus, the prepositions are permitted and later not ruled out at the translation phase as they indeed produce valid English translations. This type of error also accounts for the presence of "Seienden," whose second constituent translates as a German article. This type of error can be remedied by imposing an additional condition on constituent translations requiring that they be nouns, otherwise immediately excluding words such as these for which a proposed constituent is not a noun but a morpheme.

Strangely, "Methode" has been classified as a noun compound because the proposed split produces two valid nouns ("das Meth" and "die Ode") which have nothing to do with the word's meaning. Although the correct translation equivalent is identified through the frequency analysis, the assumedly large semantic distance between the falsely-split segment translations and the TECs 'overpowers' the stability score of the word, resulting in a low translatability. Further refinement of the scoring procedure in order to rebalance the degree to which the respective scores may influence each other would be a welcome improvement to the algorithm.

When erroneously considered non-compounds are removed from the list one finds that the scoring again roughly confirms intuitions regarding the translatability of segments, revealing that one of the major problems with the algorithm, at this stage, remains the correct filtering of irrelevant non-compounds.

Finding words with relatively clear meanings such as "Blickwinkel," "Weltbild," and "Kunstwerk" at the top of the list whilst more complex or polysemous terms such as "Widersinn," "Weltanschauung," and "Zusammenspiel" cluster towards the bottom forms a promising indication that the scoring algorithm is in some way able to capture the notion of translatability we sought to implement. Still, small errors prevent the results from being a wholly accurate representation. For example, the initial segment of "Machterweiterung" has been translated as the verb "makes" instead of the correct noun "power," which would have improved the distance score given the most common TEC. Again, the translation phase of the algorithm deserves additional improvement given the degree of importance translations hold for accurate calculations later in the algorithm.

The German "Zusammenspiel," which can be translated as "interaction," "interplay," or "cooperation," reveals an unfortunate side-effect of Assumption 6, which stipulates that meaning in the source language be controlled in order to evaluate inconsistency in the target language as an independent measure of 'stability'. However, this case clearly reveals a polysemous compound which, due to its varied meanings in German, disrupts the assumed source meaning control we assumedly take advantage of in our calculations. What the low score of this word more likely indicates is the polysemy of this word reflected in the degree of difference between the source constituent translations and the varied usages of the word in different contexts. Thus, from this case we may infer that the algorithm is unduly sensitive to cases of noun compound polysemy, conflating these cases with the notion of translatability we attempted to isolate. This error might be overcome by expanding the original noun list to account for variations in word sense rather than treat every compound as a monosemic entity, but it is difficult to conceive of a practical implementation which could both accurately assess these differences of meaning and thereupon proceed through the rest of the algorithm without imposing generic calculations upon all compounds of the same lexical form.

The compounds "Weltbild" and "Weltanschauung," appearing at opposite ends of the score spectrum for this text, pose an interesting dialectic for assessing the relative accuracy of the translatability score. From the source constituent translations alone we may note that, as sums of parts, these compounds should theoretically correspond to approximately the same English meaning, yet our scoring indicates that the German "Weltanschauung" is a more difficult to translate concept. Furthermore, a religious connotation is indicated by the most frequent TEC, suggesting an additional layer of meaning not found in the TECs for "Weltbild," which correspond very closely to the source constituents. Our intuitions here, as German non-native speakers, limit us to speculation regarding the precise connotational differences between these two words, but the comparison alone poses an interesting outcome of the scoring algorithm.

The results of this second text further indicate the algorithm's inability to properly rule out irrelevant candidates, while the importance of accurate translations emerges as a key issue demanding further attention to the relevant phase. Furthermore, restructuring of the scoring phase may rebalance the degree to which stability scores or similarity scores may negatively influence the other, while the 0.7 score cap clearly emerges as an artificial ceiling requesting normalization.

Text Three: Historical Text with Culture-Specific Vocabulary[29]

```
(0.7318679503263511, 'Gesetzesentwurf', ['law', 'draft'],
[('bill', 0.4122546393258974, 8464, 20531), ('draft', 0.2595587160878671, 5329, 20531)])

(0.6010621950144079, 'Bundesregierung', ['federal', 'government'],
[('government', 0.8863212773724966, 138384, 156133), ('state', 0.012969711720136038, 2025, 156133)])

(0.5506881873356225, 'Ermittlungsverfahren', ['discovery', 'method'],
[('investigation', 0.5712868832134887, 2304, 4033), ('investigation', 0.0485990577733697, 196, 4033)])

(0.5502042546238433, 'Deutschland', ['german', 'country'],
[('district', 0.2228272438040813, 2748964, 12336750), ('municipality', 0.11966076965165055, 1476225, 12336750)])

(0.4889682110771704, 'Beitrag', ['at', 'support'],
[('contribution', 0.8026273811246694, 44756100, 55761990), ('report', 0.019508790844803062, 1087849, 55761990)])

(0.47635568032209774, 'Sendereihe', ['send', 'line'],
[('series', 0.3058103975535168, 100, 327), ('der', 0.04892966360856269, 16, 327)])

(0.4236481324580713, 'Staatsanwaltschaft', ['state', 'legal profession'],
[('office', 0.4358793213272338, 150544, 345380), ('prosecution', 0.165385951705368, 57121, 345380)])

(0.4159454839674984, 'Vortrag', ['in front', 'support'],
[('lecture', 0.7172104872094257, 168921, 235525), ('speech', 0.0695637405795563, 16384, 235525)])

(0.4075970670681618, 'Bundestag', ['federal', 'day'],
[('member', 0.5001915892326851, 31329, 62634), ('election', 0.08980745282115145, 5625, 62634)])

(0.4046689723320158, 'Untertitel', ['under', 'title'],
[('subtitles', 0.9378403336289468, 1229881, 1311397), ('subtitle', 0.021266633978879013, 27889, 1311397)])

(0.13990522225706287, 'Fernsehbeitrag', ['tv', 'contribution'],
[('programme', 0.16666666666666666, 4, 24), ('broadcast', 0.041666666666666664, 1, 24)])

(0.1392903766089428, 'Paragrafen', ['para', 'count'],
[('paragraph', 0.1856060606060606, 49, 264), ('law', 0.0946969696969697, 25, 264)])

(0.05681870213120213, 'Bundeskanzlerin', ['federal', 'chancellor'],
[('time', 0.034017430418892325, 121, 3557), ('today', 0.028113578858588697, 100, 3557)])

(0.017613636363636366, 'Eingeleitet', ['on', 'guided'],
[('au', 0.14754098360655737, 9, 61), ('dem', 0.14754098360655737, 9, 61)])
```

Computation

The computation lasted 00:54:26.

Discussion

This final assessment includes specialized terms for which English translation equivalents are not expected given that the concepts are specific to the German government and legal system. Overall the results suggest no intuitive patterns regarding the translatability of the terms. Seemingly straightforward translations such as "Untertitel" receive low scores and otherwise culturally-specific proper nouns, such as "Bundesregierung," receive high scores.

[29] Böhmermann-Affäre. Wikipedia.

Admittedly, we do not intend to evaluate proper nouns as their translatability is already expected to be low or impossible, but their inclusion here affords the chance to confirm this assumption. Some of these puzzling results may be attributed to the algorithm bugs alluded to earlier while others represent additional weaknesses presented by the highly-specialized terms. Such irregular results are somewhat expected given that the algorithm was never designed to accommodate such specialized vocabulary which defy many assumptions of the translation model upon which it is based.

Again, several non-compounds have been included as a result of unfortunate coincidences which split nouns into translatable segments. "Paragrafen," "Vortrag," and "Beitrag" all exhibit this error, again showing that additional filtering for affix segments will remove these irrelevant considerations. "Eingeleitet" also exhibits this error, but is, like "allerdings" incorrectly included because of a sentence-initial capitalization.

With these errors removed, we find many of the words to be intuitively misscored. "Deutschland" seemingly corresponds to one obvious translation, but the correct TEC was not identified through the segment frequency analysis, possibly because this contextual information is untranslated in many appearances. As a result of this translation segment omission, the TECs appear to be indirect associations, which, combined with the pseudo-metaphorical segment translation problem experienced with "Landzunge" and "Jahrhundert," further decreases the translatability score.

Again, decreasing the degree to which semantic distance scoring may punish a word could improve these results, It is also conceivable to integrate a method for distinguishing words of metaphorical composition as a distinct subclass of compounds, but it is difficult to propose a reliable method for distinguishing compounds of compositional meaning (Orangensaft) from lexical (Deutschland) or metaphorical meaning (Landzunge). The results of "Ermittlungsverfahren" also suggest the influence of this type of problem. In a sense, the algorithm is already sensitive to these distinctions through the bipartite structure of the scoring phase, and it is unfortunate that the TEC extraction has in this case failed to produce the correct segments necessary for an adequate scoring.

