The Thesis committee for Juan Daniel Pinto
Certifies that this is the approved version of the following thesis:

Creating a Frequency Dictionary of Spoken Hebrew

A Reproducible Use of Technology to Overcome Scarcity of Data

APPROVED BY
SUPERVISING COMMITTEE:

Esther L. Raizen, Supervisor

Elaine K. Horwitz, Co-Supervisor

Creating a Frequency Dictionary of Spoken Hebrew

A Reproducible Use of Technology to Overcome Scarcity of Data

by

Juan Daniel Pinto

Thesis

Presented to the Faculty of the Graduate School
of the University of Texas at Austin
in Partial Fulfillment
of the Requirements
for the Degree of

Master of Arts

The University of Texas at Austin
May 2018

Dedication

Interdum et malesuada fames ac ante ipsum primis in faucibus. Aliquam congue fermentum ante, semper porta nisl consectetur ut. Duis ornare sit amet dui ac faucibus. Phasellus ullamcorper leo vitae arcu ultricies cursus. Duis tristique lacus eget metus bibendum, at dapibus ante malesuada. In dictum nulla nec porta varius. Fusce et elit eget sapien fringilla maximus in sit amet dui.

Acknowledgements

Interdum et malesuada fames ac ante ipsum primis in faucibus. Aliquam congue fermentum ante, semper porta nisl consectetur ut. Duis ornare sit amet dui ac faucibus. Phasellus ullamcorper leo vitae arcu ultricies cursus. Duis tristique lacus eget metus bibendum, at dapibus ante malesuada. In dictum nulla nec porta varius. Fusce et elit eget sapien fringilla maximus in sit amet dui.

Mauris eget blandit nisi, faucibus imperdiet odio. Suspendisse blandit dolor sed tellus venenatis, venenatis fringilla turpis pretium. Donec pharetra arcu vitae euismod tincidunt. Morbi ut turpis volutpat, ultrices felis non, finibus justo. Proin convallis accumsan sem ac vulputate. Sed rhoncus ipsum eu urna placerat, sed rhoncus erat facilisis. Praesent vitae vestibulum dui. Proin interdum tellus ac velit varius, sed finibus turpis placerat.

Creating a Frequency Dictionary of Spoken Hebrew

A Reproducible Use of Technology to Overcome Scarcity of Data

by

Juan Daniel Pinto, M.A.
The University of Texas at Austin, 2018
SUPERVISORS: Esther L. Raizen, Elaine K. Horwitz

Interdum et malesuada fames ac ante ipsum primis in faucibus. Aliquam congue fermentum ante, semper porta nisl consectetur ut. Duis ornare sit amet dui ac faucibus. Phasellus ullamcorper leo vitae arcu ultricies cursus. Duis tristique lacus eget metus bibendum, at dapibus ante malesuada. In dictum nulla nec porta varius. Fusce et elit eget sapien fringilla maximus in sit amet dui.

Mauris eget blandit nisi, faucibus imperdiet odio. Suspendisse blandit dolor sed tellus venenatis, venenatis fringilla turpis pretium. Donec pharetra arcu vitae euismod tincidunt. Morbi ut turpis volutpat, ultrices felis non, finibus justo. Proin convallis accumsan sem ac vulputate. Sed rhoncus ipsum eu urna placerat, sed rhoncus erat facilisis. Praesent vitae vestibulum dui. Proin interdum tellus ac velit varius, sed finibus turpis placerat.

Table of Contents

1 Introduction

This thesis provides an in-depth look at the creation of the *Frequency Dictionary of Spoken Hebrew* (FDOSH)—a list of the most common words in spoken Modern Hebrew. Its two-fold aim is (1) to explore the theory behind the creation of the FDOSH, along with implications for similar projects, and (2) to describe the methods and provide the tools to make the process as reproducible as possible.

The complete dictionary itself, consisting of 5,000 items, is included as an electronic supplement and can be downloaded free of charge.[1] A partial list that includes the first 1,000 items can be found in *Appendix A*.

A review of the literature will first highlight the difficulties that exist for less commonly taught languages (LCTLs). Because the overwhelming majority of previous research on vocabulary frequency lists has focused on English (and a handful of other European languages), some important nuances are yet to be addressed. More often than not, the few non-English frequency dictionaries that do exist, along with much of the research in vocabulary acquisition, have taken at face value some of the findings of this limited-scope research—often without questioning whether the same methodologies and conclusions should be applied to different languages.

The present paper is, therefore, an effort to partially fill that gap in order to help educators interested in creating and/or using frequency dictionaries for their own classrooms, for wider dissemination, or simply for general research purposes. In doing so, it will provide an overview of some of the key decisions that must be taken into account for such a project.

The various uses of word frequency lists can be loosely classified into research applications and practical applications. Examples of research applications include traditional linguistic studies that look for common morphological patterns, corpus-linguistic studies seeking to understand language through "real world" texts, and psycholinguistic studies that explore connections between a speaker's mental lexicon and word frequency. Practical applications of frequency lists include curriculum and

[1] Supplements can be downloaded directly from the thesis archive of the University of Texas at Austin. A separate repository at GitHub also contains the complete FDOSH at *https://github.com/juandpinto/opus-frequencies*.

textbook planning for language teachers, vocabulary selection for graded readers and dictionaries, and even independent language study.

Of course, some of the most influential studies straddle both sides of this divide and attempt to answer questions such as: How can vocabulary knowledge be appropriately tested and measured (McLean & Kramer, 2015; Nation, 2016; Nation & Webb, 2010)? What is the role of extensive reading (as opposed to intensive reading) in incidental vocabulary acquisition (Restrepo Ramos, 2015)? What level of vocabulary do learners need in order to read extensively for pleasure (Hirsh & Nation, 1992; Nation, 2006; Schmitt, Jiang, & Grabe, 2011)? What level of vocabulary do learners need in order to succeed in an academic setting (Coxhead, 2000, 2016; Xue & Nation, 1984)? What role does specialized vocabulary play in reaching understanding (Nation & Kyongho, 1995)? These questions and their answers rely heavily on the creation and use of trustworthy frequency dictionaries. Yet due to the resources and effort required to create these lists, they are rarely found for less commonly taught languages.

The primary research question guiding this project is this:

> What are the most common words in spoken Modern Hebrew?

The resulting study also addresses the following secondary research questions:

> What is an effective alternative for a corpus of spoken language when one is lacking in the desired language, as is often the case for less commonly taught languages?

> How can the process of creating a frequency dictionary be simplified so that it is easy for others to reproduce while maintaining a high level of customizability?

> What implications might these findings have for frequency list creation and use as it pertains to other less commonly taught languages?

The literature review will serve as background for many of the important decisions that went into the creation of the FDOSH. These will be explained more in-depth in the *methods* section, where the entire process will be laid out in detail. For the sake of clarity, these key decisions are listed here at the outset. They are as follows:

Corpus size The corpus from which the FDOSH was created needed to contain a minimum of 20 million tokens, though 50 million was preferred. In the end, it used a corpus of nearly 200 million tokens.

Corpus text types In order to best fit with the FDOSH's intended audience (Hebrew learners), the corpus consists of a single text type: conversation. But because of a lack of large corpora of spoken Hebrew, a corpus of film subtitles was used instead. The reasoning for and validity of such an approach will be elaborated on.

Use The primary intended audience for the FDOSH is composed of beginning-to-low-intermediate learners of Hebrew as a foreign language. It is designed for both receptive and productive language use.

Word family levels The word family level that is best suited for the FDOSH's intended audience is the lemma, consisting of a word and all of its inflected forms, but counting derived forms as separate words.

Criteria The FDOSH was created using exclusively objective criteria, meaning that it is the product of calculations, and it was not manually tweaked in any way. The words are sorted by dispersion (specifically, Gries' U_{DP}), and also include the measures of frequency and range.

Following the review of the literature and explanation of theory, the process of the FDOSH's creation will be explained in detail, along with some findings from the project. As already mentioned, the goal of this is to make the process easy to follow and reproduce for other languages. Finally, the FDOSH and all scripts used will be provided in the appendices.

2 Background: Review of the literature

The theoretical foundation of frequency dictionaries—sometimes referred to as lists, word lists, vocabulary lists, and variations thereof—rests on the observation, made popular by the linguist George Kingsley Zipf in the 1930s and 40s, that the first word in any large-enough text occurs roughly twice as often as the second word, three times as often as the third word, and so on (Zipf, 1935, 1949).

This exponential distribution is significant because it means that a small number of words make up the bulk of a text, whereas the majority of the words occur very few times(Sorell, 2012). Paul Nation, one of the most influential scholars in the field of vocabulary acquisition, has pointed out that Zipf's Law—as it is has come to be known—can serve as motivation to language learners and teachers, since learning the most common vocabulary in a language covers such a large percentage of natural communication (Nation, 2013, p. 34).

This observation guides the entire endeavor of frequency dictionary creation and use. Though the FDOSH is not sorted using raw frequency alone[2], the effect of Zipf's law can be easily seen in the listed frequencies that accompany each item.

Beyond understanding this theoretical basis and its implications, other considerations play an important role in the creation of a frequency dictionary. These include corpus size, corpus text type, whether the list will be for general or specialized use, word family levels, and objective criteria. This literature review will treat each of these themes in turn. Because the most comprehensive studies deal with more than one of these issues, some of them will be brought up at various times to illustrate the point under discussion.

2.1 Corpus design

Before designing a frequency dictionary, a careful plan must be made for the design of the corpus from which the list is extracted. The corpus must be representative of the language context that the dictionary wishes to depict. Of course, capturing that

[2]The sorting method and key measures used by the FDOSH is explained in detail in the *objective design* section of this chapter.

context in its entirety is an impossible feat. For this simple reason, researchers must make do with an approximation of the whole: a bounded corpus of language.

Though the focus of this literature review is the creation of word frequency dictionaries, the truth is that relatively few corpora have been created for this specific purpose. Most corpora have aimed at being general collections that cover the language (usually English) as a whole in an attempt to serve different theoretical and applied uses. Yet despite this broad objective, the creation and use of corpora have historically revolved around two big questions: (1) how large should the corpus be, and (2) what kinds of texts should it include. Both of these issues will be addressed here, with the recurring emphasis being corpus design for frequency list creation.

2.1.1 Corpus size

Conventional wisdom in corpus creation states that more is better. If a frequency list is to accurately reflect the frequencies of words in the language as a whole, then a corpus must contain enough text to approximate the overall use of discourse. This line of thinking is equivalent to the maxim in quantitative research that a sample should be as representative of the target population as possible. And in order to maximize the statistical probability of this representation, the sample must be of an appropriate size for the study.

True, larger sample sizes often increase this probability, but they also tend to be more resource-intensive for the researcher. The same is true of corpus size. When creating a frequency dictionary, then, what is an "ideal" corpus size?

The first project to create a one-million-token corpus was a joint effort by Henry Kučera and W. Nelson Francis of Brown University to compile a corpus of American English texts printed in 1961 (Kučera & Francis, 1967), known today simply as the *Brown Corpus*. They strived to create a corpus with equal amounts of texts from different sources by randomly selecting 500 passages of 2,000 words each from different published materials found at the Brown University Library and the Providence Athenaeum. This mixed design would be used as a model by many of the corpora created during the next few decades, which began to be compiled at increasingly faster rates. Many of these corpora were created—in part—to serve as parallel corpora of

different varieties of English.

As an example of how quickly corpora have grown in recent decades, consider the history of COBUILD. What began in 1980 as a collaboration between Collins Publishing and a group of researchers led by John Sinclair—the Collins Birmingham University International Language Database (COBUILD)—led to the creation of the *Collins Corpus* of 7-million-tokens by 1982. It continued expanding until transforming into the *Bank of English* in the 1990s, which reached 320 million words in 1997. In 2005, as part of the Collins World Web, which also comprises French, German, and Spanish corpora, it reached 2.5 billion words (*Collins Cobuild English grammar*, 2005). The Collins Corpus now contains over 4.5 billion words ("The history of Collins COBUILD," n.d.).

Today, with the use of web-crawling applications that scour the internet and collect text at unprecedented speed, the sky's the limit.The *enTenTen12* corpus is composed of 12 billion English tokens, all of which were collected in 12 days (Jakubíček, Kilgarriff, Kovář, Rychlý, & Suchomel, 2013)! At what point, then is a corpus sufficiently large for frequency-list creation?

Researchers have approached this specific problem by creating multiple frequency lists—from varying sizes of corpora—and then comparing the efficacy of these lists themselves. The way that efficacy is operationalized, however, varies among studies.

Some studies have explored how closely the rankings of items on a word frequency dictionary correlate with reaction times in a lexical decision task—a widely-used procedure in psychological and psycholinguistic research (Forster & Chambers, 1973; Oldfield & Wingfield, 1965). In a lexical decision task, participants are presented with a series of words and non-words, one after the other, and they are asked to judge which is which as quickly as possible. Their reaction times are then analyzed for each word. It is generally agreed that the average time it takes participants to react to a word provides insights into the arrangement of the mental lexicon(Gilquin & Gries, 2009; Jurafsky, 2003). For our purposes, multiple studies have found that there exists an inverse correlation between word frequency and reaction time on a lexical decision task (Balota & Chumbley, 1984; Whitney, 1998). In other words, more common words are accessed and recognized more quickly than less common words. Therefore, an effective word frequency list should correspond to and reflect

this reality.

This was precisely the approach taken by Brysbaert and New (2009), who compared respond times collected as part of the massive Elexicon Project (Balota et al., 2007) to words on a series of frequency lists made from increasingly larger corpora. The corpora used were all subcorpora extracted from the British National Corpus (BNC). With each subsequent increase in token count, the word list correlated more and more closely with the response times from lexical decision tasks. Brysbaert and New hoped to find an "ideal" corpus size, after which the increase in effectiveness would no longer be significant enough to justify the additional cost of resources. After conducting several regression analyses on the two sets of data, they found that the variance in the response times that could be accounted for by corpus size reached a plateau at about 16 million tokens. In other words, for corpora with less than 16 million words, the size of the corpus had a significant effect on the correlation between word frequencies and average response times for those words on lexical decision tasks. For corpora with more than 16 million words, the effect of increasing corpus size became considerably more subtle. In the end, they concluded that in order to construct an effective frequency dictionary for *high-frequency* words, a corpus of about 1–3 million tokens is needed. However, in order to reach the same effectiveness for *low-frequency* words, a corpus size of at least 16 million words is preferable (Brysbaert & New, 2009, p. 988).

A different, more straightforward methodology is to directly compare frequency lists made from differently sized corpora. Rather than judging the "effectiveness" of a list, this approach measures similarities shared between different lists. Hypothetically, doing this at increasing corpus sizes should allow one to find a size after which the variance between lists only minimally decreases. As with the previous approach, the goal here is to find a point at which the benefits of increasing size no longer outweigh the needed additional resources.

Essentially, then, all corpora of sufficient size should result in nearly the same frequency dictionary—a theory based on a strict interpretation of Zipf's law. If the appropriate criteria can be found— Sorell (2013) suggests—then this would, at last, provide a solution to the observation made by Nation (2013, p. 24) that, problematically, frequency lists tend to disagree rather drastically on both the words included

and their respective ranking.

Inspired by the computational linguistic measure of *rank distance* (Popescu & Dinu, 2008)—a method for comparing stylistic differences between texts— Sorell (2013) developed a variant of this methodology. First, he used different corpora of the same size to create multiple frequency lists, one for each corpus, ranked entirely by frequency. He then identified the percentage of words that are *not* shared between each set of two lists. Finally, he averaged these percentages to find the level of variability created at that specific corpus size. The levels of variability he found were remarkably close to each other (2013, p. 80)—despite using a wide variety of entirely different corpora (with no overlap on texts within each one). He then increased the size of each corpus and repeated the process.

In order to calculate this level of variability, Sorell used a modified version of a complex formula that he borrowed from the natural sciences, and called his resulting calculation the *Dice distance*. Though this Sørensen–Dice coefficient that he altered is widely used in botany and other fields[3] to measure similarity in areas and samples of different sizes (Dice, 1945; Sørensen, 1948), the frequency lists measured by Sorell were all purposefully of the same size. What this means is that—apparently without realizing it—his *Dice distance* was ultimately just a simple fraction:

$$\frac{\textbf{number of different words between frequency lists}}{\textbf{total size of frequency list}}$$

In essence, this measure can be accurately described as the average proportion of difference for frequency lists at that particular corpus size.

Sorell found that a stable list (about 2% variation) of the most frequent 1,000 words, or a reasonably stable list (less than 5% variation) of the most frequent 3,000, words can be created using a corpus of 50 million tokens (2013, p. 203). In other words, 1,000-word frequency lists created from different 50-million-token corpora will likely only differ by 20 words. At the 3,000-word level using the same corpus size, the lists will likely vary by less than 150 words. This is a remarkable level of similarity. Expanding the list to 9,000 words will still only yield about 4–7% variation, or 360–

[3]It has even been used in corpus linguistics studies before, primarily as a way to measure collocation (Rychlý, 2008).

630 words. Even corpora of 20 million tokens can be considered sufficient in many cases, since they will result in 3,000-word frequency lists with roughly 5% variation and 9,000-word frequency lists with less than 10% variation.

Taking a similar comparative approach, Brezina and Gablasova (2015) evaluated frequency lists created from four corpora of various sizes: the *Lancaster-Oslo-Bergen Corpus* (LOB), the *BE06 Corpus of British English* (BE06), *The British National Corpus* (BNC), and *EnTenTen16*. These corpora have respective token sizes of 1 million, 1 million, 100 million, and 12 billion. The frequency dictionary created from each corpus was, in this case, ranked by a combination of frequency and dispersion—a measure that will be discussed in more detail in the *dispersion* section of this chapter. In addition to finding the percentage of shared items between frequency lists, the researchers calculated the correlations between the rankings for each shared word. Contrary to Sorell (2013), Brezina and Gablasova considered this final comparison an important part of understanding the effect of corpus size.

The aim of this study was not to find a corpus size after which the difference was negligible, but rather to find if there was a significant difference between frequency lists made from corpora of different sizes. The study found a 78%–84% overlap between each of the 3,000–word lists. 71% of the words were shared among all four of the lists. Based on this number, Brezina and Gablasova concluded that regardless of corpus size—at least for anything larger than one million tokens—"similar results" are obtained (2015, p. 18).

This conclusion differs significantly from Sorell's, who concluded that a corpus of at least 20 million tokens (though 50 million is preferable) is needed for a stable frequency list with low variability (Sorell, 2013, p. 203). These disagreements are primarily the result of a difference in what should be considered "stable." At 71% vocabulary overlap—which is sufficient for Brezina and Gablasova—870 words were only found in one of the four lists. This is drastically higher than Sorell's threshold, which at the 3,000-word level varies in roughly 150 words. Note that Nation and Kyongho (1995) found a level of overlap similar to Brezina and Gablasova when comparing the GSL, the LOB, and the Brown corpora—a percentage of overlap that they deemed to be not particularly high. As Nation later put it, "Brezina and Gablasova are a bit too tolerant in accepting that 71% or even 78%-84% overlap is

good enough. If roughly one out of every four or five words is different from one list to another, that is a lot of difference" (Nation, 2016, p. 100).

One issue that has yet to be studied (to my knowledge) is the difference in units of counting between these two studies. Sorell made lists based on *types*, whereas Brezina and Gablasova preferred the use of *lemmas*. The exact difference between these two units is explained later under *identifying words (word family levels)* in this chapter. The effect of these different measures for comparing frequency lists created from differently sized corpora is an area that could benefit from further research.

Regardless of differences between approaches, the studies in this section have demonstrated the importance of having a sufficiently large corpus in order to create a trustworthy frequency dictionary. The next section deals with the second aspect of corpus design: the types of texts that are included.

2.1.2 Text types

Deciding on the texts that make up a corpus, and their corresponding text types, is a critical aspect of corpus design. Designing a corpus for the goal of creating a frequency dictionary needs to take the dictionary's intended purpose into account. Many corpora take a conglomerate approach, meaning that they simply amass as many texts as possible, regardless of their type. This often results in frequency lists that serve no distinct purpose.

Some published corpora—especially those designed for a specific purpose rather than "core vocabulary" or the language as a whole—do take a more strategic approach. For example, Coxhead's (2000) *Academic Word List* was created from a carefully designed corpus that used equally sized subcorpora of texts from different disciplines. This suited the purpose of the frequency list well, since it was intended to serve students from a variety of disciplines.

In order to better understand text types, some studies have sought a taxonomy that would make the selection process more objective. These seek to establish, for example, if there are distinguishable linguistic differences between an informal correspondence and a narrative work of fiction, or between a romance and a fantasy novel.

One influential attempt at this categorization was conducted by Biber (1988), who analyzed a variety of texts using large corpora to tag syntactic markers and other linguistic attributes that could potentially be used to define different types of texts. He found a series of five categories (each consisting of two opposite ends of a spectrum) in which texts varied:

1. Involved vs. informational
2. Narrative
3. Situated vs. elaborated
4. Persuasive
5. Abstract

Biber then conducted an in-depth follow-up study that found eight distinct, recurring patterns of different combinations of these categories (1995). These groupings serve as a linguistically-based taxonomy that divides texts along objective lines, rather than subjective, culturally-defined genres.

