W(ha/il)t's the (wo/ba)rd?*

A cross-sectional analysis of English word pairs that satisfy the properties of the Split Decisions word puzzle

Oliver Daniel

April 30, 2022

Abstract

From three English wordlist corpora, pairs of words are generated which might appear in the Split Decisions word puzzle as published by the New York Times, i.e., at least 5 characters in length and differing by exactly two consecutive characters. Even in absence of usage-related data, several ortho-phonotactic trends emerge that puzzle creators can employ to granularly increase or decrease the difficulty of guessing such a pair of words. Also discussed are morpheme and syllable boundaries, and how alterations of these boundaries within a word pair can make mentally recalling one given the other more difficult.

Keywords: crosswords, split decisions, computational linguistics

Code available on GitHub (link).

Contents

Introduction . . . 1
- About Split Decisions . . . 2
- Rules, Notation and Terminology . . . 2

Data . . . 3
- Corpora: What's a "word"? . . . 3
- Processing & Caching . . . 4
- Variables . . . 5

Results . . . 7
- 1- and 2-grams vs. Zipf's Law . . . 7
- Split pairs . . . 10

Discussion . . . 12
- Orthotactics . . . 14
- Phonotactics . . . 15
- Local phonological interactions . . . 16

Appendix . . . 19
- Link to Python preprocessing script . . . 19

References . . . 20

*"What's the word?" or "Wilt's the bard?" (nonsense).

Introduction

This paper analyzes pairs of words, specifically those bearing between them a particular orthographic property described below, as used in a crossword-like puzzle. In particular, we are interested in what factors of these words may make correctly guessing an arbitrary pair more or less difficult, given only a few letters of each. Although many words would be ostensibly ineligible for said puzzle for a number of reasons, especially those sourced from more extensive English lexicons, strong trends emerge from merely analyzing distributions of the characters that make up these word pairs. A combination of the grammatical, orthographic, and phonological systems which underlie spoken and written English converge within these trends, allowing our research to attempt to identify which of these systems might aid or hinder a solver's deduction process. We also identify multiple avenues for future study, in particular by psycholinguists who are interested in the mental connection between written English words and their pronunciations.

About Split Decisions

A few times a year, the New York Times games department releases a new issue of a word puzzle called Split Decisions. Similar to crosswords, the game board consists of boxes arranged in sequence, one for each letter of a word running either left-to-right or top-to-bottom, intersecting with other words at right angles, as shown below in Figure 1. The Split Decisions website (Piscop 2021) attributes the original puzzle format to the late George Bredehorn; after his death, the mantle was taken up by Fred Piscop, whose puzzles appear in the *Times* to this day.

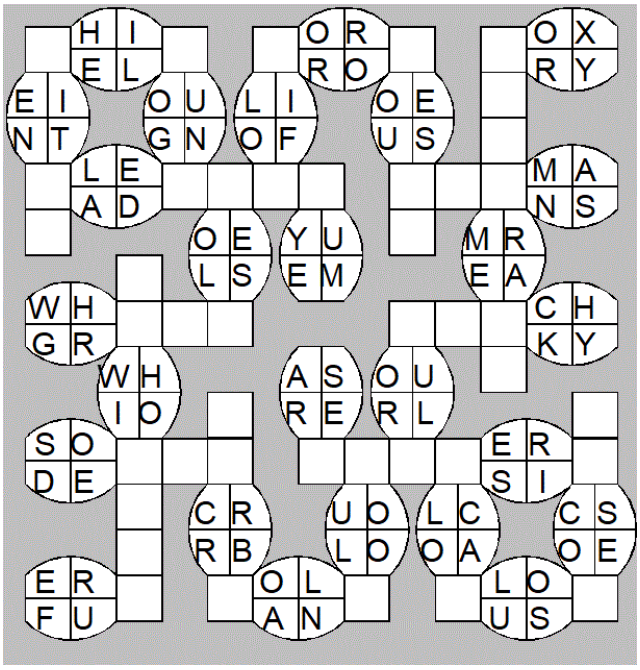

Figure 1: An example Split Decisions puzzle from Piscop's website.

Evidently, instead of cryptic descriptions of correct answers, Split Decisions solvers are only given pairs of two-letter substrings for each word. The remaining boxes, each containing one letter as in a typical crossword, combine with either of these substrings to form an English word. For example, the top-rightmost word, spelled horizontally ("across"), could take a P in its singular empty box to form the words POX and PRY, or an F to form FOX and FRY, potentially among other combinations. However, in filling out the rest of the puzzle, it becomes evident that only the latter solution accommodates both the across word pair, and the down pair that intersects it.