Several domain-specific terms seem to have posed a challenge to the algorithm, perhaps again due to metaphorical compositions or indirect associations. "Bundestag," "Bundesregierung," "Staatsanwaltschaft," and "Bundeskanzlerin" have all been correctly split and constituents are appropriately translated, but because of indirect association errors the scores show a greater spread than expected.

When comparing the source translations and extracted TECs and keeping the scoring mechanism in mind, "Gesetzesentwurf" and "Bundesregierung" perform exactly as expected in receiving relatively high scores. It is a matter of debate whether these terms, notably "Bundesregierung" can actually be well-translated into English, as their meaning is largely tied to specific objects within the context of the German governmental system.

The scores of "Bundestag," "Bundeskanzlerin," and "Staatsanwaltschaft" reveal that the problem of indirect association may be more present with certain specialized terms as the scores have clearly been negatively affected by the inadequately-identified TECs. Most likely, as is the case with "Bundeskanzlerin," certain translation conventions propagate the error, as although this formal title is mentioned in German source texts, it is, for ease of translation, simply omitted in many target segments. In a sense, this practice is reflected in the results, but is not the notion of translatability we intended to identify and is again a conflated circumstance. Thus, we find that translation practices producing texts with word-distributions similar to those we target for our intended notion of low-translatability also confuse such 'irregular translations' with those we intend to punish. It is open for debate whether the title of "Bundeskanzlerin" can in fact be appropriately translated into English without further explanation, but ultimately this low score does not reflect our initial intent and is thus an area for improvement. It is, however, difficult to imagine a method of distinguishing such circumstances from our intended concept of low-translatability, given that both textual representations are so similar.

Several television-related words have received similarly low scores, and it is again unclear whether these should be interpreted as miscalculations or reflections of the institutionally-specific meanings of these terms, given that they apply to the German television system. "Untertitel" is clearly misscored as a result of the same WordNet morpheme indifference which produced low scores for words like "Einheit." In all other aspects, the algorithm seems to have performed according to expectations and it is likely the case that the WordNet distance scores were unable to correctly associate the English prefix "sub-" with the German preposition-derived prefix "unter-". This 'more sophisticated' form of semantic assessment, where a German segment is consistently translated into a segment inaccessible to the distance scoring procedures, may be remedied through integration of cross-language morpheme correspondences.

"Fernsehbeitrag" and "Sendereihe" again pose a somewhat interesting pair of terms within a relatively small semantic space which nevertheless receive greatly different scores. Perhaps the notion of a German Sendereihe is more readily-translatable into English while a Fernsehbeitrag is a more specific notion which lacks direct analogies in English television culture. Further investigation of such cases would be facilitated by posing the algorithm with additional, similar dyads, such as the pair of "Weltbild" and "Weltanschauung."

Overall, close investigation of each case reveals clear explanations suggesting the algorithm performs as expected but it regularly challenged by unique word structures or translation conventions. It remains, however, unclear whether many scores are in fact reflections of the condition we intend, or simply conflations with other patterns in word form or translation practice.

Conclusion

Discussion

From these evaluation texts a number of regular bugs appear which suggest clear areas of improvement for the algorithm. Most errors concern compound classification, as the ability of the algorithm to correctly filter German words according to our intended word class of study continues to fail with undesired tokens. Another notable area of improvement is the translation of source constituents into English, which when producing inappropriate translations has the potential to dismantle later calculations. Greater flexibility during translation allowing the algorithm to better select ideal translations will improve this important stage.

For the first two evaluation texts, scoring based on categorized translation 'stability' and semantic distance seems to capture some sense of translatability we intended.

However, certain groups of words which regularly puzzle the algorithm also emerge, particularly those for which metaphorical meanings contribute to the compositional meaning of compound constituents. Further adjustments to the scoring procedure and perhaps more sophisticated statistical calculations may improve the perceived accuracy of results.

Overall, this relatively simple scoring algorithm represents a promising first step towards the ideal of translatability scoring we aimed for with our study. Although still containing many bugs, the algorithm suggests the possibility of improvement and future application.

Applications

As discussed, the imagined applications of such an algorithm lie in the domains of semi-automated translation and OOV word identification, wherein such a procedure may be used to identify for a given translation those words for which additional context or explanation is useful. This algorithm also promises the automation of OOV word identification, thus facilitating further research into machine translation methods for accurately translating such difficult words. Recently our algorithm has been applied by Pömsl[30] to verify the relative translatability of a list of words used in such a study.

Limitations, Improvements

Although the algorithm itself promises several applications and improvements, the study as a whole suffers from several limitations and inadequacies which render our results open to improved replication and question.

Several assumptions imposed by our somewhat naïve conception of translatability guided the algorithm's design and thereby introduced complications and room for error that might have been avoided by a more rigorous investigation and modelling of the semantic concerns at stake during translation. Further investigation into the semantic models behind translation might yield insights towards advantageous restructurings of the algorithm.

As implemented, the algorithm is relatively unsophisticated, assuming a decision tree model for word classification and employing relatively unsophisticated calculations for manipulating TEC frequency scores and semantic distance measures. The scoring procedure may ultimately be improved through implementation of more specialized categorization methods and more subtle mathematical analyses in order to capture in greater detail the features of the words we intend to score as well as the statistical patterns behind the notion of translatability we aim to calculate.

In another sense, the algorithm is overly complex through its varied and somewhat inconsistent application of translations, corpus data, and distance scores. Many of these integrations are questionably arbitrary and thus open to redistribution or supplementation. Furthermore, the sources as leveraged seem to overfit the procedure as the algorithm is seemingly unable to score words outside of the class of noun compounds, ultimately limiting the application of the procedure. Further investigations to generalize the applicability of the algorithm are welcome, with the ultimate goal of producing a

[30] Citation Missing

translatability scoring procedure capable of assessing multiple language pairs for multiple classes of words.

Our evaluation procedures, as entirely qualitative assessments, are limited by the assumptions of subjective interpretation and our inadequate understanding of German connotations as non-native speakers. Guided by the notion of perceived accuracy compared against a relatively inchoate and admittedly invented translatability metric, our assessments are limited to speculations based on intuition. Nevertheless, we are confident in concluding that translation-related patterns do emerge from the results, and additional evaluation will better confirm or deny our suspicions.

Regarding these further evaluations, it would be optimal to integrate some form of quantitative assessment to complement our interpretive discussion with more objectively-produced analyses. One potential methodology for a quantitative assessment would be a native-speaker-verified list of words scored according to perceived translatability and compared against the algorithm's scores for the same set of words. Through machine learning methods, this kind of quantitative testing may also provide an effective means of tuning the algorithm's parameters.

One interesting and important set of words identified through our qualitative analysis are those pairs which although seemingly similar in meaning when compared based on constituent translations alone nevertheless pattern differently as full compounds. It will be prudent to gather and test further pairs of words such as "Weltbild" and "Weltanschauung," as scores may in the very least confirm or disconfirm the ability of the algorithm to distinguish between high and low translatability.

As it currently stands, our study suggests that it is possible to capture and assess relationships between word meaning and translation attempts as a rudimentary, algorithmic approximation of the nebulous concept of 'translatability.'

Appendices

Paper and Resource Citations

Barrus, Tyler (2019). pyspellchecker 0.5.0. Retrieved from https://pypi.org/project/pyspellchecker/

Bosák, Radomír (2018). Duden. Retrieved from https://github.com/radomirbosak/duden

Cliff Goddard, Anna Wierzbicka (2014). Words and Meanings: Lexical Semantics across Domains, Languages and Cultures. Oxford: Oxford University Press.

Duden (1996). Die deutsche Rechtschreibung, 21st edition, p. 71.

Google (2019). Cloud Translation API. Retrieved from https://cloud.google.com/translate/docs/reference/rest/

Hätty, Anna & Schulte Im Walde, Sabine. (2018). Fine-grained Termhood Prediction for German Compound Terms using Neural Networks.

Heidegger, Martin. (1962). Being and Time (John Macquarrie & Edward Robinson). Oxford: Blackwell Publishers Ltd. (Original work published 1926)

Krzysztof Wołk and Krzysztof Marasek: Building Subject-aligned Comparable Corpora and Mining it for Truly Parallel Sentence Pairs., Procedia Technology, 18, Elsevier, p.126-132, 2014

Landes, Shari, and Claudia Leacock, and Christiane Fellbaum. 1998. Building semantic concordances. In Fellbaum (1998), chapter 8, pages 199–216.

Melamed, D. (2001). Empirical Methods for Exploiting Parallel Texts. The MIT Press, Cambridge Massachusetts, London, England, 195 pages.

Meibauer, Jörg. (2013). Expressive compounds in German. Word Structure. 6. 21-42. 10.3366/word.2013.0034.