Similar but independent studies have been conducted for Somali, Korean, Nukulaelae Tuvuluan, Taiwanese, and Spanish (Biber, 1995; Jang, 1998). For each language, a unique set of text types have been identified. Yet, significantly, the texts were found to align along similar distinguishing linguistic dimensions as the English texts (Biber, 1995, p. 270).

Sorell (2013) sought to simplify Biber's eight text types into categories suitable for corpus design. He did this by identifying the similar ways that some of the text types lined up along Biber's five linguistic categories while incorporating some extra-linguistic features, such as shared contexts (e.g. predominantly spoken types). He excluded Biber's (1995) two smallest text types—"situated on-line reportage" and "involved persuasion"—deeming them impractical for corpus study and difficult to isolate (Sorell, 2013, p. 68). In doing this, he came up with four simplified text types:

1. Interactive (conversation)
2. General reported exposition (general writing)

3. Imaginative narrative (narrative writing)
4. Academic

Using his comparison method of Dice distance (described above under *corpus size*), Sorell found each simplified text type to be equidistant from the next in this order: conversation, narrative, general writing, and academic writing (2013, pp. 153–154). This allowed him to claim that his own study of vocabulary frequency—using simplified text types as a base—has "validated Biber's studies by adding a vocabulary dimension to the description of each of the key text types" (p. 201).

Similar efforts to simplify Biber's text types have also been carried out in the *Longman Grammar of Spoken and Written English* (Biber, Johansson, Leech, Conrad, & Finegan, 1999, p. 16) and the *Longman Student Grammar of Spoken and Written English* (Biber, Conrad, & Leech, 2002, p. 23)

2.1.2.1 Conversation text type Despite the importance of the conversation text type for language learners and linguistic studies, corpora of spoken language remain much smaller than more traditional corpora. This is due to the costs and resources involved with gathering large amounts of spoken data that then need to be transcribed by hand in order to be analyzed. It is true that speech recognition software has come a long way in recent years, but its rate of error remains too high for research purposes. It has been estimated that it takes 40 hours to professionally transcribe one hour of audio recording, making the task too costly (New, Brysbaert, Veronis, & Pallier, 2007, p. 662). For this reason, some researchers have begun looking at alternative sources of speech corpora, including the internet and movie subtitles (Kilgarriff & Grefenstette, 2003).

New et al. (2007) created a 50-million-token corpus of French subtitles. They divided this into four subcorpora, one for each of the type of media from which the subtitles were extracted: French films, English movies, English television series, and non-English-language European films. The reason for using French subtitles from English media is the sheer dominance of English in the film industry. In order to counterbalance the much larger sizes of the two subcorpora extracted from English media, the researchers measured word frequencies for each subcorpora separately, then averaged

them to arrive at the final frequency used for their ranked word list.

In order to test the validity of their new approach, New et al. used two different methods. First, they compared their subtitle frequency dictionary with word lists created from more traditional corpora. Second, they used lexical decision times—similar to Brysbaert and New (2009) above—to test the rankings of words on their list.

The first test found a .73 correlation with another French spoken corpus, the *Corpus de Référence du Français Parlé* (CRFP) (Delic, Teston-Bonnard, & Véronis, 2004). A closer look revealed that the majority of significant differences were caused by the monologue nature of the CRFP. This corpus was created from a large number of interviews (each asking the same questions to the interviewee), whereas movie subtitles tend to be composed primarily of people interacting in conversations. This results in more colloquial expressions having higher frequencies in the subtitle corpus. The nature of movies themselves also played a role, resulting in an overrepresentation of words related to action movies and police matters—words like *tuer* (to kill), *prison* (jail), and *armes* (weapons) (New et al., 2007, p. 665).

On the second test, New et al. found that their subtitle list's ability to predict lexical decision times was at least equally as accurate as the CRFP frequencies or those from a traditional corpus of written French (New et al., 2007, p. 675). In many cases, it actually fared much better, surprising even the researchers themselves. However, this can only be considered a preliminary finding for two reasons:

1. The sample sizes of the lexical decision task experiments were very small (234 and 240 words).
2. The study's dependence on *translated* subtitles—while understandable given the prevalence of English in the film industry—requires more thorough study before it can be considered a valid alternative. For now, these early findings seem to indicate that it may very well be.[4]

Picking up on the findings of New et al. (2007), and expanding the lexical decision task to a much larger sample size, Brysbaert and New (2009) compiled a corpus

[4]I deal with a similar limitation of the *Frequency Dictionary of Spoken Hebrew* in the *methodological challenges* section of this thesis.

of English subtitles ($\text{SUBTLEX}_{\text{US}}$) and evaluated it as part of their study. This corpus is composed of subtitles from a wide variety of American films since 1900, though a majority are from 1990, as well as a large number of American television series. They found that the subtitle frequencies were especially good at predicting the lexical decision times of short words, often surpassing the accuracy of rankings based on the many written corpora they tested. It had more difficulty explaining the response times of longer words, which are more rarely found in film than in literature. Overall, their own conclusion confirmed that of the New et al. (2007) study: word frequencies derived from subtitle corpora are as good as—and sometimes better than—those from true speech corpora.

Though both of these studies arrive at the same conclusion regarding the use of subtitles, more research is needed in this area. If, indeed, subtitles can be considered as appropriate sources for corpora of the conversation text type, their availability facilitates the creation of frequency dictionaries of spoken language—something that is otherwise too cost-prohibitive due to the difficulty of the collection medium.

Precisely because of this ease of access, subtitles provide the perfect corpus for a project that seeks to be both representative of *spoken* language and easily reproducible. This is therefore the approach taken in the present thesis to create the *Frequency Dictionary of Spoken Hebrew*. I will give a detailed description of the corpus chosen in the next chapter.

2.2 List design

Perhaps even more complex than appropriately designing the corpus from which to extract word frequencies is designing the frequency dictionary, or list, itself. Many distinct variables are involved in the process. Questions addressed in the literature deal with the difference between a general service list and a specialized list, differences in the way that a "word" is defined and measured, and different ranking criteria used, among other issues.

2.2.1 General use or specialized use

The majority of frequency dictionaries aim to describe the vocabulary of the language as a whole. They are designed to be all-encompassing so that they can serve any number of uses and scenarios. This broad nature of general-use lists is reflected in the name of the one that has historically been most widely used: West's *General Service List* (1953). Others include Nation's *BNC2000* list (2006), Browne's *New General Service List* (2014), Brezina and Gablasova's *New General Service List* (2015), and Dang and Webb's *Essential Word List* (Nation, 2016, pp. 153–167).

Another way of understanding general-use lists is that their objective is to find what is often termed the *core* vocabulary. Though not always explicitly stated, the theory behind this approach is that the language contains at its center a self-contained lexicon of essential vocabulary that is fundamental to the entire language. There are layers of frequency and increasing complexity beyond this, with regions of specialized language demarcated for specific purposes such as fields of study or geographically specific dialects. Still, this core vocabulary remains at the center of it all, and the purpose of a frequency dictionary is to identify what words fall within its boundaries. Sorell (Sorell, 2013) has evaluated a number of existing definitions of core vocabulary in the literature.

Fewer researchers have created frequency dictionaries for a more specific purpose or target audience. Specialized-use lists can be designed to only include words that belong to a specific domain, such as a discipline or trade. They can also encompass vocabulary found in a broad range of disciplines, but which are common in a specific context, such as academic texts. In this case, they usually serve as supplements to aid language learners who are already familiar with the "core vocabulary" of the language.

The most well-cited example of a specialized-use list is Coxhead's *Academic Word List* (2000), which replaced the *University Word List* (Xue & Nation, 1984) as the go-to vocabulary list for aspiring students intent on attending an English-speaking university or those entering the academic world. This could be considered a *general* academic word list, since it is intended for academic use in general, and not for a specific discipline.

More specialized frequency dictionaries include those designed for business English or medical English courses. This is sometimes designated *technical vocabulary*. Technical vocabulary is most often taught after students have mastered general-use vocabulary, and after they have some familiarity with academic vocabulary (Nation, 2016). Chung and Nation (2003, 2004) have analyzed the typical makeup of technical vocabulary. By studying specialized words in the fields of anatomy and applied linguistics, they found that a large number of technical words are also found in the language's core vocabulary, or have a general academic use as well. However, when used in a technical text, these words often take on a specialized definition that is particular to that domain.

The important takeaway is that the intended purpose of a frequency dictionary needs to be thoroughly considered, since this decision will affect both the process and outcome of the project. As already mentioned, the *Frequency Dictionary of Spoken Hebrew* is designed for Hebrew learners who wish to focus on *spoken* Hebrew. This narrows the focus from the general core vocabulary of the entire language, but it is not as restricted as a specialized list would be.

2.2.2 Identifying words (word family levels)

Another essential aspect of creating a frequency dictionary is deciding on how to measure a word. Though this may seem like a straightforward task, it requires an understanding of the theory behind the decision. Should *jump* and *jumped* be counted as two different words or just one? What about irregular inflections such as *go* and *went*? In an article aimed at raising awareness of what he calls the "word dilemma," Gardner (2007) points out that the validity of much vocabulary research hinges "on the various ways that researchers have operationalized the construct of *Word* for counting and analysis purposes" (p. 242).

The literature has generally come to accept some helpful, key terms. Beginning with the most basic measurement and progressing to the most complex, we can choose to count tokens, types, lemmas, or word families.

Consider this example sentence:

I like small dogs and medium dogs, but even her big dog can be likable.

Measuring *tokens* means simply measuring the total number of words. The example sentence contains fifteen tokens—fifteen words in total. Counting *types* refers to the number of separate and distinct words. That is, *dogs* and *dogs* are the same type, but *dog* is a different type—even a single difference makes them different types. The example sentence is composed of fourteen types. Types are often the simplest measure to use for frequency dictionaries, since they are relatively easy to identify and count.

A *lemma* includes the stem of the word and its inflected forms, but not any derived forms of the word (derived forms are usually considered a different part of speech). So *likes*, *liked*, and *liking* (the verb) are all the same lemma, but *likable* is not. This is because *likable* has the derivational affix *-able*, which turns it into an adjective. Francis et al. define lemma as "a set of lexical forms having the same stem and belonging to the same major word class, differing only in inflection and/or spelling" (Francis, Kučera, & Mackie, 1982, p. 1). The example sentence is made up of thirteen lemmas.

Finally, the term *word family* is used to describe an even more inclusive level than the lemma, though its precise definition has often varied among researchers. Bauer and Nation (1993) sought to rectify this problem through an in-depth classification of English affixes. Borrowing from Thorndike's (1941) study of English suffixes, their grouping was based on a series of eight criteria: frequency, productivity, predictability, regularity of the written form of the base, regularity of the spoken form of the base, regularity of the spelling of the affix, regularity of the spoken form of the affix, and regularity of function (Bauer & Nation, 1993, pp. 255–256). They identified seven "levels" of word families, with each successive one including a larger number of affixes, and therefore a larger number of types per word family. One very useful aspect of their particular system is that it places all the previous levels (type, lemma, etc.) within the same framework. Under their taxonomy, a level 1 word family is the same as a type, a level 2 word family is a lemma (including all regular inflected affixes), and level 7 (the highest level) consists of classical roots and affixes beyond what most speakers any longer consider separate affixes.

Nation himself suggests that for the purpose of language learning, these specific family word levels can be used simply "as a starting point as an initial framework of reference" (Nation, 2016, p. 36). That is, they are one interpretation of how to systematically count words for a frequency dictionary. These levels are based on criteria that reflect the needs of language learners, rather than on any psycholinguistic theory of how speakers' mental lexicon is arranged. Still, the idea of word families aligns closely with theoretical models that dictate morphological decomposition as a constant. These theories propose that words are often deconstructed into independent morphemes in receptive tasks and recognized that way, for example by deconstructing *jumping* into *jump* and *-ing*. At the other end of the spectrum stand theories that would place *jump* and *jumping* as separate lexical entries (Brysbaert & New, 2009, pp. 982–983).

Either way, there is strong evidence to suggest that inflected/derived forms and their base forms do affect each other in some way, suggesting that word families are a measure of a real representation in speakers' mental lexicon. In one such study, Nagy et al. (1989) explored the effect of both inflectional and derivational family frequency during a lexical decision task. They found that both types of morphological relationships lowered word recognition times, leading to the conclusion that inflections and derivational relationships are both represented in the mental lexicon, either through the grouping of related words under the same entry, or through linked entries. However, all the participants were native English speakers, so to what extent do L2 learners' lexicons reflect the same level of linking?

More recent studies have found that L2 learners' morphological knowledge and word-building ability are not nearly as developed. Ward and Chuenjundaeng (2009) conducted a study that tested the receptive ability of Thai engineering and doctoral students learning English. They were tested for their knowledge of a series of base words, together with various derived forms of the same words. They found a surprising lack of familiarity with the derived words, even when participants knew the base forms from which they were derived. Similarly, but from a productive and not receptive standpoint, Schmitt and Zimmerman (2002) found that learners could produce only a limited number of derived forms when presented with a word family headword. These results challenge the common assumption that "once the base word or even a derived word is known, the recognition of other members of the family requires little

or no extra effort" (Bauer & Nation, 1993, p. 253).

There is evidence to suggest a positive correlation between vocabulary size and morphological knowledge (Mochizuki & Aizawa, 2000; Schmitt & Meara, 1997), and between morphological knowledge and reading comprehension (Jeon, 2011). If this is the case, then using higher-level word families to create frequency dictionaries, such as Nation's *BNC2000* list (2006) or Coxhead's *Academic Word List* (2000), may not be appropriate for learners with limited knowledge of vocabulary—the very learners that many of these lists target.

When creating a frequency dictionary, then, the unit of word counting needs to suit the list's purpose and target audience. Brezina and Gablasova (2015) contend that Bauer and Nation's (1993) higher word family levels ignore the lack of transparency that exists between many of the entries that would be placed under the same word family. This is especially troublesome when used in frequency lists for language learners, whose morphological knowledge is often not well developed. Because their *New General Service List* was created for beginners, and since it is intended to aid vocabulary acquisition for both receptive and productive purposes, Brezina and Gablasova chose the lemma as their unit of measure.

Given the similar target audience, and using the same reasoning as Brezina and Gablasova (2015), I have chosen to use lemmas as the word unit to measure in creating the *Frequency Dictionary of Spoken Hebrew*.

One last note regarding Bauer and Nation's (1993) word family levels: they are specific to English. Because they are based entirely on affixation of morphemes, they cannot be readily applied to other languages. Whereas types and lemmas can be more easily understood to be equivalent across languages (with some deviation for highly agglutinative or synthetic languages), extending the concept of word family levels beyond English requires creating a similar taxonomy for each specific language. This is an area of study that has yet to receive more attention.

2.2.3 Objective design

Many frequency dictionaries—including some of the most widely-known ones—take what could be called a semi-objective approach. They begin by creating a list that bases word rankings on statistical measures such as frequency, range, and dispersion. Then, because certain words don't fit the researcher's intuitions, or because some rankings simply seem out of order, the list is tweaked here and there (Nation, 2016, p. 133).

For example, one common tweak is to group lexical sets together on a list, such as days of the week or numbers (Nation, 2016, pp. 118–119). This is true of West's GSL, resulting in a list that "brought a large element of subjectivity into the final product" (Brezina & Gablasova, 2015, p. 3). West himself laid out his argument as to why he chose to use such an approach (West, 1953, pp. ix–x). Nation has also defended the use of subjective criteria (2016, pp. 119–120).

Despite a few pedagogical advantages, however, a semi-objective approach (which is therefore also a semi-subjective approach) has important implications for reproducibility. This alone makes it unfit for the present project, since one of the primary goals of this thesis is to present an easily reproducible process that can be used to create frequency dictionaries in many different languages. Additionally, the simple fact is that by inserting subjective criteria into the list-creation process, it ceases to be based on the data directly. Rather than letting a particular corpus speak for itself, the whims and opinions of the researcher come into play. This can affect secondary tests that may be performed using the list, such as a lexical decision test.

Some frequency dictionaries that use strictly objective criteria include *Word Frequencies in Written and Spoken English* (Leech, Rayson, & Wilson, 2001), Brezina and Gablasova's *New General Service List* (2015), and Dang and Webb's *Essential Word List* (Nation, 2016, pp. 153–167). This thesis also uses exclusively objective criteria to create the *Frequency Dictionary of Spoken Hebrew*: frequency, range, and dispersion. I will now discuss each of these in turn.

2.2.3.1 Frequency Frequency can refer to either raw frequency (sometimes called absolute frequency) or normalized frequency. Raw frequency is simply the

total number of times that a specific word is attested in the corpus. Normalized frequency is a measure of how many times the item appears *for every x tokens* in the corpus. This is usually calculated to be per-million-tokens, though the exact count can vary. Using normalized frequency is more meaningful since it is easier to compare with frequencies found in other corpora.

Frequency forms the core of frequency dictionaries, and it is also their most simple measure. A word list can be created using frequency alone. However, other measures, such as range, help take into account important factors that frequency ignores.

2.2.3.2 Range Range is a measure of the number of sub-corpora—or sections of a corpus—in which the word can be found (Fries & Traver, 1960). Range is also sometimes referred to as *contextual diversity* (Brysbaert & New, 2009). To measure this, a corpus must first be divided into a series of sub-corpora. As of now, there is no real consensus on a specific way to do this, so different frequency dictionaries may contain very different range measures based on the method chosen by the researcher. Like frequency, range can also be normalized to make the number more meaningful for inter-study comparison.

Nation has gone as far as to suggest that "range figures are more important than frequency figures, because a range figure shows how widely used a word is" (Nation, 2016, p. 103). This conclusion is corroborated by studies such as that of Adelman et al. (2006), which found that range better explained the findings of lexical decision tasks by 1%–3%. Similar results were found by Ellis, who attributed better predictive power to range than to word frequency (Ellis, 2002a, 2002b).

The value of calculating range is that it provides a simple way to evaluate skewed frequency results. For example, a word may be rare overall in a language, but if it happens to be very common in only a few texts, it can still attain an inappropriately high place on the frequency list. This often occurs with specialized words that are only used by a very specific subset of the population but with high frequency. By calculating range, it becomes easy to identify these words.

The question then becomes, what to do once these words are found. How can range and frequency be used in tandem? One possibility, suggested by Nation (2016, pp.

121–122) and used by Coxhead (2000), is to decide on a minimum range, discard any words that fall below this threshold, and rank only the remaining words by frequency. This approach, however, relies on a subjective decision that becomes difficult to replicate with other corpora. The fate of words with range measures close to the cutoff point is to be either completely thrown out or kept in their original position. Shifting the word's position on the list—its rank—is more sensical, but this can quickly become messy and subjective as well. Dispersion tries to solve this problem.

2.2.3.3 Dispersion In a (simplistic) nutshell, dispersion is a combination of both frequency and range. It serves as a single number—a distributional statistic—that incorporates the benefits of both of these measures, while also allowing a list to be ranked in a methodical, objective manner.

Whereas frequency and range are found simply by counting, dispersion requires a calculation that incorporates multiple variables. Unfortunately, there is still little agreement on how best to measure dispersion. Many ideas have been proposed, such as Juilland's D (Juilland, Brodin, & Davidovitch, 1970), Carroll's D_2 (1970), Rosengren's S (1971), Lyne's D_3 (1985), and Zhang's *Distributional Consistency* (*DC*) (Zhang, Huang, & Yu, 2004). One additional measure, *Average Reduced Frequency* or *ARF* (Hlaváčová, 2006; Savický & Hlavácová, 2002) was used by Brezina and Gablasova to create the *New General Service List* (2015, p. 8) mentioned above. *ARF* takes a different approach, in that it sees the entire corpus as one long string of text rather than a series of subcorpora.

A thorough overview of all these and more dispersion measures was published by Gries, who then provided his own suggested method: *deviation of proportions*, or *DP* (Gries, 2008, 2010). Unlike earlier proposals, however, Gries' *DP* stands out as a comparatively simple calculation that takes into account some of the biggest shortcomings he identified in the others. Gries himself lists the advantages of *DP* as: flexibility to use differently sized subcorpora, simplicity, extendability to different scenarios, and appropriate sensitivity.

The idea behind *DP* is simple. For each word, it aims to find the difference between the frequency one would expect to find in each subcorpus (if the word was perfectly

evenly distributed) and the word's actual frequency. Finding the sum of the absolute values of all these distances from perfect dispersion, and then dividing the result in half (since the differences are found in both directions—higher and lower frequencies than expected), one is left with a value between 0 and 1. A *DP* of 0 represents a perfectly even dispersion, and a *DP* close to 1 means a more uneven distribution, where fewer subcorpora contain a larger load of the word's overall frequency. A *DP* of 1 is not actually possible, though Gries explains how to use a normalized value, DP_{norm}, for those who prefer a true 0–1 range (Gries, 2008, p. 419; Lijffijt & Gries, 2012). The entire equation looks like this:

$$DP = 0.5 \sum_{i=1}^{n} \left| \frac{tokens\ in\ subcorpus_i}{tokens\ in\ corpus} - \frac{frequency\ of\ lemma_x\ in\ subcorpus_i}{frequency\ of\ lemma_x\ in\ corpus} \right|$$

Because frequency does not play a direct role in calculating *DP*, Gries suggests—as a quick fix—using the product of *DP* and frequency (Gries, 2008, p. 426). This is similar to previous adjusted frequency measures such as Juilland's (1970) usage coefficient *U*. Gries goes on to explain how his proposed U_{DP} may obscure what is actually being measured. However, he does not elaborate on a better measure that could be used to rank items on a frequency dictionary. U_{DP}, therefore, continues to be used by for this purpose (Matsushita, 2012, p. 99; Sorell, 2013, p. 89). It is also the ranking measure used to create the *Frequency Dictionary of Spoken Hebrew*.