Rules, Notation and Terminology

In order to analyze different aspects of the game, we must first lay out a few rules of how Split Decisions puzzles work in general, as well as some additional constraints we introduce for the purpose of study. Additionally, we lay out some notation and terminology we will use to succinctly describe different segments of a puzzle word.

For lack of a robust LaTeXmechanism for creating authentic-looking Split Decisions clues, like in Figure 1, we will use simpler notation to represent word pairs. Firstly, we will largely use underscores to represent those boxes left unfilled by the puzzle setter, and some visual delineator (often vertical alignment or slashes) to represent the provided two-letter substrings. For certain purposes, we may use a full equation-level LaTeXenvironment, like:

$$\binom{PL}{GR}----$$

But, the above pair may also more succintly be represented in-text by a form like (PL/GR)_____ when more appropriate.

A correct fill for the remaining boxes might be (PL/GR)EASE, creating both PLEASE and GREASE, or as ___EASE when the two missing letters are inessential. For the remainder of this paper, such a pair of words will be referred to as a *word pair*, sharing a *common substring*[1] and differing by a *split pair*. The common substring and split pair meet at *boundaries*: using the convention of left-to-right for English, we might describe the above diagram as possessing a *right boundary*, but lacking a left one (as the split begins the word). In the (PL/GR)EASE notation used above, this right boundary is represented by the concatenation of the right parenthesis with another letter. In general, the shorthands *prefix* and *suffix* will be used to refer to common fill before the left boundary and after the right, respectively. Returning to the above example once more, (PL/GR)EASE lacks a prefix and has a suffix of "EASE".

Note that each clued split pair might accept several different fills for the common substring, e.g., (PL/GR)UNGE (PLUNGE, GRUNGE); however, as with crosswords, only one solution will also satisfy all the crossing constraints produced by the rest of the puzzle.

Although a letter may appear on both sides of a split, say, (IT/TA), the letter will *never* appear at the same index in both words, e.g., (IT/AT), as such a pair of words would only differ by a single letter instead of two. Also, although such a puzzle would be fascinating to both solve and analyze, crossing words do *not* go through split pairs, only common letters.

[1] Or *fill*: this part of the puzzle is filled in by the player rather than the constructor.

Finally, although a word pair can theoretically be as short as three letters, this paper examines only pairs of length five and above.

Upon understanding the rules of the puzzle, one may start to wonder: what sorts of words form split pairs? Are some splits more common than others? How do the properties of English, as both a spoken and written language, affect the guessability of a given split pair? In this paper, we perform preliminary statistical analysis to investigate these questions, using a variety of computational methods.

Data

Corpora: What's a "word"?

One of the most common side effects of continued study in linguistics is a diminished ability to describe what defines a "word". For the purpose of this study, we have taken three lists that purport to contain a wide variety of English words, though certainly not their totality, and used them to generate all possible combinations that produce a valid Split Decisions word pair (in terms of a two-character split pair, etc.). As a result, many entries that appear in one or more of these corpora may seem esoteric, or even offensive, to be cosnidered as words; however, the only predicate for their inclusion in these lists is attested usage, not whether they prescriptively *ought* to be a word. Words appear in these corpora in lexicographical order (i.e., sorted alphabetically), one on each line, in lower-case letters.

The first and smallest corpus comes from an open-source distribution of GNU/Linux, specifically Ubuntu. For brevity, I will usually refer to this corpus as `linux` (note the lower-case letters and monospace font). This corpus, developed over time by online contributors through the development of Ubuntu, is accessed on Linux machines through the symbolic link `/usr/share/dict/words`. It is primarily used as a baseline for other text processing programs' spell-checking and text prediction functionalities.

The `collins` corpus (*Collins Official Scrabble® Words* 2019) is privately produced by HarperCollins LLC, a publishing company well known for producing English- and multiple-language dictionaries. This dictionary in particular finds notable use in competitive Scrabble play: competitors in so-called 'Collins divisions' at tournaments will study from and refer to this list as the superset of all playable Scrabble words.

Similarly, the `nwl` corpus (NASPA Dictionary Committee 2020) was produced directly by members of the North American Scrabble Association (NASPA) for the purpose of tournament play. In Canada and the United States, this word list serves as the *de facto* standard for competitive Scrabble, and is usually the default wordlist unless another is specified, as in the aforementioned Collins divisions.