P. Lison and J. Tiedemann, 2016, OpenSubtitles2016: Extracting Large Parallel Corpora from Movie and TV Subtitles. In Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC 2016)

Paul, Michael & Arora, Karunesh & Sumita, Eiichiro. (2009). Translation of Untranslatable Words - Integration of Lexical Approximation and Phrase-Table Extension Techniques into Statistical Machine Translation. IEICE Transactions. 92-D. 2378-2385.

Tiedemann, J. (2012). Parallel Data, Tools and Interfaces in OPUS. In Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC 2012)

Tufiş, D., Barbu, A., & Ion, R. (2004). Extracting Multilingual Lexicons from Parallel Corpora. Computers and the Humanities,38(2), 163-189.

Tuggener, Don (2016). Incremental Coreference Resolution for German. University of Zurich, Faculty of Arts.

University of Michigan (2017). German Words that Express Concepts for which English Lacks Suitable Words. Retrieved from https://resources.german.lsa.umich.edu /vokabeln/deutschhilftenglisch/

Smith, Steph (2018). Eunoia. Retrieved from https://eunoia.world/

Wikipedia contributors. (2019, January 26). BLEU. In Wikipedia, The Free Encyclopedia. Retrieved from https://en.wikipedia.org/w/index.php?title=BLEU&oldid=880303382

Wikipedia contributors. (2019, January 5). Sauregurkenzeit. In Wikipedia, The Free Encyclopedia. Retrieved from https://de.wikipedia.org/w/ index.php?title=Sauregurkenzeit&oldid=184411602

Zakaria, Ingie. (2017). Quantifying a successful translation: A cognitive frame analysis of (un)translatability.

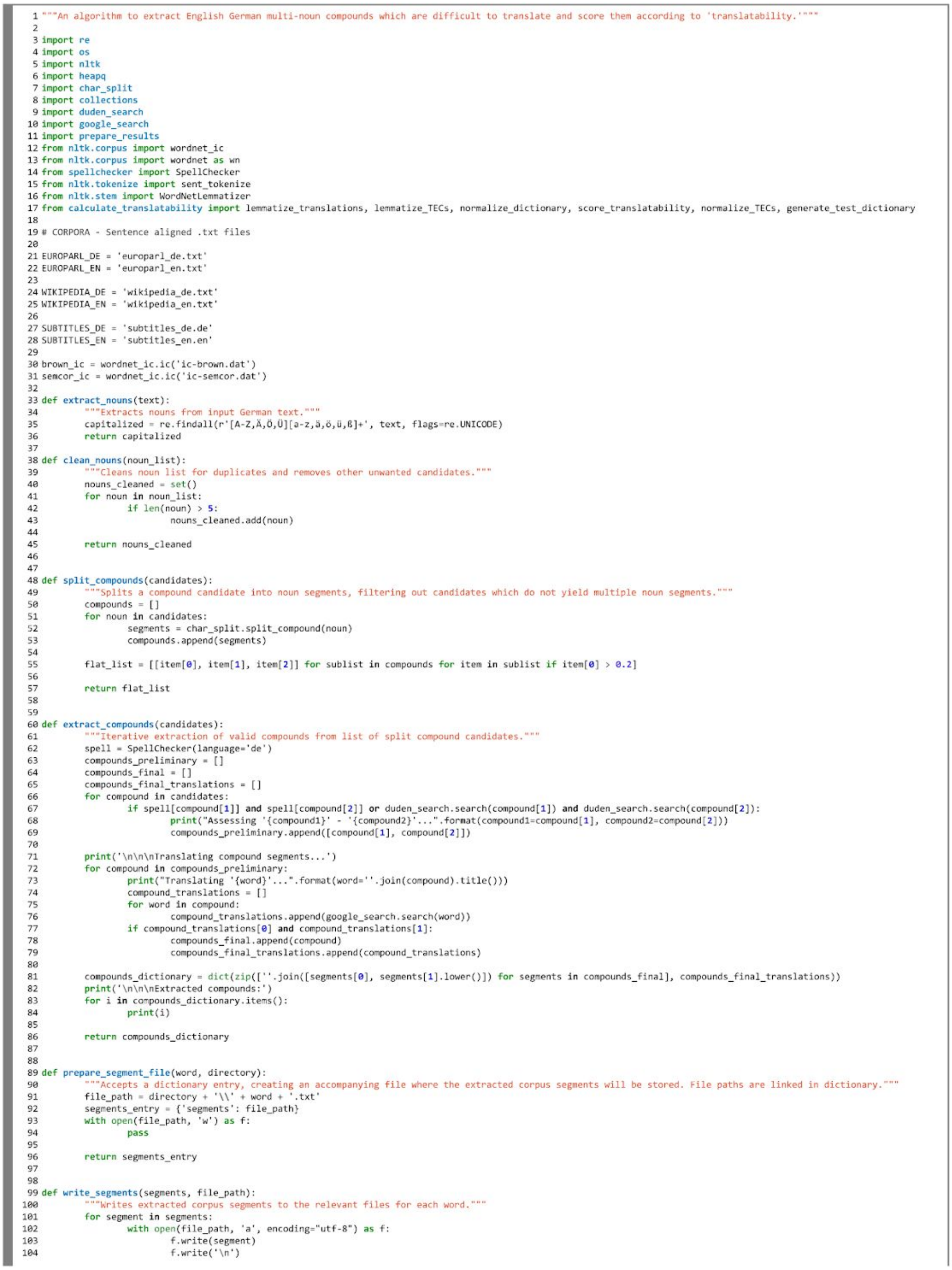
GitHub Repository

https://github.com/christianj6/Assessing-Translatability

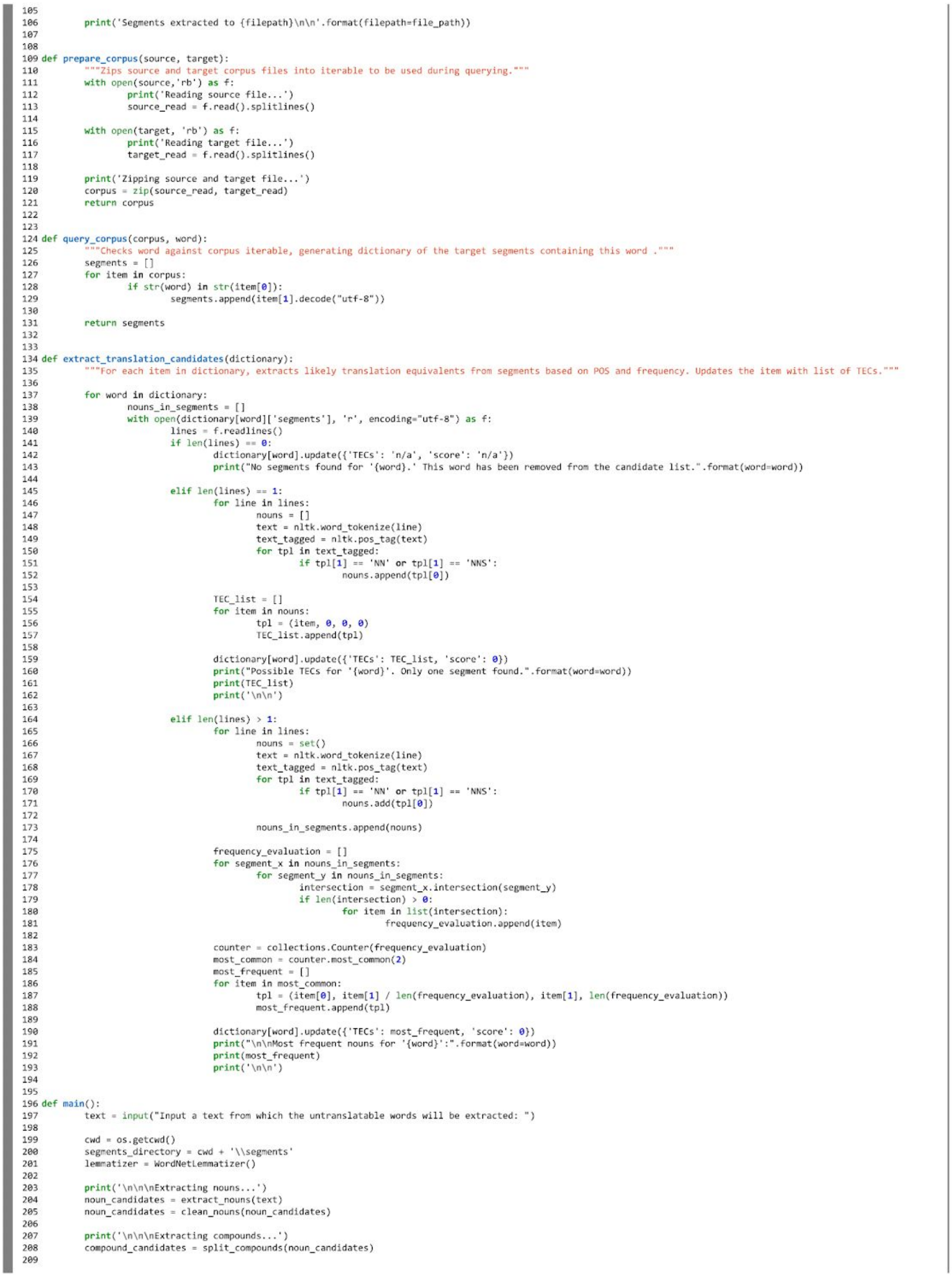
Dependencies

- CharSplit (please use the modified script included in the GitHub repository. Other files associated with this package should be downloaded from the source repository.
- nltk (stem, corpus, tokenize)
- heapq
- bs4
- Spellchecker