2.3 Summary and applications

This literature review has outlined some of the most pressing issues that must be considered when creating a word frequency dictionary. As we have seen, research into some of these questions has led to general agreement, in other areas the research is only beginning, and a few issues have generated much discussion but still no true consensus. This overview has laid the groundwork for the decisions that underlie the methods used to create the *Frequency Dictionary of Spoken Hebrew*.

On the matter of corpus design, I have chosen to work with a corpus of *at least* 20 million tokens, and preferably 50 million, in accordance with Sorell's (2013) find-

ings. As for the corpus' text type, because the FDOSH aims to be a list based on interpersonal interactions, it is created from a homogenous *conversation* corpus.

Though not a true core vocabulary list, the FDOSH has been created to serve as a foundation for learners of Hebrew, with the goal of reaching conversational proficiency in a wide range of areas, rather than in a specific discipline or setting. Due to the lack of large, high-quality corpora of spoken Hebrew, the FDOSH is based on a corpus of film subtitles. This approach is justified by the findings of studies that compare subtitle corpora to traditional corpora of spoken language, though this area of research is admittedly in need of further study (Brysbaert & New, 2009; New et al., 2007). The specific details of the corpus used for the FDOSH will be addressed more in depth in the following chapter.

Because the FDOSH is designed primarily for language learners, Bauer and Nation's (1993) higher word family levels were deemed inappropriate, based on evidence of learners' weak morphological knowledge and word-building ability (Brezina & Gablasova, 2015; Mochizuki & Aizawa, 2000; Schmitt & Meara, 1997; Ward & Chuenjundaeng, 2009). Instead, it uses the lemma—or level 2 in Bauer and Nation's taxonomy—in its counting and arrangement.

Finally, the FDOSH seeks to establish an entirely objective approach to frequency dictionary creation. It does this by ranking words based on a usage coefficient of Gries' deviation of proportions, or U_{DP} (Gries, 2008, 2010). This allows for all three key factors of frequency, range, and dispersion to play a role in deciding the order of the words. The FDOSH also includes normalized frequency and range for each item.

3 Methods: Creating the Frequency Dictionary of Spoken Hebrew (FDOSH)

As we have seen, the brunt of the work in high-quality vocabulary frequency list creation has focused on *English* frequency lists. Outside of the English-speaking world, and especially when dealing with less commonly taught languages, it's difficult to find well-researched word lists, if they exist at all. Why have not more educators—those who may benefit from these lists the most—decided to undertake such a task?

This need not be a project that one starts from scratch every time. Many tools already exist to make the process smoother. Still, with the rapid pace at which technology changes, these tools tend to quickly become obsolete. They are also usually restrictive to the specific preferences of their creators.

Rather than using these tools, I chose to create a series of simple scripts to create the Frequency Dictionary of Spoken Hebrew.

The two most widely-used languages for the type of data analysis involved in a word list creation are Python and R. I chose to use Python for this project. Python was designed specifically to be a very readable programming language. That is, it is easy to read and understand the purpose and flow of the code. This was one of my primary reasons for choosing to use it, since it increases the ease with which this project can be reproduced by other researchers and educators to create their own word lists. R, on the other hand, requires a deeper familiarity with the syntax and conventions of the language in order to understand.

The second characteristic that makes Python ideal for an open-source project of this nature is its mild learning curve. Though considerable effort must be made to learn any programming language, Python is widely considered good for beginners because of its simplicity. With only a rudimentary knowledge of Python, even educators or enthusiasts without a coding background will be able to modify the scripts used here to suit their own needs. To this end, I will also carefully explain what, exactly, the code does.

Though all of the code is included in this thesis (*Appendix B*), it can also be found

in an online repository at https://github.com/juandpinto/opus-frequencies. The repository can easily be cloned, or individual files can be downloaded, for modification and use. The repository uses the version control system *Git*. This means that anyone can easily look through the history of each file to see specific changes that have been made over time.

Suggestions for improvements can also be submitted through the GitHub interface, allowing for a system of cooperation and incremental innovation among researchers. The exported Frequency Dictionary of Spoken Hebrew, in its entirety, can also be found in the repository.

This thesis, then, beyond explaining the theory behind the creation of the FDOSH, aims to make the process as reproducible as possible. This section contributes to that aim by carefully documenting each step of the process.

3.1 The corpus

Before coding or analyzing anything, it's important to find an appropriate corpus to use and to become familiar with its structure. A useful place to begin is OPUS[5], which is part of the Nordic Language Processing Laboratory (NLPL), and hosted by the CSC IT center in Finland. OPUS is a database of many open, parallel corpora. These include corpora of movie and television subtitles, TED talks, web-crawled data, newspapers, and of course, books. The corpora are all free and open to the public.

The FDOSH was created using one of OPUS's corpora, the *OpenSubtitles2018*[6] corpus. The corpus can be downloaded in a variety of formats, and it can be downloaded either as *parallel* corpora or as a monolingual corpus. A parallel corpus consists of two languages interwoven together. For example, a line from the English subtitles of a movie will be paired with the same line from the French subtitles of the same movie. In theory, this means that each line of the corpus should have the same meaning in two different languages. The creation of parallel corpora has made possible many interesting and useful tools for linguistics, translators, and language learners. These

[5]http://opus.nlpl.eu
[6]http://opus.nlpl.eu/OpenSubtitles2018.php

include the open-source CASMACAT[7] project and the ReversoContext[8] tool.

For the purpose of creating a word list, a monolingual corpus is best. Note that parallel corpora will often be composed of fewer tokens than monolingual ones. This is because parallel corpora will only include movies for which the subtitles exist in both selected languages.

Though it's possible to download plain text files, the most useful format available for download is XML. Indeed, the most common file format used for large corpora is XML. The XML structure allows for nested key-value pairs, which are especially useful for parsed corpora that contain extensive metadata. XML is comparable to JSON, which we will use later to extract specific movie metadata directly from a database.

Another factor to consider is whether to download an untokenized, tokenized, or parsed corpus. An untokenized corpus contains simply the raw lines of text as found in the original subtitle files (divided into lines as they would appear while watching the movie, and labeled with the appropriate time for them to be shown):

```
<s id="49">
  <time id="T39S" value="00:03:22,280" />
שרלוק, אומר אתה מה?
  <time id="T39E" value="00:03:24,120" />
</s>
```

A tokenized corpus has further been split into individual words and punctuation, such that each word is tagged on its own:

```
<s id="49">
  <time id="T39S" value="00:03:22,280" />
  <w id="49.1">מה</w>
  <w id="49.2">אתה</w>
  <w id="49.3">אומר</w>
```

[7] http://www.casmacat.eu
[8] http://context.reverso.net/translation/

```
  <w id="49.4">,</w>
  <w id="49.5">שרלוק</w>
  <w id="49.6">?</w>
  <time id="T39E" value="00:03:24,120" />
</s>
```

A parsed corpus contains much more information for each token. The data included depends on the features of the language and on the parsing script used, but it can include things such as part of speech, syntactic role, lemma, and even specific features like gender, person, and number. Here is an example:

```
<s id="49">
  <time value="00:03:22,280" id="T39S" />
  <w xpos="ADV" head="49.3" feats="PronType=Int" upos="ADV"
  ↪  lemma="מה"
     id="49.1" deprel="obj">מה</w>
  <w xpos="PRON" head="49.3" feats="Gender=Masc|Number=Sing|Person=2|
     PronType=Prs" upos="PRON" lemma="הוא" id="49.2"
↪  deprel="nsubj">אתה</w>
  <w xpos="VERB" head="0"
  ↪  feats="Gender=Masc|HebBinyan=PAAL|Number=Sing|
     Person=1,2,3|VerbForm=Part|Voice=Act" upos="VERB"
↪  misc="SpaceAfter=No"
     lemma="אמר" id="49.3" deprel="root">אומר</w>
  <w xpos="PUNCT" head="49.3" upos="PUNCT" lemma="," id="49.4"
     deprel="punct">,</w>
  <w xpos="NOUN" head="49.3" feats="Gender=Masc|Number=Sing"
  ↪  upos="NOUN"
     misc="SpaceAfter=No" lemma="שרלוק" id="49.5"
↪  deprel="obj">שרלוק</w>
  <w xpos="PUNCT" head="49.3" upos="PUNCT" misc="SpaceAfter=No"
  ↪  lemma="?"
     id="49.6" deprel="punct">?</w>
```

```
  <time value="00:03:24,120" id="T39E" />
</s>
```

All of the data used to create the FDOSH came from a monolingual parsed corpus of Hebrew. The parsing was all done automatically—a process that will be discussed in the *automatic parsing* section of the next chapter.

3.2 Cleaning the corpus

Unlike many corpora, the OpenSubtitles2018 corpus as presented in its downloadable form has already undergone significant preprocessing by the OPUS team.(Lison & Tiedemann, 2016) This is good news, since data cleaning is often the most laborious part of the process. However, there is one issue that must be addressed before the corpus can be used to create a word list: deduplication.

The files inside the downloaded folder are organized as follows:

```
Zipped folder in GZ format
    Folder for year X
        Folder for movie A
            Zipped XML in GZ format
            Zipped XML in GZ format
            Zipped XML in GZ format
        Folder for movie B
            Zipped XML in GZ format
            Zipped XML in GZ format
    Folder for year Y
        Folder for movie C
            Zipped XML in GZ format
        Folder for movie D
            Zipped XML in GZ format
            Zipped XML in GZ format
            Zipped XML in GZ format
```

```
    Folder for movie E
        Zipped XML in GZ format
        Zipped XML in GZ format
Folder for year Z
      Folder for movie F
          Zipped XML in GZ format
          Zipped XML in GZ format
```

This organization is straightforward, except for the fact that there are multiple XML files for each movie. The subtitle files that OPUS has collected, parsed, organized, and made available for mass download were all obtained from the *Open Subtitles*[9] project (hence the name of the corpus). Because this is a database where users can upload the subtitle files they extract from their own movie collection, there are often multiple uploads for the same movie. For our purposes, this results in movies that can have anywhere from a single subtitle file to dozens of them. Unfortunately, though the tokens in the files themselves are usually the same (with only minor variations in the XML metadata), this is not always true. Some few variations seem to be different and independent translations.

Part of cleaning the corpus, then, entails getting rid of these duplicates. As a means of simplifying the entire process, I chose simply to use the first file in each movie folder. I've included the short Python script for this in its entirety in *Appendix B.3*. However, I will here explain what it does in detail so that it can be easily modified to fit different circumstances.

The script first makes a copy of the entire folder structure in the original downloaded (and unzipped!) corpus into a new directory. It then finds the first XML file in each movie folder and copies it into the appropriate place in the new folder structure. This means that it doesn't delete or otherwise change the files in the original corpus in any way.

The first block of code imports necessary modules that are used later in the script (`shutil` and `os`). Lines 7 and 8 define where the original corpus is (`source`), and where the new one will be placed (`destination`).

[9] https://www.opensubtitles.org/

```
import shutil
import os

source = '../OpenSubtitles2018_parsed'
destination = './OpenSubtitles2018_parsed_single'
```

Next, a single line of code copies all directories and subdirectories into their new location.

```
shutil.copytree(source, destination,
↪  ignore=shutil.ignore_patterns('*.*'))
```

Lastly, we create a variable that holds all the XML files located in each movie folder, trim the list to just one, and copy that one into its new location. This process is carried out for one movie folder at a time. The originals are left untouched.

```
for dirName, subdirList, fileList in os.walk(source):
    for fname in fileList:
        if fname == '.DS_Store':
            fileList.remove(fname)
    if len(fileList) > 0:
        del fileList[1:]
        src = dirName + '/' + fileList[0]
        dst = destination + dirName[27:] + '/'
        shutil.copy2(src, dst)
```

With a newly organized version of the corpus, it's now possible to begin the process of reading and processing data. At this stage, I took some time to gather metadata for all the movies in the corpus in order to identify movies that were originally filmed with Hebrew as their primary language (as opposed to translated subtitles). Because I ultimately decided against this approach for the creation of the FDOSH, I will skip that step here. However, a description of the entire process will be discussed later under *using original-language movies exclusively*.

3.3 Extracting data

Before calculating any measures such as frequency, individual lemmas must be extracted from the XML files in the downloaded corpus. There are two ways to go about this. Because XML consists of nested tags and key-value pairs, a dedicated XML parsing tool can be used to extract specific information. In this case, we would be creating a list of all *values* in the `'lemma'` *key* within each `<w>` *tag*. The value that corresponds to the `'lemma'` tag below for the word אומר is אמר.

```
<w xpos="VERB" head="0"
↪  feats="Gender=Masc|HebBinyan=PAAL|Number=Sing|
   Person=1,2,3|VerbForm=Part|Voice=Act" upos="VERB"
↪  misc="SpaceAfter=No"
   lemma="אמר" id="49.3" deprel="root">אומר</w>
```

A different approach is to use *regular expressions* to search for a specific string of characters and extract every instance of that string. This is a more brute-force approach, since it ignores the structure of the XML file and treats it all simply as raw text. To find a lemma, a very simple regular expression is sufficient: `lemma="[א-ת]+"`. This will search for any instance of the characters `lemma="`, followed by a combination of any number of Hebrew letters (at least one), followed by the character `"`.

Despite the existence of various Python modules for parsing XML files, I found a simple search using regular expressions to be more efficient for various reasons. First, not all elements in the parsed corpus contain *lemma* attributes. Second, punctuation and non-Hebrew words are often lemmatized. This means that even after extracting all the *lemma* values in a file, I would still need to use regular expressions to search through the results and delete any that contain non-Hebrew characters. I chose instead to skip the XML parsing step altogether.

I will now explain the code in the script used to create the FDOSH. As with the other code, the entire script in its entirety can be found in *Appendix B.1.*

After importing necessary packages and initializing variables, two functions near the beginning of the script serve to open a file and extract a list of lemmas from it.

```
# Open XML file and read it.
def open_and_read(file_loc):
    with gzip.open(file_loc, 'rt', encoding='utf-8') as f:
        read_data = f.read()
    return read_data
```

```
# Search for lemmas and add counts to "lemma_by_file_dict{}".
def find_and_count(doc):
    file = str(f)[40:-3]
    match_pattern = re.findall(r'lemma="[א-ת]+"', doc)
    for word in match_pattern:
        if word[7:-1] in lemma_by_file_dict:
            count = lemma_by_file_dict[word[7:-1]].get(file, 0)
            lemma_by_file_dict[word[7:-1]][file] = count + 1
        else:
            lemma_by_file_dict[word[7:-1]] = {}
            lemma_by_file_dict[word[7:-1]][file] = 1
```

We then run both of these functions for each XML file in the corpus directory (defined earlier in `corpus_path`).

```
for dirName, subdirList, fileList in os.walk(corpus_path):
    if len(fileList) > 0:
        total_files_int = total_files_int + 1
        f = dirName + '/' + fileList[0]
        find_and_count(open_and_read(f))
```

The `find_and_count()` function finds each instance of the string described above using a regular expression, then adds the Hebrew part of the string—the lemma itself—to a dictionary. The dictionary is named `lemma_by_file_dict`, and its structure looks like this:

```
'lemma': {'path of file': 'frequency of lemma in file'}
```

A dictionary is at its core a list of *key:value* pairs. Much like an actual dictionary consists of words and their definitions, this dictionary's keys are made up of all the individual lemmas found by our search. For each lemma, the value is another dictionary—making it a nested dictionary, or a dictionary within a dictionary. The keys for each inner dictionary are the paths of all the XML files (movies) that the lemma appears in, and the value of each is an integer that represents how many times that lemma appears in that file (frequency).

After the script reads each file, it returns a complete dictionary. Here is a sample:

```
'ב': {
        '/he/0/5753574/6853341.xml': 168,
        '/he/0/3607000/5764778.xml': 94},
'פרק': {
        '/he/0/5753574/6853341.xml': 3},
'קודם': {
        '/he/0/5753574/6853341.xml': 6,
        '/he/0/3607000/5764778.xml': 2,
        '/he/0/1278351/3777598.xml': 1}
```

Throughout the rest of the script, this nested dictionary serves as the basis for all of the calculations needed.

3.4 Calculations

For each lemma, the FDOSH includes three measures: frequency, range, and dispersion. It uses dispersion as its sorting value. Though the theoretical underpinnings of each have already been discussed in the *objective design* section of the previous chapter, I will here give a brief reminder of what each measure is and explain how it is calculated. Range will be addressed afterward in the (*sort and export*)[#sort-and-export] section, since the script calculates it on the spot as the list is created.

3.4.1 Frequency

Since we've already calculated the frequency of each lemma for each individual file, calculating total frequency per lemma is straightforward. The script simply creates a new dictionary, `lemma_totals_dict`, and adds to it every lemma in the corpus as its keys, with the corresponding value being a sum of the frequencies in all files for that lemma. In other words, {'lemma1':'frequency1', 'lemma2':'frequency2', . . . }

```
119 for lemma in lemma_by_file_dict:
120     lemma_totals_dict[lemma] =
↪   sum(lemma_by_file_dict[lemma].values())
```

This returns Using the short example given above, this would result in the following dictionary:

```
262:'ב',
3:'פרק',
9:'קודם'
```

3.4.2 Dispersion (U_{DP})

Dispersion is more complicated. In theory, it should provide a single quantifiable measure that incorporates both frequency and range, and which can then be used to sort the word list. The model of dispersion I have chosen to follow for this project is a usage coefficient of Gries' deviation of proportions, or U_{DP} (Gries, 2008, 2010).

In order to calculate U_{DP} for lemma_x, we must first make two calculations for each file in the corpus (file_i): the lemma's *expected frequency* if it were perfectly distributed, and its *observed frequency*—or its actual frequency.

$$\textbf{expected frequency} = \frac{tokens\ in\ file_i}{tokens\ in\ corpus}$$

$$\textbf{observed frequency} \;=\; \frac{frequency\ of\ lemma_x\ in\ file_i}{frequency\ of\ lemma_x\ in\ corpus}$$

We must then subtract the lemma's observed frequency from its expected frequency, which will return a value between -1 and 1. We can normalize this result by finding the absolute value. Now the closer the result is to 0, the closer that lemma's frequency is in that particular file to what we would expect if it were perfectly distributed throughout the corpus. A higher number (closer to 1), would indicate a heavier load in that file that we would expect.

By performing this calculation for every file in the corpus, adding them all together, and dividing the result by two (since we're using the absolute value and are therefore adding values originally in both directions), we now have Gries' *DP*. Where `n` is the number of files:

$$\textbf{DP} \;=\; 0.5 \sum_{i=1}^{n} |\ \text{expected frequency} \;-\; \text{observed frequency}\ |$$

A *DP* of 0 represents a perfectly even dispersion, and a *DP* close to 1 means a more uneven distribution, where fewer files contain a larger load of the lemma's overall frequency. A *DP* of 1 is not actually possible.

Gries' usage coefficient, or U_{DP}, is an attempt to make this number more useful. *DP* is first subtracted from 1 and the result is multiplied by the lemma's total frequency. The full equation for U_{DP} is as follows:

$$\left(1 - 0.5 \sum_{i=1}^{n} \left| \frac{file_i\ tokens}{total\ tokens} \;-\; \frac{frequency_x\ in\ file_i}{total\ frequency_x} \right| \right) \times total\ frequency_x$$

In order to calculate this, the script must first find the number of tokens in each file. Like before, this is done by creating a dictionary, `token_count_dict`, which contains the *key*:*value* pairs of *file*:*tokens*. Since we already have a dictionary with the number of times that each lemma appears in each file, `lemma_by_file_dict`, we don't need to open and read the files again. Instead, we can add the values in this

dictionary and rearrange them into what we want.

```
123 for lemma in lemma_by_file_dict:
124     for file in lemma_by_file_dict[lemma]:
125         token_count_dict[file] = token_count_dict.get(
126             file, 0) + lemma_by_file_dict[lemma][file]
```

We also need to know the total number of tokens in the entire corpus. This is a simple matter of adding all the values in the `token_count_dict` dictionary. The final count is saved into an integer variable, `total_tokens_int`.

```
129 for file in token_count_dict:
130     total_tokens_int = total_tokens_int + token_count_dict.get(file,
    ↪  0)
```

Finally, the script uses all these measures to calculate DP and then U_{DP} for each lemma, and places them into their respective dictionaries, `lemma_DPs_dict` and `lemma_UDPs_dict`.

```
140 # Calculate DPs
141 for lemma in lemma_by_file_dict.keys():
142     for file in lemma_by_file_dict[lemma].keys():
143         lemma_DPs_dict[lemma] = lemma_DPs_dict[lemma] + abs(
144             (token_count_dict[file] /
145              total_tokens_int) -
146             (lemma_by_file_dict[lemma][file] /
147              lemma_totals_dict[lemma]))
148 lemma_DPs_dict = {lemma: DP/2 for (lemma, DP) in
    ↪  lemma_DPs_dict.items()}
149
150 # Calculate UDPs
151 lemma_UDPs_dict = {lemma: (1-DP)*lemma_norm_dict[lemma] for (lemma,
    ↪  DP) in
```

```
152                         lemma_DPs_dict.items()}
```

With these values all calculated for each lemma, the only thing left is to sort and create the final list.