From each of these corpora, only those entries consisting solely of five or more Latin characters without diacritics were accepted for further processing.

Processing & Caching

The algorithm for determining whether two strings of the same length form a Split Decisions pair is quite simple.

1. Traverse the indices of both strings, comparing their characters pairwise. That is, compare the first letters of both strings, then the second, and so on.

Table 1: Post-processing statistics for each corpus of English words.

	Words		Word pairs	Most frequent (occurrences)		
Corpus	Total	Valid	Total	First split	Second split	Split pair
linux	102774	92243	124332	ck (2603)	st (3135)	ng/on (640)
collins	279496	272384	763072	er (15851)	st (14809)	ng/on (1632)
nwl	191852	186471	434847	er (8996)	st (9008)	ng/on (1235)

2. Record each index at which the two characters differ.
3. Once traversal is complete, if there are exactly two differences and they differ in index by exactly one, then the two strings form a word pair. Otherwise, continue with another combination of strings.

However, initial experiments with identifying these word pairs directly in R (R Core Team 2021) proved to be untenably slow. Since every word in a corpus has to be compared to every other word of the same length at least once, any algorithm[2] used to do so cannot run in better than quadratic time, and even R packages with bindings to C took several hours to run on even the smallest of the corpora. Instead, a Python script was written to process the words more efficiently, and then write the results in a CSV file that could be more readily imported into the R environment. This precomputation phase significantly reduced start-up time when creating R sessions to author this paper, or to knit this document into a PDF using `knitr` (Xie 2021). A link to this script can be found in the Appendix.

Apart from these, the remainder of data processing from this paper was performed in R using the popular `tidyverse` package collection (Wickham et al. 2019), and visualized using the indispensible `ggplot2` (Wickham 2016).

Variables

These cached CSVs accounted for the following variables of each word pair:

1. `length`: the length of each word in the pair. Ranges from 5 to 21.
2. `x`: the lexicographically-first full word in the pair, with no additional notation for the split or any boundaries.
3. `y`: the lexicographically-second full word in the pair.
4. `start`: the 1-index of the beginning of the split. For example, a split at the beginning of a word would have value 1.
5. `end`: The 1-index of the end of the split. This is a convenience variable: each split has a length of exactly two, and hence the ending index is always equal to `start + 1`.
6. `split_x`: The lexicographically-first of the two splits. As a property, the `start`th to `end`th substring of `x` is always this value.
7. `split_y`: The lexicographically-second of the two splits, as above.
8. `corpus`: The corpus from which this pair was determined. One of `linux`, `collins`, or `nwl`.

In certain context-specific scenarios, other variables are computed:

9. `type`: divides each word pair into one of three categories, depending on the positioning

[2] i.e., any naive iterative algorithm.

of its split pair. If the split pair is at the beginning of the word, the type is "prefix"; if at the end, "suffix"; and otherwise, "middle". (PL/GR)EASE is a "prefix"-type pair, for example.

10. `rank`: when comparing different pairs, splits, etc. by their frequency, a ranking variable is introduced to keep track of which is the most frequent, the second most frequent, and so on.

Figure 2, below, demonstrates the distribution of these different split types across word pairs of different corpora and lengths. For example, 15-letter word pairs from the `linux` corpus (in green) overwhelmingly favour suffix-type splits.