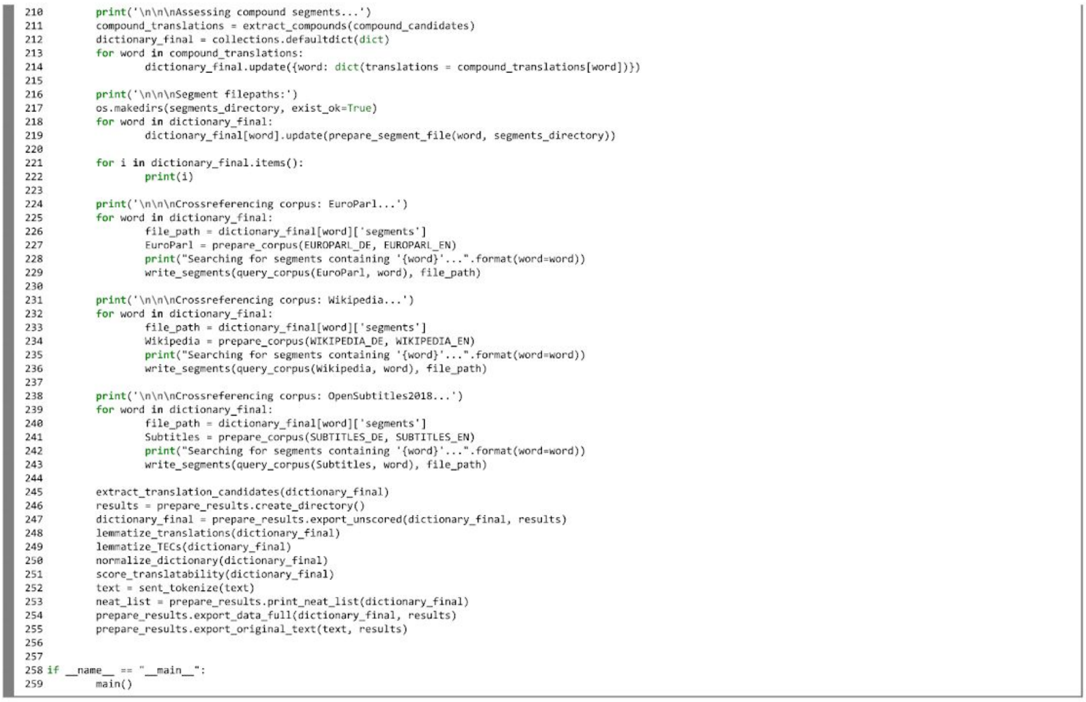
Source Code



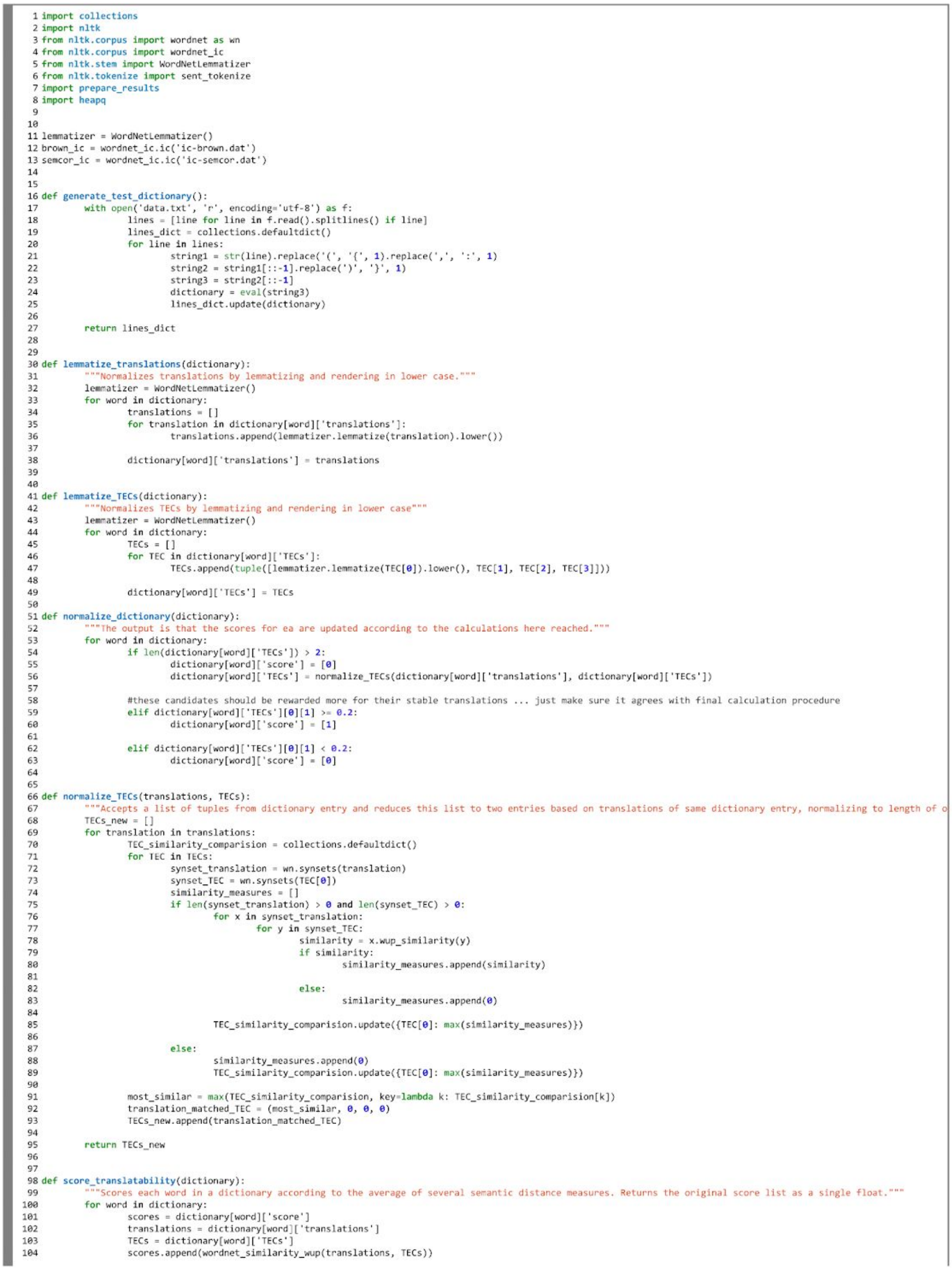
```python
"""An algorithm to extract English German multi-noun compounds which are difficult to translate and score them according to 'translatability.'"""

import re
import os
import nltk
import heapq
import char_split
import collections
import duden_search
import google_search
import prepare_results
from nltk.corpus import wordnet_ic
from nltk.corpus import wordnet as wn
from spellchecker import SpellChecker
from nltk.tokenize import sent_tokenize
from nltk.stem import WordNetLemmatizer
from calculate_translatability import lemmatize_translations, lemmatize_TECs, normalize_dictionary, score_translatability, normalize_TECs, generate_test_dictionary

# CORPORA - Sentence aligned .txt files

EUROPARL_DE = 'europarl_de.txt'
EUROPARL_EN = 'europarl_en.txt'

WIKIPEDIA_DE = 'wikipedia_de.txt'
WIKIPEDIA_EN = 'wikipedia_en.txt'

SUBTITLES_DE = 'subtitles_de.de'
SUBTITLES_EN = 'subtitles_en.en'

brown_ic = wordnet_ic.ic('ic-brown.dat')
semcor_ic = wordnet_ic.ic('ic-semcor.dat')

def extract_nouns(text):
        """Extracts nouns from input German text."""
        capitalized = re.findall(r'[A-Z,Ä,Ö,Ü][a-z,ä,ö,ü,ß]+', text, flags=re.UNICODE)
        return capitalized

def clean_nouns(noun_list):
        """Cleans noun list for duplicates and removes other unwanted candidates."""
        nouns_cleaned = set()
        for noun in noun_list:
                if len(noun) > 5:
                        nouns_cleaned.add(noun)

        return nouns_cleaned


def split_compounds(candidates):
        """Splits a compound candidate into noun segments, filtering out candidates which do not yield multiple noun segments."""
        compounds = []
        for noun in candidates:
                segments = char_split.split_compound(noun)
                compounds.append(segments)

        flat_list = [[item[0], item[1], item[2]] for sublist in compounds for item in sublist if item[0] > 0.2]

        return flat_list


def extract_compounds(candidates):
        """Iterative extraction of valid compounds from list of split compound candidates."""
        spell = SpellChecker(language='de')
        compounds_preliminary = []
        compounds_final = []
        compounds_final_translations = []
        for compound in candidates:
                if spell[compound[1]] and spell[compound[2]] or duden_search.search(compound[1]) and duden_search.search(compound[2]):
                        print("Assessing '{compound1}' - '{compound2}'...".format(compound1=compound[1], compound2=compound[2]))
                        compounds_preliminary.append([compound[1], compound[2]])

        print('\n\n\nTranslating compound segments...')
        for compound in compounds_preliminary:
                print("Translating '{word}'...".format(word=''.join(compound).title()))
                compound_translations = []
                for word in compound:
                        compound_translations.append(google_search.search(word))
                if compound_translations[0] and compound_translations[1]:
                        compounds_final.append(compound)
                        compounds_final_translations.append(compound_translations)

        compounds_dictionary = dict(zip([''.join([segments[0], segments[1].lower()]) for segments in compounds_final], compounds_final_translations))
        print('\n\n\nExtracted compounds:')
        for i in compounds_dictionary.items():
                print(i)

        return compounds_dictionary


def prepare_segment_file(word, directory):
        """Accepts a dictionary entry, creating an accompanying file where the extracted corpus segments will be stored. File paths are linked in dictionary."""
        file_path = directory + '\\' + word + '.txt'
        segments_entry = {'segments': file_path}
        with open(file_path, 'w') as f:
                pass

        return segments_entry


def write_segments(segments, file_path):
        """Writes extracted corpus segments to the relevant files for each word."""
        for segment in segments:
                with open(file_path, 'a', encoding="utf-8") as f:
                        f.write(segment)
                        f.write('\n')
```