3.5 Sort and export

In order to ensure that the words on the list do not have an abnormally high frequency in some subcorpora (movies) and are nearly absent in others, some have suggested setting a minimum range or dispersion and discarding words below this threshold (see the *objective design* section in the previous chapter).

Rather than setting an arbitrary bar, the FDOSH is sorted entirely by U_{DP}. This *modus operandi* ensures that the order of words itself—not just which words make it onto the list and which don't—is decided by a combination of both relevant measures: frequency and dispersion. This approach also has the added benefit of being entirely objective.

Since we've already calculated the U_{DP} for each lemma, sorting the list is simple.

```
160 UDP_sorted_list = [(k, lemma_UDPs_dict[k]) for k in sorted(
161     lemma_UDPs_dict, key=lemma_UDPs_dict.__getitem__,
162     reverse=True)]
```

A final table is then created (using a list of tuples, `table_list`), with each line consisting of a lemma, its overall frequency, its range, and its U_{DP}. This table is already sorted by U_{DP} as it's being created.

Because the script has not calculated range by this point, it must do so on the spot as it's entering each lemma into the table. It does this with a simple dictionary comprehension that quickly counts the number of files included in the `lemma_by_file_dict`. Here is the resulting code:

```
i = 0
for k, v in UDP_sorted_list[:list_size_int]:
    i = i + 1
    table_list.append((k,
                       i,
                       '{0:,.2f}'.format(v),
                       '{0:,.2f}'.format(lemma_norm_dict[k]),
                       '{0:,.2f}'.format(sum(1 for count in
    ↪  lemma_by_file_dict[k].values() if
                                             count > 0) /
                                          total_files_int * 100)))
```

Lastly, now that everything is organized into a table, the script opens (or creates, if it doesn't yet exist) a TSV file, writes a header line into it (`LEMMA RANK DISPERSION FREQUENCY RANGE`), and exports the entire table into the file. It then closes it to clear the computer's memory cache.

```
result = open('./export/frequency-dictionary.tsv', 'w')
result.write('LEMMA\tRANK\tDISPERSION\tFREQUENCY\tRANGE\n')
for i in range(list_size_int):
    result.write(str(table_list[i][0]) + '\t' +
                 str(table_list[i][1]) + '\t' +
                 str(table_list[i][2]) + '\t' +
                 str(table_list[i][3]) + '\t' +
                 str(table_list[i][4]) + '\n')
result.close()
```

The list is now complete. The next section will explore the list itself more in-depth.

4 The FDOSH: A vocabulary list of conversational Modern Hebrew

The Frequency Dictionary of Spoken Hebrew in its entirety can be found as an electronic supplement to this thesis (in CSV format) or at the following GitHub repository: *https://github.com/juandpinto/opus-frequencies*. It contains the most common 5,000 lemmas of conversation Modern Hebrew, as found in the *OpenSubtitles2018* corpus. A sample of the first 1,000 lemmas is included in *Appendix A*.

For discussion purposes, a small sample of the first 30 items is here presented.

Table 1: Sample of the first 30 items on the FDOSH.

LEMMA	RANK	DISPERSION	FREQUENCY	RANGE
הוא	1	114,718.51	121,008.92	99.99
ה	2	47,244.93	50,841.12	100.00
את	3	32,811.28	35,337.28	99.92
ל	4	27,415.19	29,102.77	99.97
לא	5	24,888.86	27,213.76	99.94
זה	6	23,817.89	26,418.69	99.96
ב	7	23,081.75	24,839.48	99.98
של	8	18,214.68	20,088.89	99.97
ש	9	18,203.83	20,028.64	99.95
היה	10	11,861.33	13,312.52	99.91
מה	11	10,879.07	12,192.80	99.87
ו	12	8,711.82	9,840.85	99.93
על	13	8,246.82	9,119.70	99.93
כול	14	6,062.08	6,842.01	99.90
ידע	15	5,328.40	6,205.85	99.69
כן	16	5,011.86	6,232.26	99.46
מ	17	4,871.00	5,479.15	99.89
יש	18	4,840.57	5,519.12	99.81
עשה	19	4,180.99	4,941.68	99.66
אבל	20	4,052.79	4,757.33	98.86

LEMMA	RANK	DISPERSION	FREQUENCY	RANGE
טוב	21	3,954.48	4,891.35	99.61
רצה	22	3,949.30	4,671.67	99.41
אם	23	3,846.61	4,444.59	99.68
עם	24	3,756.06	4,333.17	99.71
אמר	25	3,515.24	4,128.07	99.39
אז	26	3,370.31	4,052.24	99.41
סדר	27	3,197.62	4,305.52	98.33
צריך	28	2,862.13	3,501.64	99.18
רק	29	2,543.93	2,996.30	99.65
חשב	30	2,511.54	3,021.85	99.09

Besides each lemma and its respective rank on the list, the FDOSH includes three pieces of information: frequency, range, and U_{DP}. Frequency, in this case, is not raw frequency—the total number of times the lemma appears in the corpus—but rather how many times the lemma appears for every million tokens in the corpus. Using this normalized frequency measure makes the number more meaningful since it aims to reflect the per-million count of all spoken Hebrew, not just the OpenSubtitles2018 corpus. It also makes it easier to compare frequencies with those found in other corpora. The range is the number of sub-corpora—or, in this case, movies—the lemma appears in.

The most important piece of information the list provides, however, is dispersion, which acts as the ranking measure for the FDOSH and is discussed more in-depth in the *dispersion* section of the previous chapter.

The percentage of the corpus that is covered by the first n items on the list is referred to as coverage. This is a simple matter of finding the total number of tokens in the corpus, and dividing from it the sum of all the *raw* frequencies from the first n items.

For example, the sum of the frequencies of the first 20 lemmas in *Table 1* (84,656,819) divided by the total size of the corpus (193,755,220) is 0.436926649. In theory, this means that by knowing just the first 20 lemmas on the FDOSH one would be able to understand 43.7% of the words in the entire OpenSubtitles2018 corpus! That is

a clear example of the power of Zipf's Law (see the *introduction* for more on Zipf's Law).

Table 2 presents a listing of some important coverages provided by different amounts of lemmas on the FDOSH.

Table 2: Breakdown of coverage percentages.

n Lemmas	Frequency Sum	÷ Corpus Size	= Coverage
374	135,767,644	193,755,220	70%
939	155,016,588	193,755,220	80%
4,246	174,380,519	193,755,220	90%
13,758	184,067,666	193,755,220	95%

The entire FDOSH consists of 5,000 lemmas. This number was chosen in order for it to include the required items for 90% coverage, while also making it an even factor of 1,000. In its entirety, the FDOSH covers 90.8% of the corpus from which it is created.

4.1 Challenges and future direction

Throughout the course of this project, I have encountered several issues that are worth discussing. Some of these are questions that require further study in order to address adequately. Others are technical issues related to the complex task of pre-processing and parsing the corpus—something not directly dealt with in this thesis. Others yet are simple suggestions that I simply did not have time to implement given this project's time constraints. And finally, there are limitations that are the inevitable result of the tools at hand.

I have divided all of these issues into two categories: methodological challenges of a bigger nature, and functional challenges of a more limited scope.

4.1.1 Methodological challenges

One of the more obvious issues of this project is the use of a corpus of movie subtitles as a substitute for a corpus of true conversational language. This issue in a way forms the backbone of the FDOSH, and it is at the heart of what this project is all about. Though I discuss several points related to this in the *Background* section of this thesis, I will here discuss some of its implications for future work.

4.1.1.1 Ideal vs. practical corpora The use of a subtitle corpus has both positive and negative aspects. As described in the literature review, the early research that has been done on the topic indicates that movie subtitles share many features with spontaneous, spoken language (Brysbaert & New, 2009; New et al., 2007). This includes a high level of correlation between the two, as well as a strong ability to predict the outcomes of a lexical decision task.

One especially positive aspect of subtitle corpora is their accessibility. Thanks to the efforts of organizations such as *http://opensubtitles.com* and OPUS[10], very large corpora are available to the public for free. And they already come pre-processed, as an additional incentive for the time-constrained researcher.

This free and open nature makes subtitle corpora excellent tools for research in languages that don't yet have large, high-quality corpora of spoken language. Though advances in technology are rapidly making this type of data-collection more accessible, the costs remain too high for many less-commonly taught languages as of now. This is largely due to the arduous process of transcribing audio recordings.(Izre'el, 2004)

An ideal corpus for this sort of task would consist of many millions of tokens of recorded, transcribed, and parsed spontaneous spoken language. Several attempts have been made to create a corpus of this nature in Hebrew.

The most prominent of these is the *Corpus of Spoken Israeli Hebrew* (CoSIH)[11], created at Tel Aviv University between 2000 and 2002 (Izre'el, Hary, & Rahav, 2001).

[10] http://opus.nlpl.eu
[11] http://cosih.com/

Designed and initiated by a team of distinguished scholars, it unfortunately ran out of funding long before its goals were met. The CoSIH website (*http://cosih.com/*) makes available to the public a total of 13.5 hours of recorded Hebrew, with just over five hours of it having been transcribed.

Though a few publications have used data from CoSIH, these have been primarily methodological studies for the design of the project itself (Amir, Silber-Varod, & Izre'el, 2004; Izre'el et al., 2005; Mettouchi, Lacheret-Dujour, Silber-Varod, & Izre'el, 2007). At least one dissertation, by Nurit Dekel, uses data exclusively from CoSIH. Her entire corpus consists of 44,000 tokens (Dekel, 2010, p. 7).

Other corpora of spoken Hebrew include the Haifa Corpus of Spoken Hebrew (Yael, 2014) and the Hebrew CHILDES corpus (Albert, MacWhinney, Nir, & Wintner, 2013; Gretz, Itai, MacWhinney, Nir, & Wintner, 2015). The first consists of 17.5 hours of audio recordings, along with a limited selection of transcribed text. The latter is a collection of recordings of interactions between adults and children, comprising a total of 417,938 transcribed tokens. The CHILDES corpus is unique in that the transcriptions are provided using a Latin-based phonemic transliteration. This was done in order to avoid many of the textual ambiguities of using the Hebrew script, which are addressed below under *functional challenges*.

Though ideal in some ways, these corpora remain far too small to be effectively used for the creation of frequency lists. Even combined into a single corpus (which would introduce a series of new issues to solve), the total size would not be bigger than two million tokens. As discussed earlier in this thesis, Sorell (2013) provides evidence to suggest that a corpus of 20–50 million tokens is the minimum for a stable word list.

Are movie and television subtitles a suitable substitute for spontaneous, spoken language? Early studies suggest it is at least adequate, but much more research is needed to answer this question definitively. For now, it remains as a practical and appealing option.

4.1.1.2 Using original-language movies exclusively One of the potential downsides of using the OpenSubtitles2018 corpus is that it includes all subtitles of a specific language, even *translated* subtitles from movies filmed in other languages.

The question is, does a translated script represent true conversational language as faithfully as an original script?

This is a question that requires more research in order to answer satisfactorily. Though translated subtitles don't need to try to approximate the utterance length and visual cues that a dubbed script does, their quality still largely depends on the skills of a translator. Most importantly, a translation may not accurately reflect the register of the original, no longer serving as a representation of conversational language. Again, these are important points to consider.

One solution is to simply use movies that were originally filmed in the target language of the corpus. Another possibility is to calculate frequency measures for original and translated subtitles separately, then average them. This latter approach was used by New et al. (2007). In either case, the first step is to extract the subtitle files that represent the original language of the movie, in this case Hebrew. In theory, each XML file in a monolingual *OpenSubtitles2018* file should contain a tag that identifies the original language of the movie (Lison & Tiedemann, 2016). In practice, I found that the overwhelming majority of the files contained an empty `<lang>` tag instead. Luckily, there is a way to obtain the desired metadata for each movie in the corpus.

This can be done with a script that uses an application programming interface (API) to fetch specific information from an online movie database. The name of each movie folder in the corpus, which is simply a series of numbers, corresponds to that movie's IMDb identifier, which is a unique ID registered with the Internet Movie Database[12]. This makes the process relatively easy, as we simply need to query the database using this ID to receive all of the movie's metadata.

Though IMDb does provide their own API, I decided instead to use an API created for the Open Movie Database (OMDb)[13]. This API can be used free-of-charge, but it has a 1,000 movie limit per day. Since the OpenSubtitles2018 Hebrew corpus contains nearly 50,000 movies, I decided instead to pay for a daily limit of 100,000 movies. This only requires a $1.00 donation for each month that one is registered to use the OMDb API.

[12]http://www.imdb.com/
[13]http://www.omdbapi.com/

Once an API key is obtained, a script can be written to obtain the information desired for every movie all at once. In this case, we want to know the original language(s) for each movie.

This script in its entirety is found in Appendix B.2. It uses an imported Python wrapper for the API, written by Derrick Gilland[14], which can be found at https://github.com/dgilland/omdb.py. This package can be installed through PIP by entering `pip install omdb` into the command line.

For practical purposes, the script requires one to enter a specific year (or, more accurately, corpus folder name). If desired, an asterisk can act as a wildcard: `python OMDb-fetch.py 1988` will fetch data for movies from 1988, while `python OMDb-fetch.py 198*` will do it for all movies in the 1980s. In order to fetch data for all movies in the database at once, use `python OMDb-fetch.py *`. I don't recommend this, however, since it may overload the server and cause the script to time out.

I also found that, unfortunately, OMDb does not contain every movie in its database. However, these mystery movies were few.

The script begins by creating a list of all movie directory paths for the desired year.

```
15 for name in glob.glob(
16         './OpenSubtitles2018_parsed_single/parsed/he/' + year +
           ↪  '/*/'):
17     IDs.append(name)
```

Each item in the list is then trimmed to include only the name of the movie folder, which is *almost* equivalent to the IMDb ID.

```
20 IDs = [os.path.basename(os.path.dirname(str(i))) for i in IDs]
```

In order to make the IDs match those in the database, additional zeros must be added to the beginning until they are seven digits long.

[14]https://github.com/dgilland

```
23 for i in IDs:
24     while len(i) < 7:
25         IDs[IDs.index(i)] = '0' + i
26         i = '0' + i
```

The list is then sorted numerically in order to more easily interpret the results: `IDs.sort()`.

The API key is set in line 32, but be sure to replace `906517b3` with your own key, which can be obtained at http://www.omdbapi.com/.

```
32 omdb.set_default('apikey', '906517b3')
```

The script then prints a table header, fetches the title, year, and language(s) for each movie, and prints the results directly into the computer terminal.

```
35 print('# ' + year + '\n' +
36       'IMDb ID\tTitle\tYear\tLanguage(s)')
```

```
39       for i in IDs:
40           doc = omdb.imdbid('tt' + i)
41           print('tt' + i + '\t' +
42                 doc['title'] + '\t' +
43                 doc['year'] + '\t' +
44                 doc['language'])
```

Using a simple search program that allows for extraction of specific lines, such as those labeled with the language `Hebrew`, one can make a list of all the subtitle files that represent the original primary language of the movie. I used the open-source coding program *Atom*[15] to do this, though many options exist.

[15]https://atom.io

I modified the main script to use only movies from this list. The instructions for how to do this are included in the comments within the main script itself, which can be found in *Appendix B.1.

In the end, however, I found that the total token count for this entire mini-corpus of original Hebrew subtitles was only 615 thousand. This was well below my minimum goal of a 20-million-token corpus. In comparison, the entire Hebrew *OpenSubtitles2018* corpus that I used (with translated and original language subtitles) contains over 194 million tokens. I have explained how to use the scripts for this purpose so that they can be used for languages that have sufficient original subtitles. The *Frequency Dictionary of Spoken Hebrew*, however, is created using the entire corpus. As I mention in the *conversation text type* section of the literature review, the findings of a study by New et al. (2007) suggest that translated subtitles may be a valid alternative, but more research is needed in this area.

4.1.2 Functional challenges

A quick scan of the FDOSH reveals some notable items. Some of these are mere quirks of the automatic parser, while others are the result of ambiguities.

For example, the very first lemma on the list is a bit unexpected. "הוא" is certainly not the most common lemma in Modern Hebrew. A quick look at some of the files in the corpus, however, reveals that all pronouns are grouped under this lemma. That is, אתה (you), היא (she), and אנחנו (we), just to name a few, are parsed as belonging to the lemma "הוא." Considering how common pronouns are in the majority of spoken dialogue (in many languages), its place at the top of the list ceases to be a surprise.

Another thing to note is that verbs are all listed in their traditional third-masculine-singular past conjugation. The first verb on the list is "היה"—a lemma referring to all forms of the verb להיות, including the infinitive. The same is true of "ידע" (item 19) and "דיבר" (item 60).

Many of the most common lemmas on the FDOSH are prepositions. Note that even inseparable prepositions, such as -ה and -ב are considered independent lemmas by the parser, and are listed respectively as the lemmas "ה" and "ב".

Other issues, however, are more difficult to explain.

4.1.2.1 Textual ambiguity of Hebrew orthography

The flexible spelling conventions of Hebrew are at the root of many of the problems with the FDOSH. For example, דִּבֶּר *he spoke* can be written as either דיבר ("full spelling") or דבר ("defective spelling"). There is also a noun, דָּבָר *thing*, that looks identical to the verb's defective spelling (דבר). Though the difference is usually clear from context, the automatic parser has some difficulty with this orthographic ambiguity.

The lemma "דבר" (item 27) includes instances of both the verb and the noun, which are completely unrelated. A simple search through the corpus reveals multiple examples of the noun דבר tagged with `lemma="דבר"`:

```
<w xpos="NOUN" head="579.3" feats="Gender=Masc|Number=Sing"
 ↪  upos="NOUN" lemma="דבר" id="579.2" deprel="nsubj">דבר</w>

<w xpos="NOUN" head="200.11" feats="Gender=Masc|Number=Plur"
 ↪  upos="NOUN" lemma="דבר" id="200.12" deprel="obj">דברים</w>
```

We also find plenty of examples of the verb with the same lemma tag:

```
<w xpos="VERB" head="0"
 ↪  feats="Gender=Fem|HebSource=ConvUncertainHead|Number=Sing|Person=3|Tense=Past"
 ↪  upos="VERB" lemma="דבר" id="2346.4" deprel="root">דברה</w>

<w xpos="VERB" head="0"
 ↪  feats="Gender=Fem,Masc|Number=Plur|Person=1|Tense=Past"
 ↪  upos="VERB" lemma="דבר" id="1270.2" deprel="root">דברנו</w>

<w xpos="VERB" head="0"
 ↪  feats="Gender=Fem,Masc|Number=Plur|Person=3|Tense=Past"
 ↪  upos="VERB" lemma="דבר" id="368.4" deprel="root">דברו</w>
```

A different lemma, "דיבר" (item 61), is the expected lemma for the verb since it follows the standard third masculine plural conjugation. Interestingly, however, the parser applies this lemma only to attestations of the word with an inserted *yod*, or with a *mem* or *lamed* prefix (present tense or infinitive). All other instances are parsed as the lemma "דבר." Though unexpected and simply wrong, at least this issue is consistent.

```
<w xpos="VERB" head="840.4"
↪  feats="Gender=Fem,Masc|HebBinyan=HITPAEL|Number=Plur|Person=1|Tense=Past"
↪  upos="VERB" lemma="דיבר" id="840.16" deprel="conj">דיברנו</w>

<w xpos="VERB" head="1451.12"
↪  feats="Gender=Masc|HebBinyan=PIEL|Number=Sing|Person=1,2,3|VerbForm=Part|Voice=A
↪  upos="VERB" lemma="דיבר" id="1451.20" deprel="obl">מדבר</w>
```

To complicate matters more, we also find the unexpected lemmas "דיברה" (item 1184), "שדיבר" (item 2588), and "שדיברה" (item 4106). Based on their context, these should clearly be parsed as two separate lemmas, "ש" and "דיבר."

These are just a few among many examples of the difficulties encountered by the automatic parser. Though the parsing was carried out by the OPUS team as part of the corpus's pre-processing stage, it is valuable to at least have an idea of how it works its magic. I will here explain the basics of the process and some of the implications entailed.

4.1.2.2 Automatic parsing Automatic parsing refers to the process of having a computer program create a syntactic tree for a corpus of natural language. Natural language, as opposed to artificial or constructed language, is notoriously complex in its structure. Natural language processing (NLP) is an entire field of research, currently at the forefront of computer science. Parsing can serve many purposes, from theoretical linguistic research to machine translation or even the creation of artificial intelligence assistants such as Siri or Alexa. For our purposes, a parsed text is important in order to use lemmas as the word family level for the FDOSH. This decision is discussed under *identifying words (word family levels)* in this thesis.

Two distinct types of syntactic parsers exist, constituency parsers and dependency parsers. These are based on the two respective linguistic theories of syntax, constituent grammar (sometimes referred to as phrase structure grammar) and dependency grammar.

Constituent grammar is the classic syntax tree structure taught in introductory-level linguistics classes. It is essentially a theory of the logic structure of language as a whole. Dependency grammar is a competing theory that treats words as more directly interconnected to each other. A thorough description of these ideas is outside the scope of this thesis and is not pertinent to the project. What is important to know is that dependency grammar, and thus dependency parsers, have played an important role in the advancement of NLP and computational linguistics as a whole. The term "automatic parser", therefore, most often refers to an automatic *dependency* parser.