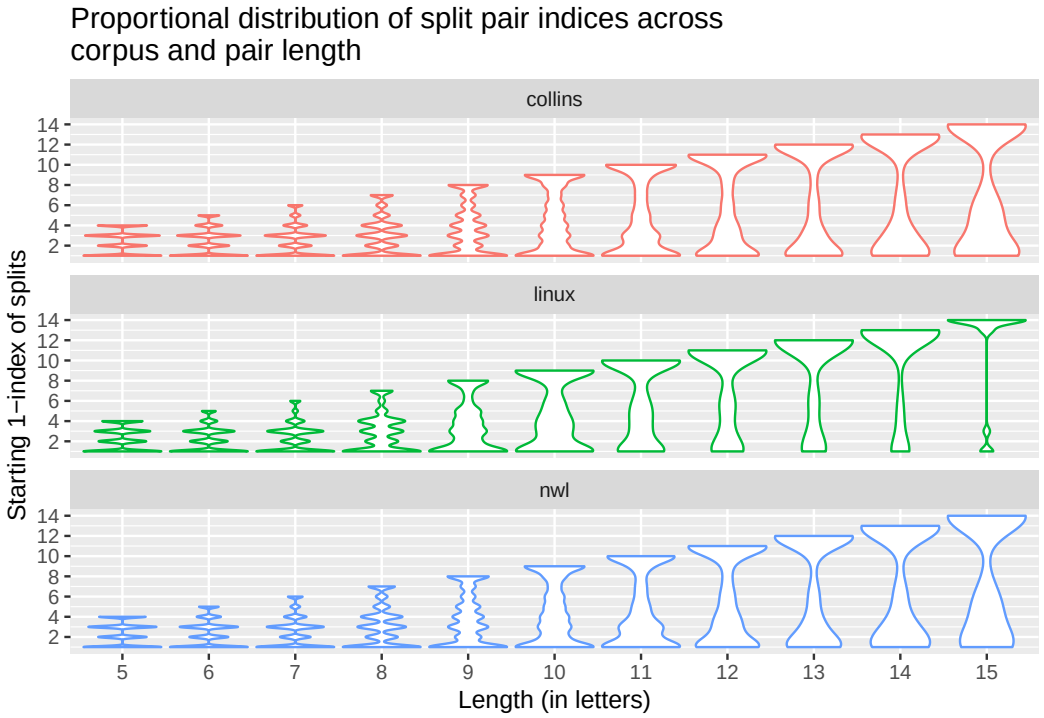

Figure 2: Width at a particular y-value represents the density of splits in word pairs of a given length, starting at that index.

As discussed later in **Results**, we hypothesize that the goblet-like shape seen predominantly in longer words (i.e,., 9+ letters) is due to a widened variety of verbal stems that differ only in their declension, as indicated by a suffix (e.g., participating, participation).

Looking at all consecutive pairings of letters (*2-grams*) in each corpus, we see that 614 out of a possible $26^2 = 676$ pairings are accounted for at least once. Figure 3 below depicts the relative frequency of these pairings, with the white squares representing those pairs that appear nowhere in the corpus (or corpora). Some more infrequent first letters, such as *q* or *z*, demonstrate large streaks of unattested pairs. This is in line with certain orthotactic[3] constraints present in the English language and will be discussed in greater

[3]Pertaining to the rules and constraints of spelling. E.g., "pnt" violates English orthotactic rules, and

detail in **Discussion**.

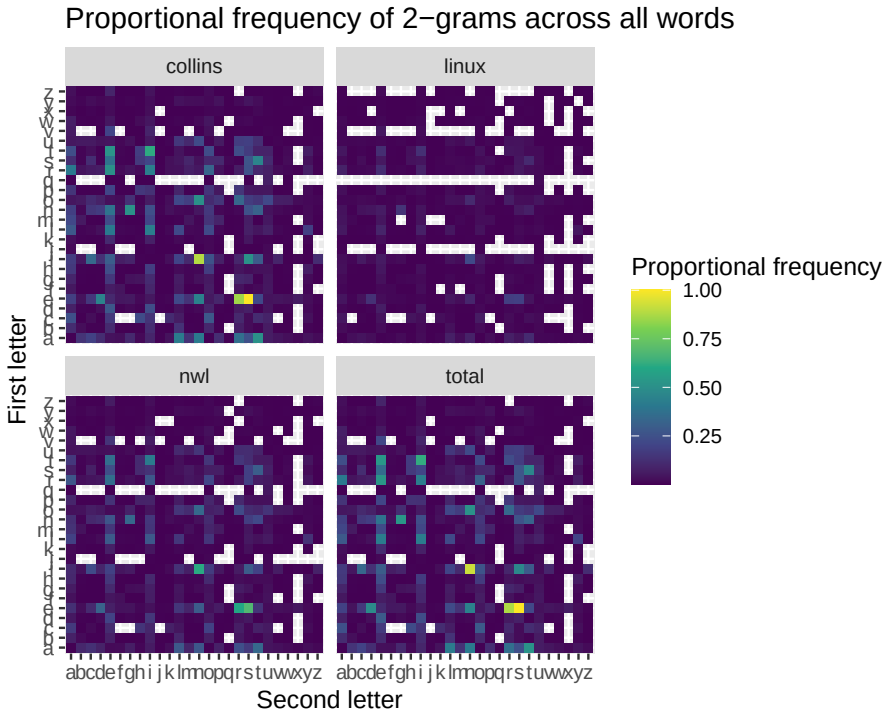

Figure 3: Distributions appear relatively consistent across corpora, except for particularly esoteric combinations: the Linux corpus contains no q's followed by anything other than a u.