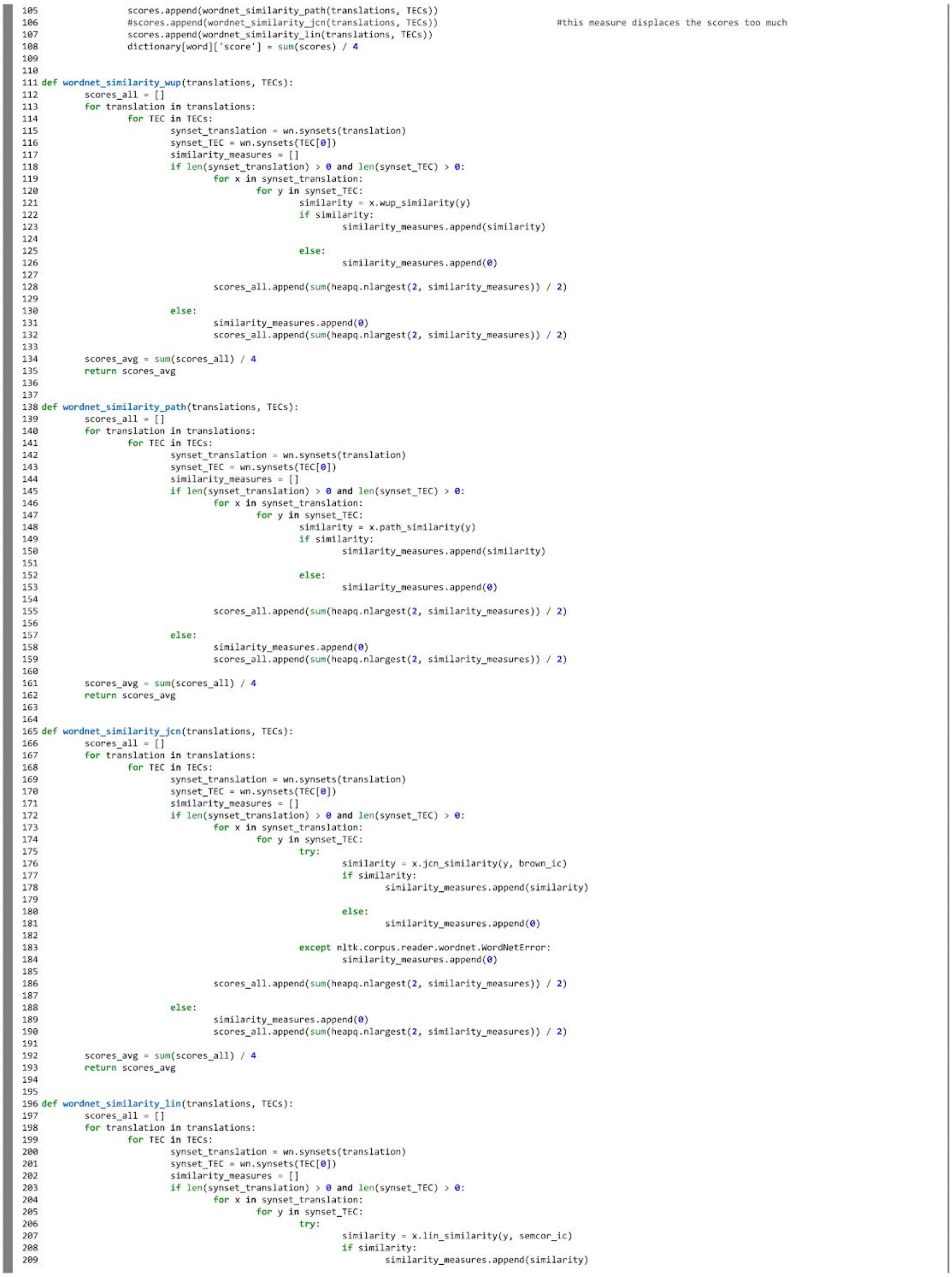
```

        print('Segments extracted to {filepath}\n\n'.format(filepath=file_path))


def prepare_corpus(source, target):
        """Zips source and target corpus files into iterable to be used during querying."""
        with open(source,'rb') as f:
                print('Reading source file...')
                source_read = f.read().splitlines()

        with open(target, 'rb') as f:
                print('Reading target file...')
                target_read = f.read().splitlines()

        print('Zipping source and target file...')
        corpus = zip(source_read, target_read)
        return corpus


def query_corpus(corpus, word):
        """Checks word against corpus iterable, generating dictionary of the target segments containing this word ."""
        segments = []
        for item in corpus:
                if str(word) in str(item[0]):
                        segments.append(item[1].decode("utf-8"))

        return segments


def extract_translation_candidates(dictionary):
        """For each item in dictionary, extracts likely translation equivalents from segments based on POS and frequency. Updates the item with list of TECs."""

        for word in dictionary:
                nouns_in_segments = []
                with open(dictionary[word]['segments'], 'r', encoding="utf-8") as f:
                        lines = f.readlines()
                        if len(lines) == 0:
                                dictionary[word].update({'TECs': 'n/a', 'score': 'n/a'})
                                print("No segments found for '{word}.' This word has been removed from the candidate list.".format(word=word))

                        elif len(lines) == 1:
                                for line in lines:
                                        nouns = []
                                        text = nltk.word_tokenize(line)
                                        text_tagged = nltk.pos_tag(text)
                                        for tpl in text_tagged:
                                                if tpl[1] == 'NN' or tpl[1] == 'NNS':
                                                        nouns.append(tpl[0])

                                TEC_list = []
                                for item in nouns:
                                        tpl = (item, 0, 0, 0)
                                        TEC_list.append(tpl)

                                dictionary[word].update({'TECs': TEC_list, 'score': 0})
                                print("Possible TECs for '{word}'. Only one segment found.".format(word=word))
                                print(TEC_list)
                                print('\n\n')

                        elif len(lines) > 1:
                                for line in lines:
                                        nouns = set()
                                        text = nltk.word_tokenize(line)
                                        text_tagged = nltk.pos_tag(text)
                                        for tpl in text_tagged:
                                                if tpl[1] == 'NN' or tpl[1] == 'NNS':
                                                        nouns.add(tpl[0])

                                        nouns_in_segments.append(nouns)

                                frequency_evaluation = []
                                for segment_x in nouns_in_segments:
                                        for segment_y in nouns_in_segments:
                                                intersection = segment_x.intersection(segment_y)
                                                if len(intersection) > 0:
                                                        for item in list(intersection):
                                                                frequency_evaluation.append(item)

                                counter = collections.Counter(frequency_evaluation)
                                most_common = counter.most_common(2)
                                most_frequent = []
                                for item in most_common:
                                        tpl = (item[0], item[1] / len(frequency_evaluation), item[1], len(frequency_evaluation))
                                        most_frequent.append(tpl)

                                dictionary[word].update({'TECs': most_frequent, 'score': 0})
                                print("\n\nMost frequent nouns for '{word}':".format(word=word))
                                print(most_frequent)
                                print('\n\n')


def main():
        text = input("Input a text from which the untranslatable words will be extracted: ")

        cwd = os.getcwd()
        segments_directory = cwd + '\\segments'
        lemmatizer = WordNetLemmatizer()

        print('\n\n\nExtracting nouns...')
        noun_candidates = extract_nouns(text)
        noun_candidates = clean_nouns(noun_candidates)

        print('\n\n\nExtracting compounds...')
        compound_candidates = split_compounds(noun_candidates)