Some parsers proceed in a two-step process of morphological tagging (part of speech) and then dependency parsing (syntactic role and conjugations). In all cases, tokenization must first take place, which refers to splitting the text into individual lemmas.

Most automatic parsers are "trained" using a small corpus that has been manually parsed by a human previously, or at least one that was automatically parsed and then checked and corrected by the researcher (Gretz et al., 2015). These "gold-standard" pre-parsed corpora are called treebanks, and repositories of them they have been created for many languages. Building on existing databases of knowledge, these many of these parsers use statistical models to determine the most likely syntactic structure and conjugation for each word in each sentence.

Some parsers, however, are instead simply given entirely unparsed corpora and no knowledge of the language's syntactic structure. Working with nothing but the text itself, the program seeks out patterns and begins to create links and relationships that it deems significant.

Unfortunately, though automatic parsers have achieved surprising levels of accuracy in recent years, even the best continue to produce erroneous parsings. Some researchers have claimed 95% or higher accuracy, including for some Hebrew parsers. When dealing with such a large corpus, such as the Hebrew *OpenSubtitles2018* corpus

consisting of nearly 200 million tokens, a best-case scenario for a 5% error threshold results in nearly 10 million incorrectly parsed words.

Undoubtedly, this can have a negative impact on the accuracy of lemma frequency counts. Many of the issues found in the FDOSH are not due to orthographic ambiguity, but simply to inaccurate parsing. Some, as shown in the previous section, are even caused by erroneous automatic tokenization (consider the lemma "שדיבר").

The good news is that automatic parsers are continually improving in accuracy. This is a problem that exists across the board, regardless of the corpus being used—unless it is manually parsed and lemmatized, which is nearly impossible for such large corpora. The tools and techniques outlined in this thesis do not directly deal with the process of parsing.

5 Conclusion

This thesis has served as an in-depth look at the creation of the *Frequency Dictionary of Spoken Hebrew* (FDOSH). It has explained both the theory and the process, and in so doing has provided tools to facilitate the creation of similar frequency lists.

By identifying the decisions and outcomes of past studies, the literature review set the background for the most important factors to consider when undertaking such a project. These include corpus size and text type(s), the purpose of the frequency dictionary, the word family level to use, and the criteria by which to rank the list.

The methods chapter described—in detail—the process used to create the FDOSH. It explained how the corpus was found and cleaned, how the necessary data was extracted from it, how various measures were calculated, and how the frequency list was then sorted and exported into a full frequency dictionary. The relevant code was also explained, as well as instructions on specific changes that can be made depending on the needs of other researchers.

The organization and uses of the FDOSH were described in the last chapter. Most importantly, some of the challenges encountered during the process were discussed, along with possible directions for future projects. Some of the weaknesses of the FDOSH were also described.

Finally, the appendices to this thesis include a more full list of the FDOSH and all of the scripts used in their entirety. The code and full dictionary can be found as a supplement to this thesis, or at the *project's GitHub repository*[16].

The primary research question asked at the outset of this thesis was the following:

> What are the most common words in spoken Modern Hebrew?

Despite some deficiencies in the *Frequency Dictionary of Spoken Hebrew*, this question has be tentatively answered by the ranking of its words. *Most common words* in this case has been operationalized to mean most highly ranked lemmas by the usage coefficient of Gries' deviation of proportions, or U_{DP} (Gries, 2008, 2010). Of course,

[16] https://github.com/juandpinto/opus-frequencies

there can be no absolute, definitive answer to such a question, but the FDOSH provides one possibility.

The secondary research questions have been answered thus:

> What is an effective alternative for a corpus of spoken language when one is lacking in the desired language, as is often the case for less commonly taught languages?

A corpus of film and television subtitles offers an effective alternative. Though more study is needed in this area, the preliminary studies are in agreement on this point (Brysbaert & New, 2009; New et al., 2007). Importantly, this thesis has shown how obtaining and using such a corpus can be done easily despite the lack of resources that often plagues research for less commonly taught languages.

> How can the process of creating a frequency dictionary be simplified so that it is easy for others to reproduce while maintaining a high level of customizability?

This project aimed at making the entire dictionary-creation process as reproducible as possible while allowing for flexibility and transparency in the tools used. By using well-documented open-source scripts written in an easily readable programming language (Python) the result succeeds in this regard. The scripts themselves are a product of this project as much as the frequency dictionary is.

> What implications might these findings have for frequency list creation and use as it pertains to other less commonly taught languages?

The findings of this thesis are applicable to the task of frequency list creation for all languages. They are especially useful, however, to languages that lack the resources to compile and use corpora of spoken language. By tackling this problem, I hope that the current project serves as a catalyst for future research that may build upon the ideas discussed here. The development and open dissemination of tools such as these can only lead to greater cooperation among educators and researchers, to the benefit of all involved.

Appendix A: Frequency Dictionary of Spoken Hebrew (FDOSH)

LEMMA	RANK	DISPERSION	FREQUENCY	RANGE
הוא	1	114,718.51	121,008.92	99.99
ה	2	47,244.93	50,841.12	100.00
את	3	32,811.28	35,337.28	99.92
ל	4	27,415.19	29,102.77	99.97
לא	5	24,888.86	27,213.76	99.94
זה	6	23,817.89	26,418.69	99.96
ב	7	23,081.75	24,839.48	99.98
של	8	18,214.68	20,088.89	99.97
ש	9	18,203.83	20,028.64	99.95
היה	10	11,861.33	13,312.52	99.91
מה	11	10,879.07	12,192.80	99.87
ו	12	8,711.82	9,840.85	99.93
על	13	8,246.82	9,119.70	99.93
כול	14	6,062.08	6,842.01	99.90
ידע	15	5,328.40	6,205.85	99.69
כן	16	5,011.86	6,232.26	99.46
מ	17	4,871.00	5,479.15	99.89
יש	18	4,840.57	5,519.12	99.81
עשה	19	4,180.99	4,941.68	99.66
אבל	20	4,052.79	4,757.33	98.86
טוב	21	3,954.48	4,891.35	99.61
רצה	22	3,949.30	4,671.67	99.41
אם	23	3,846.61	4,444.59	99.68
עם	24	3,756.06	4,333.17	99.71
אמר	25	3,515.24	4,128.07	99.39
אז	26	3,370.31	4,052.24	99.41
סדר	27	3,197.62	4,305.52	98.33
צריך	28	2,862.13	3,501.64	99.18

LEMMA	RANK	DISPERSION	FREQUENCY	RANGE
רק	29	2,543.93	2,996.30	99.65
חשב	30	2,511.54	3,021.85	99.09
כאן	31	2,490.39	3,217.62	96.09
הלך	32	2,450.30	3,297.97	99.04
דבר	33	2,378.76	2,835.26	99.39
איש	34	2,307.24	2,904.93	98.85
אל	35	2,298.28	2,829.27	99.52
כך	36	2,269.89	2,777.32	99.29
יותר	37	2,260.37	2,682.46	99.42
שם	38	2,165.61	2,640.94	99.19
יכול	39	2,041.18	2,531.17	99.27
ראה	40	1,985.58	2,399.17	99.22
עכשיו	41	1,944.48	2,398.62	98.39
אחד	42	1,897.50	2,308.83	99.11
משהו	43	1,795.76	2,190.14	98.41
למה	44	1,793.48	2,234.96	98.04
בא	45	1,743.70	2,166.77	99.06
זאת	46	1,712.67	2,365.90	96.46
או	47	1,691.60	2,131.42	98.47
זמן	48	1,689.91	2,054.03	99.02
נכון	49	1,632.87	2,037.30	98.25
כמו	50	1,610.77	2,002.99	98.60
אין	51	1,607.85	1,945.44	98.70
איך	52	1,567.05	1,898.80	98.29
מי	53	1,543.35	1,927.41	98.23
זו	54	1,441.80	2,012.55	88.36
והיי	55	1,379.86	2,006.58	84.39
כמה	56	1,377.28	1,691.00	97.91
גם	57	1,332.74	1,657.26	98.26
אולי	58	1,280.63	1,630.97	97.19
נראה	59	1,279.60	1,603.47	97.94
בית	60	1,236.65	1,689.04	94.08

LEMMA	RANK	DISPERSION	FREQUENCY	RANGE
כדי	61	1,217.36	1,580.41	94.69
קרה	62	1,208.12	1,541.01	97.01
דיבר	63	1,179.13	1,495.64	95.83
פעם	64	1,176.27	1,479.18	97.08
דרך	65	1,123.95	1,431.59	96.47
כ	66	1,113.82	1,396.49	96.82
באמת	67	1,100.74	1,443.69	95.70
הגיע	68	1,095.14	1,383.15	96.61
מן	69	1,077.03	1,342.58	96.81
חייב	70	1,062.55	1,399.35	94.33
אחר	71	1,054.79	1,319.13	96.47
עוד	72	1,051.41	1,343.73	96.74
יום	73	1,050.26	1,374.21	95.22
פשוט	74	1,043.34	1,427.29	93.05
תודה	75	1,039.36	1,390.71	93.83
כי	76	1,034.28	1,431.30	89.69
כבר	77	1,015.89	1,292.23	96.34
ילד	78	1,011.91	1,478.80	89.75
אהב	79	1,011.48	1,378.72	92.60
חיים	80	1,009.68	1,325.23	95.52
בן	81	978.82	1,397.82	92.11
מישהו	82	954.85	1,230.50	94.16
קיבל	83	944.08	1,258.75	93.83
מאוד	84	929.61	1,249.26	93.05
לפני	85	917.78	1,165.26	94.91
אלה	86	907.66	1,205.87	87.61
אף	87	899.84	1,156.10	93.95
עד	88	891.54	1,126.08	94.78
הרבה	89	873.90	1,109.41	94.77
רגע	90	865.02	1,138.53	93.84
שנה	91	846.03	1,131.09	91.30
עדיין	92	840.75	1,074.35	93.91

LEMMA	RANK	DISPERSION	FREQUENCY	RANGE
עצמו	93	835.57	1,058.48	94.34
האם	94	826.99	1,301.20	73.10
ניסה	95	813.33	1,041.05	93.58
חזר	96	803.77	1,047.71	93.37
מצא	97	787.55	1,067.02	91.19
מקום	98	785.48	1,022.76	92.76
מת	99	779.75	1,080.04	89.81
איפה	100	759.84	1,029.43	87.91
אלוהים	101	757.35	1,088.19	81.96
אדם	102	735.17	1,032.75	87.64
הצטער	103	729.92	983.47	88.71
עבר	104	726.66	935.03	92.62
הכיל	105	721.00	947.86	78.96
הבין	106	717.29	921.06	92.27
חבר	107	707.96	964.33	88.48
גדול	108	704.79	940.92	90.22
איזה	109	700.02	924.60	91.13
ממש	110	699.63	984.20	85.99
בוא	111	686.35	946.82	87.72
נתן	112	686.02	887.93	90.78
קצת	113	684.83	904.88	88.71
שמע	114	683.28	883.99	90.89
עבודה	115	678.67	926.65	85.94
הנה	116	672.49	911.68	89.07
קדימה	117	670.69	1,026.32	79.11
שני	118	664.10	871.35	90.31
עזר	119	645.67	861.50	89.29
יצא	120	642.56	838.60	90.59
ובכן	121	642.13	974.64	62.40
כש	122	632.63	819.06	89.49
שוב	123	626.95	812.25	90.64
לילה	124	626.85	873.88	82.54

LEMMA	RANK	DISPERSION	FREQUENCY	RANGE
יד	125	620.79	840.78	85.78
היום	126	619.67	831.50	87.42
בדיוק	127	613.66	795.28	89.58
אחת	128	612.17	795.49	90.08
פה	129	601.90	924.87	74.53
בבקשה	130	599.85	823.33	84.87
הגיד	131	583.51	786.67	81.35
אי	132	573.45	758.57	88.11
קטן	133	570.58	767.54	86.64
שום	134	568.39	755.65	84.65
הרגיש	135	561.78	767.59	85.08
אמא	136	560.48	862.25	61.48
בטוח	137	559.73	731.46	88.42
אפילו	138	558.97	721.87	88.48
קשר	139	554.24	757.66	83.45
קרא	140	547.23	722.63	85.88
חדש	141	546.60	736.64	86.03
תמיד	142	545.28	715.90	87.31
אחרי	143	543.47	712.94	87.05
אבא	144	535.06	820.11	67.23
בשביל	145	533.18	732.76	82.15
האמין	146	518.41	691.02	85.32
בעיה	147	515.14	687.42	83.71
הכיר	148	512.17	677.77	83.61
התכוון	149	508.17	697.25	80.85
סיפר	150	505.56	691.75	82.05
מר	151	491.41	757.16	61.14
שלום	152	489.62	694.08	78.60
תן	153	486.46	653.40	82.27
אה	154	482.37	813.58	52.37
בטח	155	475.12	643.27	81.96
כסף	156	467.64	701.11	64.63

LEMMA	RANK	DISPERSION	FREQUENCY	RANGE
שעה	157	466.93	629.52	80.40
עבד	158	462.44	615.84	81.39
הביא	159	461.67	608.22	84.36
מדי	160	459.28	605.77	78.45
נמצא	161	457.04	624.10	79.81
בגלל	162	455.18	627.50	78.12
אחרון	163	452.98	597.58	83.38
הרג	164	451.62	665.89	68.86
ספר	165	449.54	650.41	74.55
מוכן	166	449.26	595.09	82.03
עניין	167	443.88	596.93	79.88
לקח	168	431.40	566.91	82.04
גרם	169	430.29	572.16	80.92
גבר	170	428.33	616.60	72.25
סיבה	171	428.26	587.32	79.20
לב	172	424.35	578.66	78.91
ראש	173	422.60	577.71	77.42
אפשר	174	422.23	577.08	77.23
שאל	175	413.95	548.02	78.70
חברה	176	408.91	581.20	72.35
עמד	177	402.70	532.42	79.28
אכל	178	402.39	545.16	79.05
חדר	179	401.80	562.65	73.60
קשה	180	395.98	520.41	80.36
אדוני	181	394.93	617.58	53.52
התחיל	182	394.86	515.97	80.57
רב	183	392.01	537.32	75.03
הניח	184	387.25	518.51	78.21
עולם	185	387.18	548.18	72.90
נשאר	186	380.76	507.27	77.96
תראה	187	378.02	517.75	72.87
שב	188	376.71	492.98	78.49

LEMMA	RANK	DISPERSION	FREQUENCY	RANGE
מקרה	189	375.80	506.88	75.90
משפחה	190	375.65	539.47	68.62
בוקר	191	372.69	511.84	70.63
עזאזל	192	360.31	517.82	65.84
כלום	193	353.91	480.33	71.95
חוץ	194	353.91	468.18	75.61
נכנס	195	352.28	466.72	76.00
שבוע	196	351.65	475.15	70.81
הו	197	350.18	560.98	39.94
הכי	198	349.75	479.66	71.92
אמור	199	348.92	462.96	76.71
די	200	346.59	474.97	71.93
חושב	201	342.94	479.44	66.63
עסק	202	342.92	471.81	68.96
חלק	203	341.92	453.04	75.44
סוף	204	338.98	453.48	72.77
בת	205	336.98	483.08	65.31
ביותר	206	331.01	456.85	66.65
עזב	207	325.30	438.85	71.42
מצב	208	322.42	434.39	72.24
זהו	209	322.32	456.73	64.44
אינו	210	321.85	502.82	50.63
שמר	211	317.92	419.67	73.36
פנים	212	317.30	428.38	72.46
בלי	213	317.15	425.60	71.57
יפה	214	316.26	430.19	68.26
חיפש	215	316.21	427.55	71.37
הביתה	216	314.34	429.84	67.49
עובד	217	312.02	417.59	70.28
עבור	218	310.46	443.04	56.82
בין	219	309.11	411.64	71.61
רע	220	308.16	411.00	70.89

LEMMA	RANK	DISPERSION	FREQUENCY	RANGE
הפך	221	307.03	412.11	71.26
אמת	222	306.55	419.64	68.39
כאילו	223	306.54	422.08	69.30
אוקיי	224	305.25	499.98	26.71
כמובן	225	302.61	410.97	67.32
עיר	226	298.46	433.88	60.03
הספיק	227	297.32	389.09	73.49
אוכל	228	296.04	407.47	67.38
מעולם	229	296.00	398.10	66.34
השתמש	230	295.21	395.83	70.66
שמח	231	294.56	400.63	69.21
זכר	232	293.72	397.52	68.77
המשיך	233	292.62	385.28	71.28
דקה	234	290.57	394.64	65.86
אמיתי	235	290.03	392.30	67.32
העליי	236	289.22	399.75	57.05
יחיד	237	288.04	380.58	72.16
בעל	238	284.38	395.08	63.93
נהדר	239	282.88	398.01	60.48
אכפת	240	281.91	375.84	68.98
קודם	241	280.72	369.46	73.40
אלו	242	280.47	412.35	49.71
תוכנית	243	280.06	398.15	61.26
כדאי	244	279.43	381.19	66.09
משחק	245	278.67	418.83	55.37
חשוב	246	275.19	364.04	68.92
ביקש	247	274.71	367.36	66.66
נעשה	248	274.53	363.45	69.24
נשמע	249	271.99	360.32	69.65
מכונית	250	271.64	406.22	48.95
לעולם	251	270.78	367.12	65.88
מספר	252	270.45	375.71	59.62

LEMMA	RANK	DISPERSION	FREQUENCY	RANGE
סליחה	253	267.60	375.07	59.50
נחמד	254	266.50	361.55	63.64
התקשר	255	265.76	366.95	58.75
עין	256	263.71	355.23	65.68
קיווה	257	257.45	339.39	68.33
סיפור	258	256.80	359.30	60.08
שאלה	259	256.49	346.90	63.92
בחור	260	255.74	353.18	58.89
חכה	261	255.11	353.50	59.62
קרוב	262	253.32	334.26	67.11
שינה	263	252.25	333.20	68.11
הפסיק	264	250.12	332.59	66.29
לעזאזל	265	248.74	354.18	55.63
הודה	266	248.56	338.13	61.89
כתב	267	248.00	352.64	54.34
עלה	268	247.77	327.08	64.47
מהר	269	246.33	336.87	62.24
מוות	270	245.02	348.60	56.35
אופן	271	243.60	327.33	61.95
טלפון	272	241.79	344.31	52.23
ישן	273	240.41	324.50	63.19
תרא	274	238.90	329.43	59.02
מחר	275	238.41	325.52	58.14
לאן	276	237.46	319.10	62.76
בכלל	277	236.95	314.78	63.73
אך	278	235.45	385.52	35.90
כוח	279	233.82	340.59	53.74
רעיון	280	232.52	312.93	61.15
לגבי	281	232.47	325.87	54.88
ילך	282	230.18	305.77	62.19
עצר	283	229.43	313.61	60.24
מוזר	284	228.13	313.45	59.44

LEMMA	RANK	DISPERSION	FREQUENCY	RANGE
ללא	285	227.75	318.60	56.58
מזל	286	227.34	305.24	61.30
הצליח	287	224.20	305.24	59.59
שנייה	288	224.03	302.19	60.16
צדק	289	223.60	298.74	62.90
גברת	290	223.51	331.04	46.59
חיכה	291	223.28	295.03	63.22
נוסף	292	223.11	303.43	58.84
דלת	293	222.89	312.74	55.40
אח	294	222.88	321.28	50.92
חזרה	295	222.68	303.50	59.48
חודש	296	222.22	301.43	57.18
מתי	297	220.96	293.94	62.07
חזק	298	220.12	298.58	59.77
משטרה	299	217.30	323.01	42.20
במקום	300	217.17	284.12	64.20
סוג	301	216.08	294.47	56.91
שיחק	302	215.72	302.95	54.25
למד	303	215.17	294.15	57.41
שלח	304	214.87	291.17	58.66
חץ	305	214.76	288.29	59.08
אחי	306	214.57	329.83	41.09
דם	307	214.07	320.20	46.70
חלה	308	213.85	310.72	48.88
כמעט	309	213.85	281.40	62.38
צוות	310	213.40	310.61	48.51
ברור	311	213.32	286.40	59.60
ערב	312	213.21	296.13	53.37
וה	313	213.06	279.53	61.81
דולר	314	212.44	314.65	42.20
בחר	315	212.23	282.89	60.19
חי	316	210.37	284.29	59.05

LEMMA	RANK	DISPERSION	FREQUENCY	RANGE
כלל	317	209.68	278.64	59.66
החזיק	318	209.68	282.37	58.78
בדק	319	208.91	285.13	57.00
לאחר	320	208.21	287.91	53.36
כנראה	321	206.82	280.36	57.42
כדור	322	206.50	303.73	46.81
רוח	323	205.42	288.02	53.87
הבחור	324	204.84	286.50	51.38
מאוחר	325	203.40	269.37	59.32
השאיר	326	203.30	268.59	61.25
קנה	327	202.33	277.93	55.14
רצח	328	201.55	310.41	37.19
הוציא	329	199.54	266.24	58.76
איתך	330	198.55	262.46	58.47
מבין	331	196.81	260.84	58.63
סיים	332	193.82	257.89	58.53
התראה	333	193.55	266.97	52.55
פחד	334	193.23	267.35	52.45
שלוש	335	193.23	263.61	52.44
למעשה	336	193.00	264.72	52.96
משרד	337	192.73	277.96	45.19
ככה	338	192.64	261.25	54.76
שילם	339	192.54	265.52	51.50
כאשר	340	192.01	289.88	36.29
גרוע	341	190.41	254.92	58.14
כבוד	342	190.23	266.43	50.20
הבטיח	343	190.15	255.29	56.37
חסר	344	189.02	252.86	56.51
תמונה	345	188.79	268.53	47.96
מלא	346	187.81	248.01	57.96
לכן	347	187.64	257.71	52.52
לבד	348	187.53	253.05	56.70