Results

1- and 2-grams vs. Zipf's Law

Upon encountering such a wide distribution of categorical data (i.e., ordered strings of characters), our first thought was to see if some aspect of the distribution obeyed **Zipf's Law** (Zipf 1949). In brief, this empirical law states that when observations in a categorical set are ranked by frequency, the second-ranked observation occurs roughly half as often as the first, the third-ranked a third as often, and so on. In mathematical terms, over a frequency-ordered distribution X with ranking variable i,

$$\text{Zipf}(X) \iff X_i \approx \frac{X_1}{i}.$$

Before examining the splits and other puzzle-specific properties of words, we examined the distribution of one- and two-letter sequences, herefore *1-grams* and *2-grams*, respectively, to see if any Zipf-like patterns emerged. Surprisingly, however, the data outperformed Zipf's Law (i.e., contained many more observations than the power law predicts) in these

thus could never constitute a non-abbreviated English word.

categories. Figure 4 below, which examines the total occurrences of each letter in all valid words, exemplifies this. The black dashed curve represents the expected count of each letter according to Zipf's Law, whereas the total height of each bar represents the actual count. As an example, the second-most common letter, s, is expected to occur $539,056/2 = 269,528$ times, or 50% of the total occurrences of e. In the data, it instead appears roughly 84% as often, a total of 450,495 times. This disparity between Zipfian expectation and reality only increases in higher ranks, including the universally rare q and j, with a mean percent error of 166.3513% across all 26 letters.

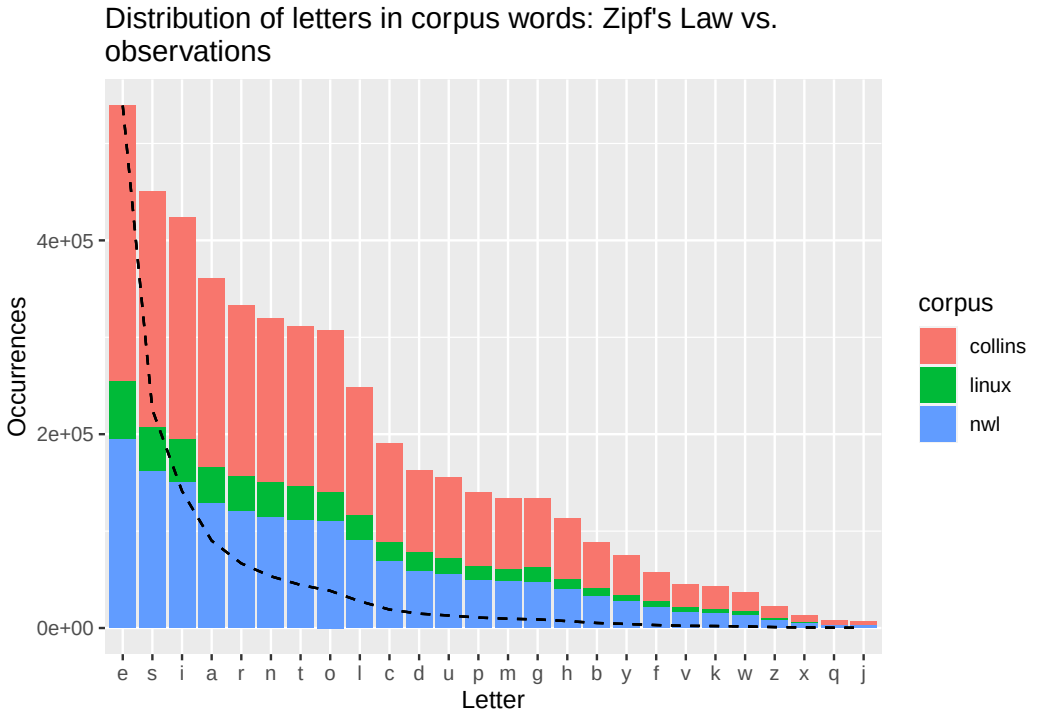

Figure 4: The total distribution of letters across all valid corpus words is consistently greater than the ideal Zipf curve (black) would predict.

Looking at each corpus individually, we can also examine their 2-grams; that is, each consecutive pair of characters that occurs in every valid word. As an example, the word "aardvark" contains the 2-grams "aa", "ar", "rd", "dv", "va", "ar", and "rk". (Note that "ar" occurs twice, and is counted as such.) Figure 5 below again pits the frequency of these 2-grams against the expectation of Zipf's Law, yielding an interesting result. For clarity and visual interest, some of these 2-grams have been included beside their place on the graph, serving as waypoints of increasingly uncommon letter pairings.