```



```
    print('\n\n\nAssessing compound segments...')
    compound_translations = extract_compounds(compound_candidates)
    dictionary_final = collections.defaultdict(dict)
    for word in compound_translations:
        dictionary_final.update({word: dict(translations = compound_translations[word])})

    print('\n\n\nSegment filepaths:')
    os.makedirs(segments_directory, exist_ok=True)
    for word in dictionary_final:
        dictionary_final[word].update(prepare_segment_file(word, segments_directory))

    for i in dictionary_final.items():
        print(i)

    print('\n\n\nCrossreferencing corpus: EuroParl...')
    for word in dictionary_final:
        file_path = dictionary_final[word]['segments']
        EuroParl = prepare_corpus(EUROPARL_DE, EUROPARL_EN)
        print("Searching for segments containing '{word}'...".format(word=word))
        write_segments(query_corpus(EuroParl, word), file_path)

    print('\n\n\nCrossreferencing corpus: Wikipedia...')
    for word in dictionary_final:
        file_path = dictionary_final[word]['segments']
        Wikipedia = prepare_corpus(WIKIPEDIA_DE, WIKIPEDIA_EN)
        print("Searching for segments containing '{word}'...".format(word=word))
        write_segments(query_corpus(Wikipedia, word), file_path)

    print('\n\n\nCrossreferencing corpus: OpenSubtitles2018...')
    for word in dictionary_final:
        file_path = dictionary_final[word]['segments']
        Subtitles = prepare_corpus(SUBTITLES_DE, SUBTITLES_EN)
        print("Searching for segments containing '{word}'...".format(word=word))
        write_segments(query_corpus(Subtitles, word), file_path)

    extract_translation_candidates(dictionary_final)
    results = prepare_results.create_directory()
    dictionary_final = prepare_results.export_unscored(dictionary_final, results)
    lemmatize_translations(dictionary_final)
    lemmatize_TECs(dictionary_final)
    normalize_dictionary(dictionary_final)
    score_translatability(dictionary_final)
    text = sent_tokenize(text)
    neat_list = prepare_results.print_neat_list(dictionary_final)
    prepare_results.export_data_full(dictionary_final, results)
    prepare_results.export_original_text(text, results)


if __name__ == "__main__":
    main()
```



```python
import collections
import nltk
from nltk.corpus import wordnet as wn
from nltk.corpus import wordnet_ic
from nltk.stem import WordNetLemmatizer
from nltk.tokenize import sent_tokenize
import prepare_results
import heapq


lemmatizer = WordNetLemmatizer()
brown_ic = wordnet_ic.ic('ic-brown.dat')
semcor_ic = wordnet_ic.ic('ic-semcor.dat')


def generate_test_dictionary():
        with open('data.txt', 'r', encoding='utf-8') as f:
                lines = [line for line in f.read().splitlines() if line]
                lines_dict = collections.defaultdict()
                for line in lines:
                        string1 = str(line).replace('(', '{', 1).replace(',', ':', 1)
                        string2 = string1[::-1].replace(')', '}', 1)
                        string3 = string2[::-1]
                        dictionary = eval(string3)
                        lines_dict.update(dictionary)

        return lines_dict


def lemmatize_translations(dictionary):
        """Normalizes translations by lemmatizing and rendering in lower case."""
        lemmatizer = WordNetLemmatizer()
        for word in dictionary:
                translations = []
                for translation in dictionary[word]['translations']:
                        translations.append(lemmatizer.lemmatize(translation).lower())

                dictionary[word]['translations'] = translations


def lemmatize_TECs(dictionary):
        """Normalizes TECs by lemmatizing and rendering in lower case"""
        lemmatizer = WordNetLemmatizer()
        for word in dictionary:
                TECs = []
                for TEC in dictionary[word]['TECs']:
                        TECs.append(tuple([lemmatizer.lemmatize(TEC[0]).lower(), TEC[1], TEC[2], TEC[3]]))

                dictionary[word]['TECs'] = TECs

def normalize_dictionary(dictionary):
        """The output is that the scores for ea are updated according to the calculations here reached."""
        for word in dictionary:
                if len(dictionary[word]['TECs']) > 2:
                        dictionary[word]['score'] = [0]
                        dictionary[word]['TECs'] = normalize_TECs(dictionary[word]['translations'], dictionary[word]['TECs'])

                #these candidates should be rewarded more for their stable translations ... just make sure it agrees with final calculation procedure
                elif dictionary[word]['TECs'][0][1] >= 0.2:
                        dictionary[word]['score'] = [1]

                elif dictionary[word]['TECs'][0][1] < 0.2:
                        dictionary[word]['score'] = [0]


def normalize_TECs(translations, TECs):
        """Accepts a list of tuples from dictionary entry and reduces this list to two entries based on translations of same dictionary entry, normalizing to length of o
        TECs_new = []
        for translation in translations:
                TEC_similarity_comparision = collections.defaultdict()
                for TEC in TECs:
                        synset_translation = wn.synsets(translation)
                        synset_TEC = wn.synsets(TEC[0])
                        similarity_measures = []
                        if len(synset_translation) > 0 and len(synset_TEC) > 0:
                                for x in synset_translation:
                                        for y in synset_TEC:
                                                similarity = x.wup_similarity(y)
                                                if similarity:
                                                        similarity_measures.append(similarity)

                                                else:
                                                        similarity_measures.append(0)

                                TEC_similarity_comparision.update({TEC[0]: max(similarity_measures)})

                        else:
                                similarity_measures.append(0)
                                TEC_similarity_comparision.update({TEC[0]: max(similarity_measures)})

                most_similar = max(TEC_similarity_comparision, key=lambda k: TEC_similarity_comparision[k])
                translation_matched_TEC = (most_similar, 0, 0, 0)
                TECs_new.append(translation_matched_TEC)

        return TECs_new


def score_translatability(dictionary):
        """Scores each word in a dictionary according to the average of several semantic distance measures. Returns the original score list as a single float."""
        for word in dictionary:
                scores = dictionary[word]['score']
                translations = dictionary[word]['translations']
                TECs = dictionary[word]['TECs']
                scores.append(wordnet_similarity_wup(translations, TECs))
```



```
        scores.append(wordnet_similarity_path(translations, TECs))
        #scores.append(wordnet_similarity_jcn(translations, TECs))                          #this measure displaces the scores too much
        scores.append(wordnet_similarity_lin(translations, TECs))
        dictionary[word]['score'] = sum(scores) / 4


def wordnet_similarity_wup(translations, TECs):
    scores_all = []
    for translation in translations:
        for TEC in TECs:
            synset_translation = wn.synsets(translation)
            synset_TEC = wn.synsets(TEC[0])
            similarity_measures = []
            if len(synset_translation) > 0 and len(synset_TEC) > 0:
                for x in synset_translation:
                    for y in synset_TEC:
                        similarity = x.wup_similarity(y)
                        if similarity:
                            similarity_measures.append(similarity)

                        else:
                            similarity_measures.append(0)

                scores_all.append(sum(heapq.nlargest(2, similarity_measures)) / 2)

            else:
                similarity_measures.append(0)
                scores_all.append(sum(heapq.nlargest(2, similarity_measures)) / 2)

    scores_avg = sum(scores_all) / 4
    return scores_avg


def wordnet_similarity_path(translations, TECs):
    scores_all = []
    for translation in translations:
        for TEC in TECs:
            synset_translation = wn.synsets(translation)
            synset_TEC = wn.synsets(TEC[0])
            similarity_measures = []
            if len(synset_translation) > 0 and len(synset_TEC) > 0:
                for x in synset_translation:
                    for y in synset_TEC:
                        similarity = x.path_similarity(y)
                        if similarity:
                            similarity_measures.append(similarity)

                        else:
                            similarity_measures.append(0)

                scores_all.append(sum(heapq.nlargest(2, similarity_measures)) / 2)

            else:
                similarity_measures.append(0)
                scores_all.append(sum(heapq.nlargest(2, similarity_measures)) / 2)

    scores_avg = sum(scores_all) / 4
    return scores_avg


def wordnet_similarity_jcn(translations, TECs):
    scores_all = []
    for translation in translations:
        for TEC in TECs:
            synset_translation = wn.synsets(translation)
            synset_TEC = wn.synsets(TEC[0])
            similarity_measures = []
            if len(synset_translation) > 0 and len(synset_TEC) > 0:
                for x in synset_translation:
                    for y in synset_TEC:
                        try:
                            similarity = x.jcn_similarity(y, brown_ic)
                            if similarity:
                                similarity_measures.append(similarity)

                            else:
                                similarity_measures.append(0)

                        except nltk.corpus.reader.wordnet.WordNetError:
                            similarity_measures.append(0)

                scores_all.append(sum(heapq.nlargest(2, similarity_measures)) / 2)

            else:
                similarity_measures.append(0)
                scores_all.append(sum(heapq.nlargest(2, similarity_measures)) / 2)

    scores_avg = sum(scores_all) / 4
    return scores_avg


def wordnet_similarity_lin(translations, TECs):
    scores_all = []
    for translation in translations:
        for TEC in TECs:
            synset_translation = wn.synsets(translation)
            synset_TEC = wn.synsets(TEC[0])
            similarity_measures = []
            if len(synset_translation) > 0 and len(synset_TEC) > 0:
                for x in synset_translation:
                    for y in synset_TEC:
                        try:
                            similarity = x.lin_similarity(y, semcor_ic)
                            if similarity:
                                similarity_measures.append(similarity)
```