LEMMA	RANK	DISPERSION	FREQUENCY	RANGE
שוטר	349	187.16	290.24	34.34
איבד	350	187.01	251.51	56.14
נסע	351	186.91	264.57	46.83
השיג	352	185.24	249.66	54.79
לגמרי	353	184.51	251.39	53.29
החוצה	354	184.43	252.68	52.01
לפחות	355	183.48	241.21	59.20
נ	356	182.95	242.49	57.30
במשך	357	181.51	244.12	53.83
פרק	358	181.35	246.34	62.67
איתי	359	181.24	241.34	55.77
חושבת	360	180.30	248.37	51.72
פגע	361	180.05	245.53	54.00
הת	362	179.43	238.73	56.64
בחייך	363	178.82	250.63	47.83
סרט	364	176.56	270.08	37.79
שכח	365	176.16	234.51	55.13
בבקש	366	176.08	236.31	53.17
צעיר	367	175.76	242.48	49.47
ישב	368	175.29	233.75	53.77
בהחלט	369	175.05	236.73	52.15
שונה	370	174.57	235.15	53.14
קח	371	173.93	237.86	50.03
א	372	173.77	246.98	47.40
צריכה	373	173.21	242.19	49.81
מעל	374	171.99	233.18	52.90
קל	375	171.68	227.86	55.62
מטה	376	171.27	240.09	46.64
ותק	377	170.96	249.98	41.43
לשם	378	170.83	225.19	55.33
אהבה	379	170.76	249.54	42.09
יחד	380	170.56	234.73	49.78

LEMMA	RANK	DISPERSION	FREQUENCY	RANGE
קורה	381	169.46	230.18	52.98
הקשיב	382	169.18	228.03	52.05
אתמול	383	169.13	234.91	49.36
מילה	384	168.96	229.60	50.72
נקודה	385	168.85	231.75	49.83
הכול	386	168.62	269.42	26.86
צורה	387	168.36	229.16	51.63
נגע	388	167.80	235.17	48.48
בלתי	389	166.79	227.90	50.14
מים	390	166.16	241.56	43.35
למעלה	391	166.03	229.19	47.96
מושג	392	166.02	222.77	53.61
פתח	393	165.93	224.93	51.60
נהג	394	165.75	228.98	47.96
סתם	395	165.39	226.73	49.72
היכן	396	165.38	249.05	36.85
סלח	397	165.00	226.18	48.80
הסתכל	398	164.83	221.65	51.17
בתוך	399	164.25	224.12	50.58
כוונה	400	164.20	219.25	53.77
מייד	401	163.89	221.99	50.53
מערכת	402	163.87	236.45	43.68
נגמר	403	163.47	221.22	52.35
הזדמנות	404	162.97	220.09	51.64
תינוק	405	162.81	253.53	34.25
הראה	406	162.39	216.72	52.56
הערב	407	161.93	232.61	42.96
עזרה	408	161.69	218.48	52.99
אלא	409	161.12	215.41	52.68
אתן	410	160.15	220.47	49.53
סיכוי	411	159.13	216.52	50.78
הפעם	412	158.99	211.44	53.61

LEMMA	RANK	DISPERSION	FREQUENCY	RANGE
ניצח	413	158.95	225.98	45.25
הציל	414	158.90	227.24	45.47
נשק	415	158.83	231.36	41.68
רופא	416	157.77	234.77	38.33
שלושה	417	157.69	216.13	48.26
י	418	155.57	217.36	46.86
קול	419	154.70	224.59	41.31
אוי	420	154.66	231.73	37.85
כאב	421	153.91	216.83	45.67
שתי	422	153.26	206.35	49.88
אעשה	423	152.46	203.70	52.09
כפי	424	151.87	210.91	45.83
רציני	425	151.28	204.36	49.26
הציע	426	150.88	205.25	48.68
וואו	427	150.62	227.03	35.91
כלא	428	150.58	224.46	35.87
אדיר	429	150.29	218.21	40.99
כלומר	430	149.98	219.69	39.09
דין	431	148.77	226.93	32.94
ביחד	432	148.04	208.36	44.50
בעוד	433	147.82	200.88	48.87
כרגע	434	147.16	205.01	45.97
שיר	435	146.93	220.27	36.09
מלחמה	436	146.01	222.93	33.55
דעה	437	145.35	198.70	47.63
כלב	438	145.35	222.64	33.51
לפעמים	439	145.23	197.11	48.33
כעת	440	145.09	223.23	32.61
נעלם	441	145.00	202.86	46.00
שיחה	442	144.86	200.39	45.78
למען	443	144.62	200.50	45.80
חמש	444	144.42	200.75	44.09

LEMMA	RANK	DISPERSION	FREQUENCY	RANGE
רחוב	445	143.55	206.14	40.28
נורא	446	143.44	198.72	45.41
שניים	447	142.72	193.25	47.83
מיוחד	448	142.44	196.17	46.15
האליי	449	142.22	199.52	42.87
ירד	450	141.36	192.70	46.57
ודה	451	141.14	190.54	48.50
קבוצה	452	140.27	209.25	36.39
שאר	453	140.18	190.62	47.67
זונה	454	140.05	207.61	35.53
שכן	455	140.05	190.95	48.53
נגד	456	139.76	196.68	42.99
אלי	457	139.28	191.31	45.89
יצר	458	138.97	197.48	42.54
יופי	459	138.92	199.93	41.59
ארץ	460	138.07	206.49	36.05
מדינה	461	137.93	201.80	36.28
תפס	462	137.85	189.40	46.09
חוק	463	137.62	198.36	39.12
גר	464	136.55	194.35	41.38
החזיר	465	134.47	185.84	46.26
גש	466	133.53	183.49	44.84
אקדח	467	133.31	206.18	29.83
שה	468	132.88	179.04	47.86
מידע	469	132.61	194.28	37.14
טיפל	470	132.26	182.77	45.35
משפט	471	131.62	202.09	30.47
גנב	472	131.38	189.43	39.75
מסוגל	473	131.22	183.37	43.42
תורגם	474	129.97	183.74	45.73
ארוחה	475	129.52	182.99	42.09
שקט	476	129.42	180.46	42.85

LEMMA	RANK	DISPERSION	FREQUENCY	RANGE
צד	477	129.00	178.71	44.24
אש	478	128.53	192.82	34.68
מצטער	479	128.12	177.03	45.04
אב	480	127.75	187.02	37.09
ליד	481	127.31	173.76	46.14
טעות	482	127.07	175.47	44.74
פחות	483	126.65	172.52	45.98
רגיל	484	126.21	173.06	45.04
תיק	485	126.16	189.62	32.84
גבוה	486	125.89	173.98	43.16
מלך	487	125.08	207.75	22.84
מדוע	488	124.68	192.68	29.46
ניתן	489	123.81	171.44	43.46
הגן	490	123.53	175.92	41.03
הצלחה	491	123.35	169.09	44.49
מספיק	492	123.25	167.51	46.60
רכב	493	123.06	182.51	34.79
כיוון	494	123.02	173.34	41.40
פשע	495	122.99	183.46	33.51
הורה	496	122.54	178.63	37.53
הסכים	497	122.35	169.25	43.19
הוריד	498	121.29	169.55	42.05
לחץ	499	121.19	172.52	39.72
דאג	500	120.55	169.20	42.22
יכולת	501	120.27	166.59	43.79
נפלא	502	120.25	174.49	36.99
תדאג	503	118.78	164.66	43.78
תחת	504	118.33	164.84	41.40
הכין	505	118.30	165.99	42.91
עץ	506	118.20	176.20	34.28
הודעה	507	117.69	169.29	38.41
חרא	508	117.38	184.92	25.65

LEMMA	RANK	DISPERSION	FREQUENCY	RANGE
מוח	509	117.32	172.45	36.39
מטרה	510	117.20	170.87	36.70
גוף	511	117.18	170.08	37.95
הא	512	115.96	178.76	29.37
אצל	513	115.70	162.63	40.47
מצחיק	514	114.66	163.76	39.10
שנא	515	114.47	162.46	39.85
לפ	516	114.34	160.76	40.08
בגד	517	113.84	161.62	39.65
סימן	518	113.58	162.60	39.08
שווה	519	113.36	157.99	41.68
קטע	520	113.24	164.09	37.72
דוד	521	112.99	177.08	27.63
עלול	522	112.17	158.56	40.32
רוב	523	112.09	158.26	39.41
כוכב	524	111.92	174.76	28.47
העביר	525	111.76	155.94	41.36
אפשרי	526	111.14	156.08	40.63
פגישה	527	111.10	163.23	34.52
אור	528	110.93	159.84	37.41
מין	529	110.51	160.27	36.73
ביי	530	110.25	167.40	29.85
מנהל	531	110.17	163.33	33.04
בנה	532	109.72	155.16	39.68
ארוך	533	109.65	151.52	42.18
זוכר	534	109.29	156.21	38.98
תפקיד	535	108.84	158.07	35.75
נעל	536	108.59	157.99	36.44
ציפה	537	108.29	149.55	42.40
מוקדם	538	107.96	149.54	42.09
בקרוב	539	107.59	150.88	41.51
מתוק	540	107.45	158.32	35.12

LEMMA	RANK	DISPERSION	FREQUENCY	RANGE
הסביר	541	107.20	149.51	41.04
אסור	542	106.07	152.51	37.85
איתן	543	105.56	151.03	38.06
לבש	544	105.53	151.09	38.26
לפי	545	105.43	150.84	37.82
מעבר	546	105.37	148.95	38.99
דירה	547	105.31	160.75	30.09
סם	548	104.96	166.80	24.92
רחוק	549	104.93	147.68	39.81
שתיים	550	104.63	151.05	36.00
ניו	551	104.63	163.64	25.61
בדיקה	552	104.60	158.91	30.45
מאשר	553	104.47	145.52	40.57
זוג	554	104.14	152.22	35.37
עובדה	555	104.04	145.57	39.79
הופיע	556	103.57	146.45	38.53
אוויר	557	103.25	151.69	34.38
החלטה	558	103.24	148.45	36.77
זז	559	102.93	148.03	37.06
גידי	560	102.92	149.70	36.06
מעט	561	102.80	146.86	37.49
כרטיס	562	102.79	153.30	32.14
טיפש	563	102.68	147.63	37.05
מפה	564	102.26	152.42	32.65
שירות	565	102.21	148.24	36.28
אישי	566	101.44	143.37	38.25
ערך	567	101.31	143.65	38.67
קיים	568	101.12	144.55	37.63
שחור	569	101.00	151.78	31.18
עורך	570	100.98	154.40	28.23
זקוק	571	100.87	146.62	36.12
בחורה	572	100.87	149.06	33.73

LEMMA	RANK	DISPERSION	FREQUENCY	RANGE
התמודד	573	100.73	143.53	38.45
נלחם	574	99.82	150.51	31.64
מיטה	575	99.38	145.10	35.00
הזמין	576	99.18	141.84	37.52
מתחת	577	98.82	140.67	38.20
מחדש	578	98.77	141.67	37.24
אלך	579	98.46	139.13	39.89
מפקד	580	98.23	169.23	15.49
אימא	581	97.80	170.32	14.49
המ	582	97.21	143.47	33.35
הלו	583	97.14	145.96	30.80
משך	584	97.08	137.63	38.07
מהלך	585	97.02	140.43	35.67
בערך	586	96.87	137.83	37.36
חומר	587	96.47	143.05	32.25
אית	588	96.38	135.28	39.55
חלום	589	96.26	146.78	29.42
שחרר	590	95.53	139.94	34.85
בתור	591	95.05	137.75	35.91
ברח	592	94.92	138.25	35.70
שולחן	593	94.91	138.30	34.43
הוטרף	594	94.58	136.60	35.85
נפגש	595	94.47	134.66	36.94
למרות	596	94.43	135.05	36.92
צעד	597	94.31	136.83	34.80
צוחק	598	94.20	140.05	31.92
קפה	599	93.78	141.11	30.51
שאמר	600	93.72	131.03	39.83
מאחורי	601	93.51	132.70	37.82
יחסים	602	92.12	137.74	31.64
גב	603	91.87	139.40	29.29
חג	604	91.54	154.76	17.68

LEMMA	RANK	DISPERSION	FREQUENCY	RANGE
מס	605	91.34	144.25	24.65
חדשות	606	91.27	132.52	35.39
לבן	607	91.21	134.60	32.32
נרגע	608	91.10	134.64	32.48
ספק	609	91.06	130.46	35.98
מושלם	610	91.00	131.33	35.73
צהריים	611	90.83	131.93	34.26
רשימה	612	90.81	136.18	30.84
גמור	613	90.10	130.97	34.35
יורק	614	89.98	143.43	22.62
חשבון	615	89.67	131.59	32.83
זכות	616	89.63	131.24	32.80
שר	617	89.61	142.37	23.82
ארבע	618	89.43	130.65	32.90
התאים	619	89.32	127.22	37.08
עלייך	620	89.01	131.93	33.36
חם	621	88.99	129.91	33.78
שלומך	622	88.89	130.64	31.47
עתיד	623	88.67	133.23	30.80
נפל	624	88.67	127.96	34.94
ים	625	88.37	138.23	26.25
הכניס	626	88.15	125.56	36.44
ברוך	627	88.12	128.21	34.48
טעם	628	88.11	127.69	35.23
כיף	629	88.08	131.81	32.38
נשיא	630	87.94	151.43	14.64
תא	631	87.63	134.30	28.82
סביבה	632	87.58	126.17	35.11
נהנה	633	87.43	126.79	35.28
חמוד	634	87.17	130.86	31.34
רצינות	635	87.15	126.73	34.80
קו	636	87.10	130.14	30.45

LEMMA	RANK	DISPERSION	FREQUENCY	RANGE
קפטן	637	86.96	151.94	13.83
תחנה	638	86.89	132.88	28.48
מיליון	639	86.81	134.79	25.18
תקשיב	640	86.71	127.70	32.78
זקן	641	86.44	130.64	29.54
הוביל	642	86.13	125.32	34.38
מאה	643	86.07	126.35	32.78
אמצע	644	86.01	122.43	36.63
זכה	645	85.97	128.73	30.64
משימה	646	85.96	135.30	25.95
מותק	647	85.74	132.18	28.21
סמך	648	85.67	125.34	34.03
מטוס	649	85.64	140.52	19.84
מיני	650	85.13	125.26	33.18
אזור	651	84.85	127.80	30.21
פנימה	652	84.72	123.61	33.38
חנות	653	84.66	129.09	29.17
פעולה	654	84.47	124.84	31.95
שטח	655	84.33	127.03	30.38
הרגל	656	84.16	123.80	32.42
סבל	657	84.15	122.28	34.08
תשובה	658	83.94	121.96	34.23
אקח	659	83.74	119.79	36.29
עשר	660	83.60	123.75	31.52
תפסיק	661	83.23	123.12	32.08
הזכיר	662	83.00	118.09	36.66
נשא	663	82.44	121.55	31.89
ן	664	82.26	118.91	34.75
נקרא	665	81.97	118.08	35.09
אבי	666	81.84	130.11	24.05
מכר	667	81.76	121.68	30.41
ראייה	668	81.69	126.22	26.71

LEMMA	RANK	DISPERSION	FREQUENCY	RANGE
צא	669	81.47	122.33	30.11
לתוך	670	81.32	119.92	32.30
כה	671	80.81	123.13	28.65
הצטרך	672	80.76	116.10	35.18
החליט	673	80.58	116.23	34.49
ביצע	674	80.51	119.32	31.49
בניין	675	80.23	125.13	25.81
מול	676	80.15	117.33	32.96
קצר	677	80.09	115.08	34.72
מחשב	678	80.02	129.25	22.19
נעים	679	79.97	117.80	31.66
במיוחד	680	79.79	114.45	34.85
מלון	681	79.58	126.80	22.70
המון	682	79.28	118.28	30.50
אדום	683	79.12	120.91	27.54
שן	684	79.11	115.04	34.84
מוצא	685	79.07	114.63	33.66
שייך	686	79.03	116.56	32.31
הבחורה	687	78.86	120.11	28.53
בצד	688	78.77	114.58	33.38
ניסיון	689	78.69	114.03	33.99
תרופה	690	78.68	126.29	23.06
חקירה	691	78.67	125.22	23.21
שש	692	78.40	117.74	29.51
עשוי	693	78.29	117.16	30.72
יחידה	694	78.16	119.76	27.82
החליף	695	78.09	113.59	33.69
התחלה	696	78.06	112.05	34.89
גילה	697	78.04	113.89	33.73
תוך	698	77.98	114.03	32.61
חתיכה	699	77.94	115.55	31.43
אתר	700	77.88	119.61	28.18

LEMMA	RANK	DISPERSION	FREQUENCY	RANGE
נוח	701	77.84	113.84	33.10
חתך	702	77.82	115.55	31.19
צבע	703	77.62	118.29	27.79
מצוין	704	77.60	117.77	28.60
צורך	705	77.23	112.41	33.55
אדון	706	77.06	124.45	21.24
שער	707	77.05	119.87	25.67
יקר	708	76.86	112.60	32.43
העדיף	709	76.81	111.09	34.24
ככל	710	76.51	111.53	33.08
מסוכן	711	76.44	113.11	32.18
חבל	712	76.30	112.44	32.11
הגנה	713	76.28	116.55	27.78
גיל	714	76.25	114.61	29.36
הצטרף	715	76.01	111.71	32.55
ישר	716	75.93	110.98	31.73
שינוי	717	75.90	112.46	31.56
דובר	718	75.67	114.53	28.82
אראה	719	75.40	109.54	33.57
הרס	720	75.36	111.16	32.84
שותף	721	75.32	116.14	26.59
תהי	722	75.26	111.41	32.08
לכי	723	75.24	115.25	28.04
אדמה	724	75.20	117.03	26.09
ענה	725	75.03	109.98	34.03
אחראי	726	74.91	111.20	31.25
מסר	727	74.90	111.96	30.16
קורבן	728	74.64	123.17	19.34
סכנה	729	74.62	111.46	31.22
פיטר	730	74.59	127.62	15.62
הרשה	731	74.45	108.65	32.46
זרק	732	74.31	110.00	31.34

LEMMA	RANK	DISPERSION	FREQUENCY	RANGE
שיעור	733	74.01	114.08	26.87
הדבר	734	73.95	108.74	32.46
אידיוט	735	73.79	112.21	28.01
מפני	736	73.69	117.21	24.33
שליטה	737	73.32	109.77	30.16
עונה	738	73.31	109.04	37.54
אשר	739	73.19	116.53	22.47
כלשהו	740	73.13	109.14	30.39
אגיד	741	73.03	108.06	31.53
סגר	742	73.02	107.96	31.26
גדל	743	72.99	107.92	31.21
מבט	744	72.88	107.36	31.41
צפה	745	72.85	107.34	32.32
הדה	746	72.78	108.86	30.15
ספינה	747	72.68	127.94	12.90
ניתוח	748	72.67	121.60	18.25
אלף	749	72.59	114.09	23.65
מהיר	750	72.35	107.88	30.27
רמה	751	71.34	106.85	29.49
תוצאה	752	71.10	107.69	28.30
חכם	753	70.87	104.96	30.89
פרטי	754	70.60	105.91	29.14
השתנה	755	70.57	105.04	30.88
מתנה	756	70.44	107.54	27.87
ירה	757	70.28	109.72	24.66
הפריע	758	70.26	103.19	32.12
טיפול	759	69.74	109.18	25.11
אמריקני	760	69.72	111.85	21.48
שקר	761	69.57	105.60	28.44
נושא	762	69.54	104.50	28.73
מחלקה	763	69.36	108.70	24.29
עוזב	764	69.15	102.84	30.58

LEMMA	RANK	DISPERSION	FREQUENCY	RANGE
שמחה	765	69.09	103.53	30.09
חייל	766	69.01	115.20	18.08
מייקל	767	68.88	122.72	11.63
ניהל	768	68.87	102.32	29.75
חשבתי	769	68.81	100.66	33.34
עקב	770	68.78	103.58	29.62
הסתובב	771	68.73	101.50	30.78
איי	772	68.64	116.05	16.43
חופשי	773	68.51	102.71	29.49
כלי	774	68.44	105.06	27.12
צבא	775	68.39	111.92	19.90
מועדון	776	68.35	110.87	20.64
גמר	777	68.18	102.83	28.32
מחיר	778	68.15	103.97	27.50
הלוואה	779	67.98	101.61	30.42
היקח	780	67.92	100.47	30.99
ביטחון	781	67.77	103.94	26.92
פעל	782	67.66	101.72	29.23
הצעה	783	67.25	103.81	25.85
לקוח	784	67.03	107.94	21.29
תאונה	785	66.81	105.63	23.75
מפתח	786	66.76	104.54	24.69
לישון	787	66.60	100.94	28.29
הגיוני	788	66.37	99.27	30.04
מתוך	789	66.15	98.82	29.47
לחלוטין	790	66.05	99.56	28.78
אפשרות	791	66.00	98.89	29.42
ודאי	792	65.35	104.29	22.65
שחקן	793	65.10	105.82	19.91
סוכן	794	64.98	106.83	19.72
תני	795	64.92	98.64	28.39
רץ	796	64.68	98.53	26.76

LEMMA	RANK	DISPERSION	FREQUENCY	RANGE
ההוא	797	64.63	99.43	26.66
שלם	798	64.47	95.55	30.74
שלב	799	64.40	98.98	26.46
פתוח	800	64.34	96.43	29.52
חלון	801	64.25	98.73	26.26
איתה	802	64.06	96.34	29.75
עמוק	803	63.96	96.44	28.67
מרכז	804	63.91	99.29	25.15
שומר	805	63.75	97.66	27.22
פ	806	63.62	97.73	26.47
מעמד	807	63.55	96.65	27.24
שעשה	808	63.39	93.56	31.16
חשש	809	63.38	96.14	28.29
ק	810	63.36	104.35	18.97
חך	811	63.36	97.29	27.03
הוכיח	812	63.29	96.36	27.50
ייתכן	813	63.27	100.63	23.32
שנוכל	814	63.11	94.29	30.69
טום	815	63.11	109.65	13.41
אצטרך	816	63.01	93.98	30.70
תקופה	817	62.79	95.31	27.61
כבד	818	62.76	95.19	27.57
מהירות	819	62.71	97.15	25.25
שכר	820	62.66	96.45	25.66
שלט	821	62.61	95.73	27.68
התחתן	822	62.54	100.46	20.98
לפה	823	62.49	99.95	22.68
יכל	824	62.42	95.77	26.56
מסוים	825	62.36	94.24	28.25
מרחק	826	62.32	95.23	27.38
מאושר	827	62.25	96.31	25.40
החלק	828	62.17	93.62	29.14

LEMMA	RANK	DISPERSION	FREQUENCY	RANGE
בעצם	829	62.13	95.39	26.12
בירה	830	62.08	97.97	23.84
חייך	831	61.94	93.32	28.94
בר	832	61.89	95.34	26.00
עוזר	833	61.81	94.92	26.86
רגל	834	61.81	94.51	26.36
בנק	835	61.63	102.96	16.71
העריך	836	61.47	92.03	29.18
זיהה	837	61.38	93.99	27.04
טלוויזיה	838	61.29	98.20	22.01
כביש	839	61.07	97.48	22.27
האשים	840	61.05	92.69	27.93
תגיד	841	60.95	91.94	28.06
אכן	842	60.95	95.50	24.46
נצטרך	843	60.92	92.24	28.75
וב	844	60.90	91.07	28.64
שמונה	845	60.87	93.89	25.39
פרנק	846	60.50	109.13	9.56
פרט	847	60.41	91.84	27.44
נישואין	848	60.15	97.21	21.02
שדה	849	60.15	95.58	22.50
אביך	850	60.11	97.35	21.35
ם	851	59.99	93.39	24.98
עצור	852	59.86	94.74	22.94
נפגע	853	59.79	91.83	26.62
ידיד	854	59.48	93.52	23.85
קרב	855	59.34	93.78	23.48
ר	856	59.26	93.64	23.03
קלט	857	59.05	92.53	24.37
תסתכל	858	58.74	90.70	25.87
מכתב	859	58.74	96.48	18.09
הפחיד	860	58.59	89.45	27.30

LEMMA	RANK	DISPERSION	FREQUENCY	RANGE
השנה	861	58.57	91.50	24.12
שכנע	862	58.56	89.20	27.56
התרחק	863	58.50	90.19	26.52
רגש	864	58.37	91.16	25.05
שבר	865	58.34	89.28	26.86
התקרב	866	58.29	88.58	27.82
מעניין	867	58.26	88.40	27.88
גישה	868	58.25	90.20	25.78
הוגן	869	58.18	89.12	26.75
ע	870	58.04	91.50	23.15
הללו	871	57.95	95.52	18.59
ויתר	872	57.89	88.70	27.00
קר	873	57.64	89.51	25.08
שופט	874	57.54	95.67	17.09
נפטר	875	57.53	88.40	26.14
צפון	876	57.32	91.14	22.52
עדיף	877	56.98	86.97	27.27
אירוע	878	56.84	88.79	24.27
נו	879	56.81	90.12	23.19
ברית	880	56.65	92.53	19.56
הבחר	881	56.59	87.17	26.13
ארבעה	882	56.59	88.03	24.34
סמל	883	56.47	95.67	15.80
רעב	884	56.45	88.17	24.75
רצון	885	56.30	87.76	24.67
דן	886	56.16	92.91	17.72
הכה	887	56.02	87.37	24.21
אבן	888	56.00	92.09	18.87
איום	889	55.99	87.46	24.57
פגש	890	55.98	85.17	27.06
זבל	891	55.76	89.73	20.95
הידי	892	55.59	85.94	25.42

LEMMA	RANK	DISPERSION	FREQUENCY	RANGE
הסתיים	893	55.44	85.13	26.49
הלאה	894	54.99	84.86	25.81
נקי	895	54.96	85.41	24.94
לאט	896	54.75	87.63	21.24
סקס	897	54.70	91.36	18.36
לאחרונה	898	54.65	83.74	27.14
אחרת	899	54.42	82.17	27.83
תראו	900	54.02	84.65	24.81
זהיר	901	53.98	83.93	24.78
זין	902	53.87	92.29	13.97
התגעגע	903	53.71	84.83	23.49
תקווה	904	53.66	83.91	24.53
אבטחה	905	53.63	88.36	19.53
חטף	906	53.59	85.14	22.70
ראוי	907	53.57	83.35	24.66
כעס	908	53.49	83.58	24.77
נחש	909	53.20	82.60	25.41
תכנן	910	53.10	81.84	26.12
גיבור	911	53.04	86.81	19.56
מולד	912	52.99	92.44	12.60
פרס	913	52.98	87.24	18.25
מכירה	914	52.92	84.74	21.50
אנושי	915	52.92	86.22	20.76
ג	916	52.82	87.19	18.43
היסטוריה	917	52.70	82.86	23.11
שהייה	918	52.64	80.11	27.10
עבורך	919	52.61	83.65	22.76
מחשבה	920	52.44	81.10	25.47
סביב	921	52.38	81.09	24.87
סגן	922	52.38	90.44	13.73
פנה	923	52.25	81.00	24.76
התעורר	924	52.09	81.45	24.47

LEMMA	RANK	DISPERSION	FREQUENCY	RANGE
הריח	925	51.90	82.25	23.34
תיקן	926	51.89	81.78	24.27
עלי	927	51.89	79.63	26.53
גבול	928	51.69	82.24	22.46
רשת	929	51.65	83.90	20.47
נשבע	930	51.58	80.59	24.00
אמריקה	931	51.57	84.61	18.46
יגע	932	51.50	80.51	23.88
מידה	933	51.47	79.88	24.56
זהב	934	51.31	85.44	16.89
כיצד	935	51.25	84.64	18.36
מישהי	936	50.95	80.92	23.39
נדבר	937	50.92	78.53	25.59
כניסה	938	50.92	79.45	24.31
מקור	939	50.89	81.02	22.32
מעשה	940	50.82	79.71	23.77
ממשלה	941	50.65	85.59	15.44
היטב	942	50.62	79.21	24.03
תיכון	943	50.55	82.49	19.67
מנה	944	50.35	80.37	21.61
ביקר	945	50.30	78.52	24.09
אורח	946	50.22	79.08	23.23
עשרה	947	50.13	80.04	21.13
נהרג	948	50.07	80.13	21.97
חתונה	949	50.05	87.07	13.22
הדע	950	50.04	77.32	25.60
ריק	951	49.99	81.76	19.51
דרש	952	49.93	77.86	24.63
בפני	953	49.85	77.94	24.13
מורה	954	49.68	82.34	17.99
הפה	955	49.65	78.57	22.46
צחק	956	49.62	78.02	23.73

LEMMA	RANK	DISPERSION	FREQUENCY	RANGE
מוכר	957	49.55	77.12	24.50
שטות	958	49.51	77.72	22.46
תלוי	959	49.42	76.01	25.74
החבב	960	49.34	79.58	21.04
הושלם	961	49.30	77.06	24.45
קבע	962	49.27	76.58	24.47
רואה	963	49.26	77.06	25.19
משום	964	49.22	80.34	19.58
גן	965	49.17	79.64	20.27
טיסה	966	49.15	82.21	16.75
אלייך	967	49.12	78.14	22.92
עשי	968	49.09	76.60	25.30
מקומי	969	49.06	77.49	22.83
הפסיד	970	49.04	77.96	21.72
סגור	971	48.94	76.32	24.22
חן	972	48.87	78.86	20.31
שלישי	973	48.69	76.90	22.26
חירום	974	48.62	79.14	20.56
גאה	975	48.61	76.32	23.42
התחה	976	48.60	77.55	21.27
תנועה	977	48.58	77.32	21.60
לשעבר	978	48.58	77.90	21.29
טען	979	48.56	77.16	22.17
פול	980	48.55	85.39	11.51
ארון	981	48.47	79.30	19.75
אגב	982	48.35	75.36	24.81
נערה	983	48.20	79.83	17.85
נער	984	48.18	78.44	19.49
העמיד	985	48.15	75.05	24.81
בעצמך	986	48.03	73.90	25.94
השקר	987	48.01	76.21	22.81
חש	988	48.00	76.70	22.06

LEMMA	RANK	DISPERSION	FREQUENCY	RANGE
עצמך	989	47.97	73.87	25.64
הסתדר	990	47.91	74.51	24.68
תת	991	47.89	75.21	23.70
תחושה	992	47.89	75.66	22.99
תקף	993	47.83	77.29	21.56
סאם	994	47.76	88.78	6.89
עשית	995	47.68	73.74	25.56
בסיס	996	47.66	78.21	18.75
ראשי	997	47.65	75.85	21.65
דיווח	998	47.60	76.23	21.61
המשך	999	47.54	74.32	24.05
בוודאי	1000	47.54	77.55	19.42

Appendix B: Scripts

Appendix B.1: create-freq-list.py

```
#! /usr/bin/env python3
# -*- coding: utf-8 -*-

import re
import os
import gzip
from collections import defaultdict


################################################################
# ----------------- INITIALIZE VARIABLES ----------------- #
################################################################

# Define path for topmost directory to search. Make sure this points
↪  to
# the correct location of your corpus.
corpus_path = './OpenSubtitles2018_parsed_single/parsed/he'

# Initialize dictionaries
lemma_by_file_dict = {}
lemma_totals_dict = {}
lemma_norm_dict = {}
token_count_dict = {}
lemma_DPs_dict = defaultdict(float)
lemma_UDPs_dict = defaultdict(float)

total_tokens_int = 0
total_files_int = 0
```

```
table_list = []

# Set size of final list
list_size_int = 5000


################################################################
# ------------------ DEFINE FUNCTIONS ------------------ #
################################################################


# Open XML file and read it.
def open_and_read(file_loc):
    with gzip.open(file_loc, 'rt', encoding='utf-8') as f:
        read_data = f.read()
    return read_data


# Search for lemmas and add counts to "lemma_by_file_dict{}".
def find_and_count(doc):
    file = str(f)[40:-3]
    match_pattern = re.findall(r'lemma="[א-ת]+"', doc)
    for word in match_pattern:
        if word[7:-1] in lemma_by_file_dict:
            count = lemma_by_file_dict[word[7:-1]].get(file, 0)
            lemma_by_file_dict[word[7:-1]][file] = count + 1
        else:
            lemma_by_file_dict[word[7:-1]] = {}
            lemma_by_file_dict[word[7:-1]][file] = 1


################################################################
# ------------------- OPEN AND READ -------------------- #
```

```
61 ##############################################################
62
63 # Open and read all files. If calculating only for a specific
   ↪ language,
64 # comment out this code and uncomment the large block that follows.
65 #
66 for dirName, subdirList, fileList in os.walk(corpus_path):
67     if len(fileList) > 0:
68         total_files_int = total_files_int + 1
69         f = dirName + '/' + fileList[0]
70         find_and_count(open_and_read(f))
71
72 ##############################################################
73 # ---------------- LANGUAGE-SPECIFIC BLOCK -----------------
74 #
75 # This large block of code is for creating a list using only movies
   ↪ #
76 # with a specific primary language (in this case, Hebrew). Be sure
   ↪ to #
77 # uncomment the relevant lines of code, and to comment out the block
   ↪ #
78 # above. #
79 #
80 #
81 # Create list of IDs for movies with Hebrew as primary language. #
82 # This makes use of a text file that must already exist with this
   ↪ list. #
83 #
84 # Hebrew_IDs_list = []
85 # with open('./Hebrew_originals.txt', 'r', encoding='utf-8') as f:
86 #     read_data = f.read()
87 #     Hebrew_IDs_list = re.findall(r'\s\stt[0-9]+\t', read_data)
88 # Hebrew_IDs_list = [line[4:-1] for line in Hebrew_IDs_list]
```

```
#
#
# Delete extra 0s at the beginning of Hebrew movie IDs. #
#
# for item in Hebrew_IDs_list:
#     if item[0] == '0':
#         Hebrew_IDs_list[Hebrew_IDs_list.index(item)] = item[1:]
# for item in Hebrew_IDs_list:
#     if item[0] == '0':
#         Hebrew_IDs_list[Hebrew_IDs_list.index(item)] = item[1:]
#
#
# Open and read files for movies with Hebrew as the primary language. #
#
# for dirName, subdirList, fileList in os.walk(corpus_path):
#     if len(fileList) > 0:
#         f = dirName + '/' + fileList[0]
#         folders = re.split('/', dirName)
#         if folders[len(folders)-1] in Hebrew_IDs_list:
#             find_and_count(open_and_read(f))
#
# ------------- END OF LANGUAGE-SPECIFIC BLOCK -------------
################################################################


################################################################
# -------------------- CALCULATIONS -------------------- #
################################################################

# Calculate total raw frequencies per lemma
for lemma in lemma_by_file_dict:
    lemma_totals_dict[lemma] = sum(lemma_by_file_dict[lemma].values())
```

```
121
122 # Calculate token count per file
123 for lemma in lemma_by_file_dict:
124     for file in lemma_by_file_dict[lemma]:
125         token_count_dict[file] = token_count_dict.get(
126             file, 0) + lemma_by_file_dict[lemma][file]
127
128 # Calculate total token count
129 for file in token_count_dict:
130     total_tokens_int = total_tokens_int + token_count_dict.get(file,
    ↪  0)
131
132 # Set value by which to measure normalized frequency (freq per x
    ↪  words)
133 freq_per_int = 1000000
134
135 # Calculate normalized frequencies per lemma
136 for lemma in lemma_totals_dict:
137     lemma_norm_dict[lemma] = lemma_totals_dict[lemma] /
    ↪  total_tokens_int * \
138         freq_per_int
139
140 # Calculate DPs
141 for lemma in lemma_by_file_dict.keys():
142     for file in lemma_by_file_dict[lemma].keys():
143         lemma_DPs_dict[lemma] = lemma_DPs_dict[lemma] + abs(
144             (token_count_dict[file] /
145              total_tokens_int) -
146             (lemma_by_file_dict[lemma][file] /
147              lemma_totals_dict[lemma]))
148 lemma_DPs_dict = {lemma: DP/2 for (lemma, DP) in
    ↪  lemma_DPs_dict.items()}
149
```

```
150 # Calculate UDPs
151 lemma_UDPs_dict = {lemma: (1-DP)*lemma_norm_dict[lemma] for (lemma,
    ↪ DP) in
152                    lemma_DPs_dict.items()}
153
154
155 ################################################################
156 # -------------- SORT LIST AND CREATE TABLE -------------- #
157 ################################################################
158
159 # Sort entries by UDP
160 UDP_sorted_list = [(k, lemma_UDPs_dict[k]) for k in sorted(
161     lemma_UDPs_dict, key=lemma_UDPs_dict.__getitem__,
162     reverse=True)]
163
164 # Create list of tuples with all values (Lemma, Rank, UDP,
    ↪ Frequency, Range)
165 i = 0
166 for k, v in UDP_sorted_list[:list_size_int]:
167     i = i + 1
168     table_list.append((k,
169                        i,
170                        '{0:,.2f}'.format(v),
171                        '{0:,.2f}'.format(lemma_norm_dict[k]),
172                        '{0:,.2f}'.format(sum(1 for count in
173
    ↪ lemma_by_file_dict[k].values() if
174                                              count > 0) /
175                                             total_files_int * 100)))
176
177 ################################################################
178 # ---------------- SORT-BY-FREQUENCY BLOCK -----------------
179 #
```

```
# Sort entries by raw frequency (total lemma count). To sort the final #
# list by frequency instead of UDP, comment out the above code within the #
# "SORT LIST AND CREATE TABLE" section, and also uncomment the relevant #
# lines of code in this block. #
#
#
# Sort entries by raw frequency #
#
# frequency_sorted_list = [(k, lemma_totals_dict[k]) for k in sorted(
#     lemma_totals_dict, key=lemma_totals_dict.__getitem__,
#     reverse=True)]
#
#
# Create list of tuples with all values (Lemma, Frequency, Range, UDP) #
#
# for k, v in frequency_sorted_list[:list_size_int]:
#     table_list.append((k, v, sum(
#         1 for count in lemma_by_file_dict[k].values() if count > 0),
#         lemma_UDPs_dict[k]))
#
# ------------- END OF SORT-BY-FREQUENCY BLOCK -------------
################################################################

# Calculate list size for 80% coverage and set that as the list size. Note
# that if the initial list_size_int (set near the beginning of the script)
```

```
# provides less than the desired coverage, it will default to that instead.
#
# added_freq_int = 0
# count = 0
# for k, v in UDP_sorted_list:
#     if added_freq_int / total_tokens_int < 0.8:
#         added_freq_int = added_freq_int + lemma_totals_dict[k]
#         count = count + 1
#     else:
#         break
# list_size_int = count

# Write final tallies to TSV file
result = open('./export/frequency-dictionary.tsv', 'w')
result.write('LEMMA\tRANK\tDISPERSION\tFREQUENCY\tRANGE\n')
for i in range(list_size_int):
    result.write(str(table_list[i][0]) + '\t' +
                 str(table_list[i][1]) + '\t' +
                 str(table_list[i][2]) + '\t' +
                 str(table_list[i][3]) + '\t' +
                 str(table_list[i][4]) + '\n')
result.close()

# Print final tallies. Uncomment this code to see the results
# printed instead of writing them to a file.
#
# for i in range(list_size_int):
#     print('Lemma: ' + table_list[i][0] +
#           '\tFrequency: ' + str(table_list[i][1]) +
#           '\tRange: ' + str(table_list[i][2]) +
#           '\tUDP: ' + str(table_list[i][3]))
```

Appendix B.2: OMDb-fetch.py

```
#! /usr/bin/env python3
# -*- coding: utf-8 -*-

from sys import argv
import os
import glob
import omdb

script, year = argv

# Initialize IDs list
IDs = []

# Create list of all movie directory paths for desired year
for name in glob.glob(
        './OpenSubtitles2018_parsed_single/parsed/he/' + year +
        ↪  '/*/'):
    IDs.append(name)

# Trim list of directories to only the movie IDs
IDs = [os.path.basename(os.path.dirname(str(i))) for i in IDs]

# Add additional zeros to beginning of IDs to match with database
for i in IDs:
    while len(i) < 7:
        IDs[IDs.index(i)] = '0' + i
        i = '0' + i

# Sort IDs numerically (easier to use results)
IDs.sort()

```

```
# Replace the API key here (906517b3) with your own (omdbapi.com)
omdb.set_default('apikey', '906517b3')

# Print table header
print('# ' + year + '\n' +
      'IMDb ID\tTitle\tYear\tLanguage(s)')

# Fetch and print movie ID, title, year, and language(s)
for i in IDs:
    doc = omdb.imdbid('tt' + i)
    print('tt' + i + '\t' +
          doc['title'] + '\t' +
          doc['year'] + '\t' +
          doc['language'])
```

Appendix B.3: single_file_extract.py

```
#! /usr/bin/env python3
# -*- coding: utf-8 -*-

import shutil
import os

source = '../OpenSubtitles2018_parsed'
destination = './OpenSubtitles2018_parsed_single'

# Copy the directory tree into a new location
shutil.copytree(source, destination,
 ↪  ignore=shutil.ignore_patterns('*.*'))

# Copy the first file in each folder into the new tree
for dirName, subdirList, fileList in os.walk(source):
    for fname in fileList:
        if fname == '.DS_Store':
            fileList.remove(fname)
    if len(fileList) > 0:
        del fileList[1:]
        src = dirName + '/' + fileList[0]
        dst = destination + dirName[27:] + '/'
        shutil.copy2(src, dst)
```

References

Adelman, J. S., Brown, G. D. A., & Quesada, J. F. (2006). Contextual diversity, not word frequency, determines word-naming and lexical decision times. *Psychological Science*, *17*(9), 814–823. https://doi.org/10.1111/j.1467-9280.2006.01787.x

Albert, A., MacWhinney, B., Nir, B., & Wintner, S. (2013). The Hebrew CHILDES corpus: Transcription and morphological analysis. *Language Resources and Evaluation*, *47*(4), 973–1005. https://doi.org/10.1007/s10579-012-9214-z

Al-Surmi, M. (2012). Review: Quaglio (2009). Television dialogue: The sitcom Friends vs. Natural conversation. Philadelphia: John Benjamins. *Corpora*, *7*(1). https://doi.org/10.3366/corp.2012.0022

Amir, N., Silber-Varod, V., & Izre'el, S. (2004). Characteristics of Intonation Unit Boundaries in Spontaneous Spoken Hebrew and Acoustic Correlates. In B. Bernard & I. Marlien (Eds.), *Speech Prosody 2004, Nara, Japan, March 23-26, 2004: Proceedings* (pp. 677–680). Nara, Japan.