Evaluation Texts

Text One: Generic Text with 'Untranslatable' Words

Die ungarische Sprache kennt diesen Begriff ebenfalls als uborkaszezon (Gurkensaison). Die magyarisierte Form des Wortes Saison lässt darauf schließen, dass dieses Kompositum in seiner heutigen Form im 19. Jahrhundert entstanden sein dürfte. Allerdings ist das dahinter stehende Brauchtum älter. Der Begriff wird sowohl analog zur deutschen Sprache im übertragenen Sinn benutzt als auch nach wie vor im wörtlichen Sinne. Es ist die Zeit zu Beginn des Sommers, wenn die salzigen „Sommergurken" (ungarisch kovászos uborka) eingelegt werden. Der Name kommt vom ungarischen Wort für Sauerteig (Kovász); denn diese Gurken werden im Gegensatz zu den „Wintergurken" nicht in Essig eingelegt und auch nicht durch Erhitzen haltbar gemacht, sondern verdanken ihren salzig-sauren Geschmack einer kurzen Milchsäuregärung. Die Gurken werden dazu nur in Salzwasser mit Gewürzen eingelegt und mit Hilfe einer Scheibe Brot vergoren, indem man die Gläser abgedeckt auf die Fensterbank beziehungsweise auf die Mauer des Arkadengangs oder auf die Terrasse stellt. Morgenmuffel Kopfkino Schnapsidee Fingerspitzengefühl Torschlusspanik Geschmacksverirrung Fernweh Weltschmerz Fremdschämen Zugzwang Feierabend Ohrwurm Sitzfleisch Erbsenzähler Lebensmüdigkeit Lebensraum Treppenwitz Geisterfahrer Landzunge

Text Two: Biographical Text with Specialized Vocabulary

In den Werken Nietzsches lässt sich zeigen, dass er schon in jungen Jahren[A 2] einen Zugang zu den Themen der Metaphysik, der Religion und der Moral, später auch des Ästhetischen, aus einem historisch-kritischen Blickwinkel forderte. Alle Erklärungsmuster, die auf etwas Transzendentes, Unbedingtes, Universales abzielen, sind nichts als Mythen, die in der Geschichte der Erkenntnisentwicklung jeweils auf der Grundlage des Wissens ihrer Zeit entstanden sind. Dieses aufzudecken ist Aufgabe der modernen Wissenschaft und Philosophie. In diesem Sinne verstand sich Nietzsche als Verfechter eines radikalen Aufklärungsgedankens. „[…] erst nachdem wir die historische Betrachtungsart, welche die Zeit der Aufklärung mit sich brachte, in einem so wesentlichen Puncte corrigirt haben, dürfen wir die Fahne der Aufklärung — die Fahne mit den drei Namen: Petrarca, Erasmus, Voltaire — von Neuem weiter tragen. Wir haben aus der Reaction einen Fortschritt gemacht."[47] Den Begriff der Genealogie verwendete er erstmals im Titel der Genealogie der Moral. Die Methodik wird dort insbesondere in der zweiten Abhandlung in den Abschnitten 12 bis 14 ausgeführt.[48] Die dahinter stehende Methode beschrieb und praktizierte er bereits in Menschliches, Allzumenschliches (Aphorismen 1 und 2), und bereits in der Zweiten Unzeitgemäßen Betrachtung reflektierte er den Wert des Historischen kritisch, zeigte dessen Grenzen, aber auch seine Unhintergehbarkeit.[49] Genealogie bedeutet für Nietzsche nicht historische Forschung, sondern kritische Erklärung von Gegenwartsphänomenen anhand von (spekulativen) theoretischen Ableitungen aus der Geschichte. Im Mittelpunkt steht eine „Deplausibilisierung"[50] bisheriger Narrative in Philosophie, Theologie und den kulturwissenschaftlichen Fragen durch historisch gestützte psychologische Thesen. Großen Einfluss hat dieses Konzept Nietzsches auf Michel Foucault.[51] Josef Simon setzte die Methode mit der modernen Dekonstruktion gleich.[52] Aus seiner Kritik von Metaphysik, Erkenntnistheorie, Moralphilosophie und Religion heraus entwickelte Nietzsche selbst ein pluralistisches Weltbild. Indem er die Welt und auch den Menschen als einen im ständigen Werden befindlichen Organismus auffasste, in dem eine Vielzahl von Elementen im ständigen Gegeneinander ihrer Kräfte danach ringt, sich durchzusetzen, löste er sich vom traditionellen Substanzdenken und von jeglichen kausal-mechanistischen sowie teleologischen Erklärungen.[53] „Alle Einheit ist nur als Organisation und Zusammenspiel Einheit: nicht anders als wie ein menschliches Gemeinwesen eine Einheit ist: also Gegensatz der atomistischen Anarchie; somit ein Herrschafts-Gebilde, das Eins bedeutet, aber nicht eins ist."[54] In diesem Organismus als Totalität wirken die verschiedensten Kräfte im Kampf gegeneinander; sie folgen ihrem jeweiligen Willen zur Macht (s. u.). „Leben wäre zu definiren als eine dauernde Form von Prozeß der Kraftfeststellungen, wo die verschiedenen Kämpfenden ihrerseits ungleich wachsen."[55] Jeder Organismus führt seinen Kampf aus seiner eigenen Perspektive. „Seien wir zuletzt, gerade als Erkennende, nicht undankbar gegen solche resolute Umkehrungen der gewohnten Perspektiven und Werthungen, mit denen der Geist allzulange scheinbar freventlich und nutzlos gegen sich selbst gewüthet hat: dergestalt einmal anders sehn, anders-sehn-wollen ist keine kleine Zucht und Vorbereitung des Intellekts zu seiner einstmaligen ‚Objektivität', – letztere nicht als ‚interesselose Anschauung' verstanden (als welche ein Unbegriff und Widersinn ist), sondern als das Vermögen, sein Für und Wider in der Gewalt zu haben und aus- und einzuhängen: so dass man sich gerade die Verschiedenheit der Perspektiven und der Affekt-Interpretationen für die Erkenntniss nutzbar zu machen weiss […] Es giebt nur ein perspektivisches Sehen, nur ein perspektivisches ‚Erkennen'; und je mehr Affekte wir über eine Sache zu Worte kommen lassen, je mehr Augen, verschiedne Augen wir uns für dieselbe Sache einzusetzen wissen, um so vollständiger wird unser ‚Begriff' dieser Sache, unsre ‚Objektivität' sein. Den Willen aber überhaupt eliminiren, die Affekte sammt und sonders aushängen, gesetzt, dass wir dies vermöchten: wie? hiesse das nicht den Intellekt castriren?" Die subjektive Sicht, die zur Perspektive führt, bedeutet nun weder Willkür noch Relativismus. Die jeweils eingenommene Perspektive führt vielmehr dazu, dass der Mensch die Welt, wie sie ihm erscheint, zu einem Bild, zu einer Interpretation zusammenfügt. „Daß der Werth der Welt in unserer Interpretation liegt (– daß vielleicht irgendwo noch andere Interpretationen möglich sind als bloß menschliche –) daß die bisherigen Interpretationen perspektivische Schätzungen sind, vermöge deren wir uns im Leben, das heißt im Willen zur Macht, zum Wachsthum der Macht erhalten, daß jede Erhöhung des Menschen die Überwindung engerer Interpretationen mit sich bringt, daß jede erreichte Verstärkung und Machterweiterung neue Perspektiven aufthut und an neue Horizonte glauben heißt—dies geht durch meine Schriften."[56] Der „Wille zur Macht" ist erstens ein Konzept, das zum ersten Mal in Also sprach Zarathustra vorgestellt und in allen nachfolgenden Büchern zumindest am Rande erwähnt wird. Seine Anfänge liegen in den psychologischen Analysen des menschlichen Machtwillens in der Morgenröte. Umfassender führte es Nietzsche in seinen nachgelassenen Notizbüchern ab etwa 1885 aus. Zweitens ist es der Titel eines von Nietzsche auch als Umwertung aller Werte geplanten Werks, das nie zustande kam. Aufzeichnungen dazu gingen vor allem in die Werke Götzen-Dämmerung und Der Antichrist ein. Drittens ist es der Titel einer Nachlasskompilation von Elisabeth Förster-Nietzsche und Peter Gast, die nach Ansicht dieser Herausgeber dem unter Punkt zwei geplanten „Hauptwerk" entsprechen soll. Die Deutung des Konzepts „Wille zur Macht" ist stark umstritten. Für Martin Heidegger war es Nietzsches Antwort auf die metaphysische Frage nach dem „Grund alles Seienden“: Laut Nietzsche sei alles „Wille zur Macht“ im Sinne eines inneren, metaphysischen Prinzips, so wie dies bei Schopenhauer der „Wille (zum Leben)“ ist. Die entgegengesetzte Meinung vertrat Wolfgang Müller-Lauter: Danach habe Nietzsche mit dem „Willen zur Macht“ keineswegs eine Metaphysik im Sinne Heideggers wiederhergestellt – Nietzsche war ja gerade Kritiker jeder Metaphysik –, sondern den Versuch unternommen, eine in sich konsistente Deutung allen Geschehens zu geben, die die nach Nietzsche irrtümlichen Annahmen sowohl metaphysischer „Sinngebungen“ als auch eines atomistisch-materialistischen Weltbildes vermeide. Um Nietzsches Konzept zu begreifen, sei es angemessener, von den (vielen) „Willen zur Macht“ zu sprechen, die im dauernden Widerstreit miteinander stehen, sich gegenseitig bezwingen und einverleiben, zeitweilige Organisationen (beispielsweise den menschlichen Leib), aber keinerlei „Ganzes“ bilden, denn die Welt sei ewiges Chaos. Zwischen diesen beiden Interpretationen bewegen sich die meisten anderen, wobei die heutige Nietzscheforschung derjenigen Müller-Lauters deutlich näher steht. Gerade der Begriff Macht weist jedoch bei Nietzsche (mit seiner stets auf das gesunde Individuum ausgerichteten Weltanschauung) auf neuere positive Verständnisformen voraus, wie wir sie z. B. bei Hannah Arendt[57] finden – hier jedoch bezogen auf den Menschen in der Gesellschaft: die grundsätzliche Möglichkeit aus sich heraus gestaltend „etwas zu machen“. Nietzsches zuerst in Die fröhliche Wissenschaft auftretender und in Also sprach Zarathustra als Höhepunkt vorgeführter „tiefster Gedanke“, der ihm auf einer Wanderung im Engadin nahe Sils-Maria kam, ist die Vorstellung, dass alles Geschehende schon unendlich oft geschah und unendlich oft wiederkehren wird. Man solle deshalb so leben, dass man die immerwährende Wiederholung eines jeden Augenblickes nicht nur ertrage, sondern sogar begrüße. „Doch alle Lust will Ewigkeit – will tiefe, tiefe Ewigkeit“[58] lautet folglich ein zentraler Satz in Also sprach Zarathustra. Eng mit der „Ewigen Wiederkunft“, für die Nietzsche trotz seiner nur sehr oberflächlichen naturwissenschaftlichen Bildung auch wissenschaftliche Begründungen zu geben versuchte, hängt wohl der Amor fati (lat. „Liebe zum Schicksal“) zusammen. Dies ist für Nietzsche eine Formel zur Bezeichnung des höchsten Zustands, den ein Philosoph erreichen kann, die Form der höchstgesteigerten Lebensbejahung.[59] Über die „ewige Wiederkunft“, ihre Bedeutung und Stellung in Nietzsches Gedanken herrscht keine Einigkeit. Während einige Deuter sie als Zentrum seines gesamten Denkens ausmachten, sahen andere sie bloß als fixe Idee und störenden „Fremdkörper“ in Nietzsches Lehren. Übermensch An einen Fortschritt in der Geschichte der Menschheit – oder in der Welt überhaupt – glaubt Nietzsche nicht. Für ihn ist folglich das Ziel der Menschheit nicht an ihrem (zeitlichen) Ende zu finden, sondern in ihren immer wieder auftretenden höchsten Individuen, den Übermenschen. Die Gattung Mensch als Ganzes sieht er nur als einen Versuch, eine Art Grundmasse, aus der heraus er „Schaffende“ fordert, die „hart“ und mitleidlos mit anderen und vor allem mit sich selbst sind, um aus der Menschheit und sich selbst ein wertvolles Kunstwerk zu schaffen. Als negatives Gegenstück zum Übermenschen wird in Also sprach Zarathustra der letzte Mensch vorgestellt. Dieser steht für das schwächliche Bestreben nach Angleichung der Menschen untereinander, nach einem möglichst risikolosen, langen und „glücklichen“ Leben ohne Härten und Konflikte. Das Präfix „Über“ in der Wortschöpfung „Übermensch“ kann nicht nur für eine höhere Stufe relativ zu einer anderen stehen, sondern auch im Sinne von „hinüber“ verstanden werden, kann also eine Bewegung ausdrücken. Der Übermensch ist daher nicht unbedingt als Herrenmensch über dem letzten Menschen zu sehen. Eine rein politische Deutung gilt der heutigen Nietzscheforschung als irreführend.

Der „Wille zur Macht“, der sich im Übermenschen konkretisieren soll, ist demnach nicht etwa der Wille zur Herrschaft über andere, sondern ist als Wille zum Können, zur Selbstbereicherung, zur Selbstüberwindung zu verstehen.

Text Three: Historical Text with Culture-Specific Vocabulary

Als Böhmermann-Affäre (auch Fall Böhmermann, Causa Böhmermann, Erdogate oder Staatsaffäre Böhmermann) werden ein Fernsehbeitrag des deutschen Satirikers und Moderators Jan Böhmermann und die darauffolgenden Reaktionen von türkischer und deutscher Seite bezeichnet. Ein kurzes satirisches Gedicht, ausgestrahlt am 31. März 2016 in der Sendereihe Neo Magazin Royale auf ZDFneo, schlug mediale Wellen bis zur New York Times. Böhmermann trug dieses als Moderator, bezugnehmend auf ein satirisches Lied der NDR-Sendung extra 3 („Erdowie, Erdowo, Erdogan“) und türkische Reaktionen, unter dem Titel Schmähkritik vor. Es handelte vom türkischen Staatspräsidenten Recep Tayyip Erdoğan. Eingeleitet und wiederholt unterbrochen wurde der Vortrag von Hinweisen des Moderators und seines Sidekicks Ralf Kabelka, man wolle mit der Lyrik erklären, wie eine in Deutschland verbotene Schmähkritik aussehen könne. Es wurden türkische Untertitel für das Schmähgedicht eingeblendet, allerdings nicht für den erklärenden Kontext. Die Regierung der Türkei und auch Erdoğan selbst bekundeten ihr Strafverlangen bzw. erstatteten Strafanzeige gegen Böhmermann; die Staatsanwaltschaft leitete ein Ermittlungsverfahren ein. Die Äußerung der deutschen Bundeskanzlerin Angela Merkel gegenüber dem türkischen Ministerpräsidenten Ahmet Davutoğlu, der Beitrag sei ihrer Ansicht nach „bewusst verletzend“, wurde in der deutschen Öffentlichkeit und Politik kontrovers diskutiert. Später nannte sie diese Äußerung einen Fehler. Als bekannt wurde, dass die Türkei basierend auf § 103 StGB (Beleidigung von Organen und Vertretern ausländischer Staaten) einen Prozess gegen Böhmermann verlangte, forderten zahlreiche Juristen, Politiker und andere die Abschaffung des Paragrafen. Merkel ließ am 15. April 2016 die Strafverfolgung nach § 103 StGB zu und erklärte zugleich, die Bundesregierung werde bis zum Ende der Legislaturperiode einen Gesetzesentwurf zur Abschaffung des § 103 StGB in den Bundestag einbringen.[1] Am 4. Oktober 2016 gab die Staatsanwaltschaft Mainz bekannt, dass das Strafverfahren gegen Böhmermann eingestellt wurde. Es seien keine „strafbaren Handlungen […] mit der erforderlichen Sicherheit nachzuweisen“, teilte die Behörde mit. Eine Karikatur oder Satire sei keine Beleidigung, sofern „die Überzeichnung menschlicher Schwächen [keine] ernsthafte Herabwürdigung der Person“ enthalte.[2] Am 1. Juni 2017 beschloss der Bundestag einstimmig die Abschaffung des § 103 StGB; sie trat am 1. Januar 2018 in Kraft.[3]