Anglin, J. M., Miller, G. A., & Wakefield, P. C. (1993). Vocabulary development: A morphological analysis. *Monographs of the Society for Research in Child Development*, *58*(10), i–186. https://doi.org/10.2307/1166112

Balota, D. A., & Chumbley, J. I. (1984). Are Lexical Decisions a Good Measure of Lexical Access? The Role of Word Frequency in the Neglected Decision Stage. *Journal of Experimental Psychology: Human Perception & Performance*, *10*, 340–357.

Balota, D. A., Yap, M. J., Cortese, M. J., Hutchison, K. A., Kessler, B., Loftis, B., … Treiman, R. (2007). The English Lexicon Project. *Behavior Research Methods*, *39*(3), 445–459.

Bauer, L., & Nation, P. (1993). Word families. *International Journal of Lexicography*, *6*(4), 253–279. https://doi.org/10.1093/ijl/6.4.253

Biber, D. (1988). *Variation across speech and writing.* Cambridge, UK: Cambridge University Press.

Biber, D. (1993). Representativeness in corpus design. *Literary and Linguistic Computing*, *8*(4), 243–257. https://doi.org/10.1093/llc/8.4.243

Biber, D. (1995). *Dimensions of Register Variation: A Cross-Linguistic Comparison.* Cambridge University Press.

Biber, D., Conrad, S., & Leech, G. (2002). *Longman student grammar of spoken and written English.* Harlow, Essex: Longman.

Biber, D., Johansson, S., Leech, G., Conrad, S., & Finegan, E. (1999). *Longman grammar of spoken*

and written English. Harlow: Longman.

Brezina, V., & Gablasova, D. (2015). Is there a core general vocabulary? Introducing the new general service list. *Applied Linguistics*, *36*(1), 1–22. https://doi.org/10.1093/applin/amt018

Browne, C. (2014). A new general service list: The better mousetrap we've been looking for? *Vocabulary Learning and Instruction*, *3*(1), 1–10.

Brysbaert, M., & New, B. (2009). Moving beyond Kučera and Francis: A critical evaluation of current word frequency norms and the introduction of a new and improved word frequency measure for American English. *Behavior Research Methods*, *41*(4), 977–990. https://doi.org/10.3758/BRM.41.4.977

Carroll, J. B. (1970). An alternative to Juilland's usage coefficient for lexical frequencies and a proposal for a standard frequency index. *Computer Studies in the Humanities and Verbal Behaviour*, *3*(2), 61–65.

Carroll, J. B., Davies, P., & Richman, B. (1971). *The American heritage word frequency book*. Boston: Houghton Mifflin.

Chung, T. M., & Nation, P. (2003). Technical vocabulary in specialised texts. *Reading in a Foreign Language*, *15*(2), 103.

Chung, T. M., & Nation, P. (2004). Identifying technical vocabulary. *System*, *32*(2), 251–263. https://doi.org/10.1016/j.system.2003.11.008

Collins Cobuild English grammar. (2005). Glasgow: HarperCollins.

Cowie, A. P. (2009). *The Oxford History of English Lexicography*. Oxford Univ.

Coxhead, A. (2000). A new academic word list. *TESOL Quarterly*, *34*(2), 213–238. https://doi.org/10.2307/3587951

Coxhead, A. (2016). Reflecting on Coxhead (2000), "a new academic word list". *TESOL Quarterly*, *50*(1), 181–185. https://doi.org/10.1002/tesq.287

Dang, T. N. Y., & Webb, S. (2016). Evaluating lists of high-frequency words. *ITL - International Journal of Applied Linguistics*, *167*(2), 132–158. https://doi.org/10.1075/itl.167.2.02dan

Dekel, N. (2010). *A matter of time: Tense, mood and aspect in Spontaneous Spoken Israeli Hebrew*. Utrecht: LOT.

Delic, E., Teston-Bonnard, S., & Véronis, J. (2004). Présentation du Corpus de référence du français parlé. *Recherches Sur Le Français Parlé*, *18*, 11–42.

Dice, L. R. (1945). Measures of the Amount of Ecologic Association Between Species. *Ecology*,

26(3), 297–302. https://doi.org/10.2307/1932409

Droop, M., & Verhoeven, L. (2003). Language proficiency and reading ability in first- and second-language learners. *Reading Research Quarterly*, *38*(1), 78–103.

Ellis, N. C. (2002a). Frequency effects in language processing: A a review with implications for theories of implicit and explicit language acquisition. *Studies in Second Language Acquisition*, *24*. https://doi.org/10.1017/S0272263102002024

Ellis, N. C. (2002b). Reflections on frequency effects in language processing: A response to commentaries. *Studies in Second Language Acquisition*, *24*, 297–339. https://doi.org/10.1017.S0272263102002140

Forster, K. I., & Chambers, S. M. (1973). Lexical access and naming time. *Journal of Verbal Learning and Verbal Behavior*, *12*(6), 627–635. https://doi.org/10.1016/S0022-5371(73)80042-8

Francis, W. N., Kučera, H., & Mackie, A. W. (1982). *Frequency analysis of English usage: Lexicon and grammar*. Boston: Houghton Mifflin.

Fries, C. C., & Traver, A. A. (1960). *English Word Lists*. Ann Arbor: George Wahr.

Gardner, D. (2007). Validating the construct of word in applied corpus-based vocabulary research: A critical survey. *Applied Linguistics*, *28*(2), 241–265. https://doi.org/10.1093/applin/amm010

Gilner, L. (2011). A primer on the general service list. *Reading in a Foreign Language*, *23*(1), 65.

Gilquin, G., & Gries, S. T. (2009). Corpora and experimental methods: A state-of-the-art review. *Corpus Linguistics and Linguistic Theory*, *5*(1), 1–26. https://doi.org/10.1515/CLLT.2009.001

Goldberg, Y. (2011, November). *Automatic Syntactic Processing of Modern Hebrew* (PhD thesis). Ben-Gurion University, Beer-Sheva, Israel.

Goldberg, Y., & Elhadad, M. (2009). Hebrew dependency parsing: Initial results. In *Proceedings of the 11th International Conference on Parsing Technologies* (pp. 129–133). Paris: Association for Computational Linguistics.

Goldberg, Y., & Elhadad, M. (2010). Easy-First Dependency Parsing of Modern Hebrew. In *Proceedings of the NAACL HLT 2010 First Workshop on Statistical Parsing of Morphologically-Rich Languages* (pp. 103–107). Los Angeles, CA, USA: Association for Computational Linguistics.

Goldberg, Y., & Elhadad, M. (n.d.). Two Syntactic Parsers for Modern Hebrew and a large automatically parsed corpus.

Gretz, S., Itai, A., MacWhinney, B., Nir, B., & Wintner, S. (2015). Parsing Hebrew CHILDES transcripts. *Language Resources and Evaluation*, *49*(1), 107–145. https://doi.org/10.1007/s10579-013-9256-x

Gries, S. T. (2008). Dispersions and adjusted frequencies in corpora. *International Journal of Corpus Linguistics*, *13*(4), 403–437. https://doi.org/10.1075/ijcl.13.4.02gri

Gries, S. T. (2010). Dispersions and adjusted frequencies in corpora: Further explorations. In S. T. Gries, S. Wulff, & M. Davies (Eds.), *Corpus linguistic applications: Current studies, new directions* (pp. 197–212). Amsterdam: Rodopi.

Gries, S. T. (2017). *Quantitative corpus linguistics with R* (2nd ed.). New York: Routledge.

Guthmann, N., Krymolowski, Y., Milea, A., & Winter, Y. (2008). Automatic Annotation of Morpho-Syntactic Dependencies in a Modern Hebrew Treebank. *LOT Occasional Series*, *12*, 77–90.

Hirsh, D., & Nation, P. (1992). What vocabulary size is needed to read unsimplified texts for pleasure? *Reading in a Foreign Language*, *8*, 689–696.

Hlaváčová, J. (2006). New approach to frequency dictionaries: Czech example. In (p. 6). Genoa.

Hoek, J., Evers-Vermeul, J., & Sanders, T. (2015). The role of expectedness in the implicitation and explicitation of discourse relations.

Hu, M., & Nation, P. (2000). Unknown vocabulary density and reading comprehension. *Reading in a Foreign Language*, *13*(1).

Itai, A., & Segal, E. (2003). A Corpus Based Morphological Analyzer for Unvocalized Modern Hebrew.

Izre'el, S. (2004). Transcribing Spoken Israeli Hebrew: Preliminary Notes. In D. D. Ravid & H. B.-Z. Shyldkrot (Eds.), *Perspectives on Language and Language Development: Essays in Honor of Ruth A. Berman* (pp. 61–72). Kluwer: Dodrecht. https://doi.org/10.1007/1-4020-7911-7_6

Izre'el, S., Auran, C., Bertrand, R., Chanet, C., Colas, A., Di Cristo, A., … Vion, M. (2005). Intonation Units and the Structure of Spontaneous Spoken Language: A View from Hebrew. In *Proceedings of the IDP05 International Symposium on Discourse-Prosody Interfaces* (p. 20).

Izre'el, S., Hary, B., & Rahav, G. (2001). Designing CoSIH: The Corpus of Spoken Israeli Hebrew. *International Journal of Corpus Linguistics*, *6*(2), 171–197. https://doi.org/10.1075/ijcl.6.2.01izr

Jakubíček, M., Kilgarriff, A., Kovář, V., Rychlý, P., & Suchomel, V. (2013). The TenTen Corpus Family. In *7th International Corpus Linguistics Conference CL 2013* (pp. 125–127). Lancaster.

Jang, S.-C. (1998). *Dimensions of spoken and written Taiwanese: A corpus-based register study* (Ph.D. Dissertation). University of Hawaii.

Jeon, E. H. (2011). Contribution of morphological awareness to second-language reading comprehension. *The Modern Language Journal*, *95*(2), 217–235. https://doi.org/10.1111/j.1540-4781.2011.01179.x

Juilland, A. G., Brodin, D. R., & Davidovitch, C. (1970). *Frequency dictionary of French words*. The Hague: Mouton de Gruyter.

Jurafsky, D. (2003). Probabilistic modeling in psycholinguistics: Linguistic comprehension and production. In R. Bod, J. Hay, & S. Jannedy (Eds.), *Probabilistic linguistics* (pp. 39–96). Cambridge, MA: MIT Press.

Kilgarriff, A., & Grefenstette, G. (2003). Introduction to the special issue on the web as corpus. *Computational Linguistics*, *29*(3), 333–347.

Kobeleva, P. P. (2012). Second language listening and unfamiliar proper names: Comprehension barrier? *RELC Journal*, *43*(1), 83–98. https://doi.org/10.1177/0033688212440637

Kučera, H., & Francis, W. N. (1967). *Computational analysis of present-day American English*. Brown University Press.

Leech, G., Rayson, P., & Wilson, A. (2001). *Word Frequencies in Written and Spoken English: Based on the British National Corpus*. Harlow: Pearson Education.

Lijffijt, J., & Gries, S. T. (2012). Correction to Stefan Th. Gries' "Dispersions and adjusted frequencies in corpora " International Journal of Corpus Linguistics 13:4 (2008), 403-437. *International Journal of Corpus Linguistics*, *17*(1), 147–149. https://doi.org/10.1075/ijcl.17.1.08lij

Lison, P., & Tiedemann, J. (2016). OpenSubtitles2016: Extracting large parallel corpora from movie and TV subtitles. *Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC 2016)*, 7.

Lyne, A. A. (1985). Dispersion. In *The vocabulary of French business correspondence* (pp. 101–124). Geneva, Paris: Slatkine-Champion.

Matsushita, T. (2012). In What Order Should Learners Learn Japanese Vocabulary? A Corpus-based Approach.

McLean, S., & Kramer, B. (2015). The creation of a new vocabulary levels test. *Shiken*, *19*(2), 1–11.

Mettouchi, A., Lacheret-Dujour, A., Silber-Varod, V., & Izre'el, S. (2007). Only Prosody? Perception of speech segmentation in Kabyle and Hebrew. In *Intefaces discours prosodie : Actes du 2ème Symposium international & Colloque Charles Bally* (pp. 207–218).

Milton, J. (2009). *Measuring second language vocabulary acquisition*. Bristol, UK ; Buffalo N.Y.: Multilingual Matters.

Mochizuki, M., & Aizawa, K. (2000). An affix acquisition order for EFL learners: An exploratory study. *System*, *28*, 291–304. https://doi.org/10.1016/S0346-251X(00)00013-0

Nagy, W., Anderson, R. C., Schommer, M., Scott, J. A., & Stallman, A. C. (1989). Morphological families in the internal lexicon. *Reading Research Quarterly*, *24*(3), 262–282. https://doi.org/10.2307/747770

Nagy, W. E., Diakidoy, I.-A. N., & Anderson, R. C. (1991). The development of knowledge of derivational suffixes. *Center for the Study of Reading Technical Report; No. 536*.

Nation, I. (1982). Beginning to learn foreign vocabulary: A review of the research. *RELC Journal*, *13*(1), 14–36. https://doi.org/10.1177/003368828201300102

Nation, I. (2006). How large a vocabulary is needed for reading and listening? *Canadian Modern Language Review*, *63*(1), 59–82.

Nation, I. (2016). *Making and using word lists for language learning and testing*. Amsterdam: John Benjamins Publishing Company. https://doi.org/10.1075/z.208

Nation, I. S. P. (1990). *Teaching & learning vocabulary* (1 edition). Boston, Mass: Heinle ELT.

Nation, I. S. P. (2013). *Learning vocabulary in another language* (2nd ed.). Cambridge: Cambridge University Press.

Nation, I. S. P., & Webb, S. (2010). *Researching and analyzing vocabulary* (1 edition). Boston, MA: Heinle ELT.

Nation, P. (2000). Learning vocabulary in lexical sets: Dangers and guidelines. *TESOL Journal*, *9*(2), 6–10. https://doi.org/10.1002/j.1949-3533.2000.tb00239.x

Nation, P. (2004). A study of the most frequent word families in the British National Corpus. In P. Bogaards & B. Laufer (Eds.), *Language Learning & Language Teaching* (Vol. 10, pp. 3–13). Amsterdam: John Benjamins Publishing Company. https://doi.org/10.1075/lllt.10.03nat

Nation, P., & Coxhead, A. (2014). Vocabulary size research at Victoria University of Wellington, New Zealand. *Language Teaching*, *47*(03), 398–403. https://doi.org/10.1017/S0261444814000111

Nation, P., & Kyongho, H. (1995). Where would general service vocabulary stop and special purposes vocabulary begin? *System*, *23*(1), 35–41. https://doi.org/10.1016/0346-251X(94)00050-G

New, B., Brysbaert, M., Veronis, J., & Pallier, C. (2007). The use of film subtitles to estimate word frequencies. *Applied Psycholinguistics*, *28*(4), 661–677. https://doi.org/10.1017/S014271640707035X

Oldfield, R. C., & Wingfield, A. (1965). Response latencies in naming objects. *Quarterly Journal of Experimental Psychology*, *17*(4), 273–281. https://doi.org/10.1080/17470216508416445

Popescu, M., & Dinu, L. P. (2008). Rank Distance as a Stylistic Similarity. In *Coling 2008: Companion volume: Posters* (pp. 91–94). Manchester, UK: Coling 2008 Organizing Committee.

Quaglio, P. (2009). *Television dialogue: The sitcom Friends vs. Natural conversation*. John Benjamins Publishing.

Read, J. (1988). Measuring the vocabulary knowledge of second langauge learners. *RELC Journal*, *19*(2), 12–25. https://doi.org/10.1177/003368828801900202

Restrepo Ramos, F. D. (2015). Incidental vocabulary learning in second language acquisition: A literature review. *PROFILE Issues in Teachers' Professional Development*, *17*(1), 157–166. https://doi.org/10.15446/profile.v17n1.43957

Rosengren, I. (1971). The quantitative concept of language and its relation to the structure of frequency dictionaries. *Études de Linguistique Appliquée*, *1*, 103–127.

Rychlý, P. (2008). A lexicographer-friendly association score. In P. Sojka & A. Horák (Eds.) (pp. 6–9). Karlova Studánka, Czech Republic: Masaryk University.

Savický, P., & Hlavácová, J. (2002). Measures of word commonness. *Journal of Quantitative Linguistics*, *9*(3), 215–231. https://doi.org/10.1076/jqul.9.3.215.14124

Schmitt, N. (2010). *Researching vocabulary: A vocabulary research manual*. Houndmills, Basingstoke, Hampshire; New York: Palgrave Macmillan.

Schmitt, N., Jiang, X., & Grabe, W. (2011). The percentage of words known in a text and reading comprehension. *The Modern Language Journal*, *95*(1), 26–43. https://doi.org/10.2307/41262309?ref=no-x-route:cb78a69b6dc8bf1478b58d47243b1248

Schmitt, N., & Meara, P. (1997). Researching vocabulary through a word knowledge framework. *Studies in Second Language Acquisition*, *19*(1), 17–36.

Schmitt, N., & Schmitt, D. (2014). A reassessment of frequency and vocabulary size in L2 vocabulary teaching. *Language Teaching*, *47*(04), 484–503. https://doi.org/10.1017/S0261444812000018

Schmitt, N., & Zimmerman, C. B. (2002). Derivative Word Forms: What Do Learners Know? *TESOL Quarterly*, *36*(2), 145–171. https://doi.org/10.2307/3588328

Sima'an, K., Itai, A., Winter, Y., Altman, A., & Nativ, N. (2001). Building a tree-bank of modern Hebrew text. *Traitement Automatique Des Langues*, *42*(2), 247–380.

Sorell, C. J. (2012). Zipf's law and vocabulary. In C. Chapelle (Ed.), *The Encyclopedia of Applied Linguistics*. Hoboken, NJ, USA: John Wiley & Sons, Inc.

Sorell, C. J. (2013). *A study of issues and techniques for creating core vocabulary lists for English as an international language* (Unpublished Dissertation). Victoria University of Wellington, Wellington, New Zealand.

Sørensen, T. (1948). A method of establishing groups of equal amplitude in plant sociology based on

similarity of species and its application to analyses of the vegetation on Danish commons. *Kongelige Danske Videnskabernes Selskab*, *5*(4), 1–34.

The history of Collins COBUILD. (n.d.). https://www.collinsdictionary.com/cobuild/.

Thorndike, E. L. (1941). *The teaching of English suffixes*. New York: Teachers College, Columbia University.

Tiedemann, J. (2012). Parallel Data, Tools and Interfaces in OPUS, 5.

Tiedemann, J. (2016). Finding Alternative Translations in a Large Corpus of Movie Subtitles, 5.

Tyler, A., & Nagy, W. (1989). The acquisition of English derivational morphology. *Journal of Memory and Language*, *28*(6), 649–667. https://doi.org/10.1016/0749-596X(89)90002-8

Tyler, A., & Nagy, W. (1990). Use of derivational morphology during reading. *Cognition*, *36*(1), 17–34. https://doi.org/10.1016/0010-0277(90)90052-L

van Zeeland, H., & Schmitt, N. (2013). Lexical coverage in L1 and L2 listening comprehension: The same or different from reading comprehension? *Applied Linguistics*, *34*(4), 457–479. https://doi.org/10.1093/applin/ams074

Wang, M., Cheng, C., & Chen, S.-W. (2006). Contribution of morphological awareness to Chinese-English biliteracy acquisition. *Journal of Educational Psychology*, *98*(3), 542–553. https://doi.org/10.1037/0022-0663.98.3.542

Ward, J., & Chuenjundaeng, J. (2009). Suffix knowledge: Acquisition and applications. *System*, *37*(3), 461–469. https://doi.org/10.1016/j.system.2009.01.004

Webb, S. A., & Chang, A. C.-S. (2012). Second language vocabulary growth. *RELC Journal*, *43*(1), 113–126.

West, M. (1953). *A general service list of English words: With semantic frequencies and a supplementary word-list for the writing of popular science and technology* (Rev. and enl. ed.). London, New York: Longmans, Green.

Whitney, P. (1998). *The Psychology of Language*. Houghton Mifflin.

Xue, G., & Nation, I. S. P. (1984). A university word list. *Language Learning and Communication*, *3*(2), 215–229.

Yael, M. (2014). The Haifa Corpus of Spoken Hebrew. http://weblx2.haifa.ac.il/~corpus/corpus_website/.

Zhang, H., Huang, C., & Yu, S. (2004). Distributional consistency: As a general method for defining a core lexicon. In (pp. 1119–1122). Lisbon.

Zipf, G. K. (1935). *The psycho-biology of language*. Cambridge, Mass.: M.I.T. Press.

Zipf, G. K. (1949). *Human behavior and the principle of least effort: An introduction to human ecology.* New York: Hafner.