As can be observed, all three corpora (points) have an overlarge 'head' to their distributions compared to the Zipf ideal (dashed lines), but only up to a certain point. At or about rank 420, the Zipf curves of each corpus intersect with their respective observed distributions. This is indicated by the vertical dotted line on Figure 5. From there, this 'tail' of the curve begins to *underperform* Zipf, with all 2-grams less frequent than rank 420 appearing considerably

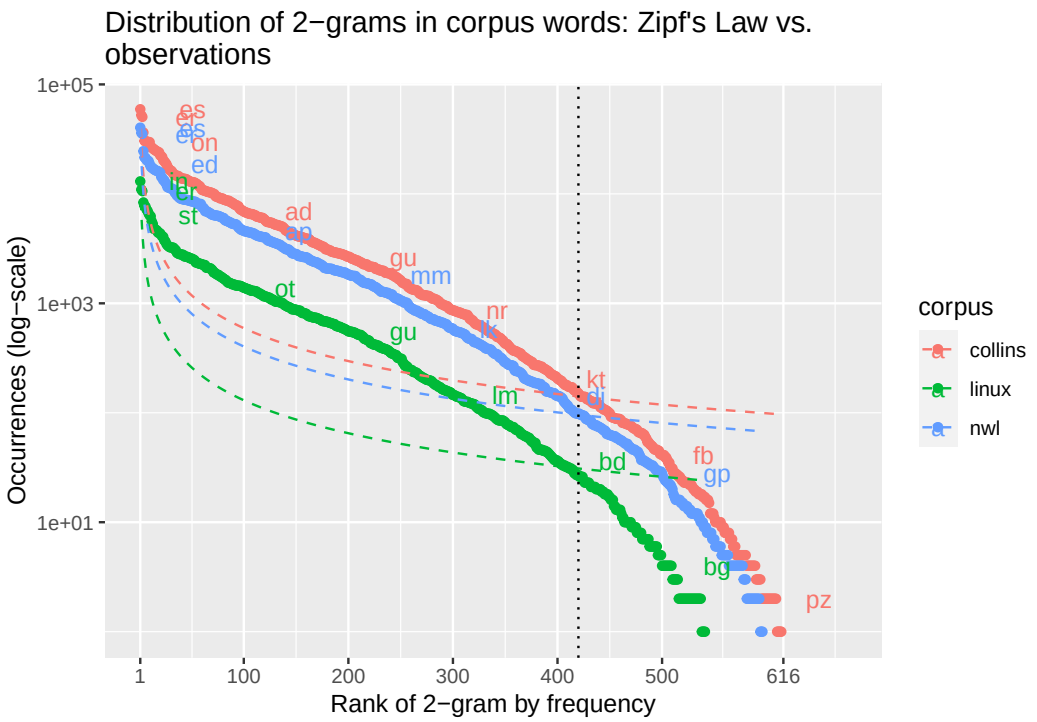

Figure 5: Looking at all 2-grams across the three corpora, distributions of all but the bottom 190 attested instances outperform a Zipf curve.

less often than predicted.

Note that the plot is log-scaled along the y-axis; to the eye, the observed distribution appears almost linear between ranks 100 and 400. It would appear, then, that frequency experiences a *logarithmic* correlation to rank, rather than the power law demanded by Zipf. Outside of these boundaries, however, frequency drops sharply with rank, indicating a greater disparity in frequency among 2-grams in these ranges.

Split pairs

Split distributions We now examine the properties of a very particular sort of 2-gram: split pairs. Given that a difference in these substrings is sufficient to differentiate at least two words, it follows that more common splits (i.e., either half of a split pair) would serve as divisors among a greater variety of corpus words. Figure 6 below is read by choosing the first letter in the split by row, then concatenating it to the letter value of the column. Looking at the heatmap labelled `total` in the bottom-right corner, we see a small handful of high-frequency pairs, shown in yellow: "er", "in", "ng", "re", "st". Curiously, the only letters to combine with all others, in either the first or second position, are the orthographic vowels *a*, *e*, *i*, *o*, and *u*. Individual corpora may have one or more cells missing, but all the corpora combined account for every combination. Likewise, letters like x only attach as the second split letter to vowels, this time also including the semivowel y.

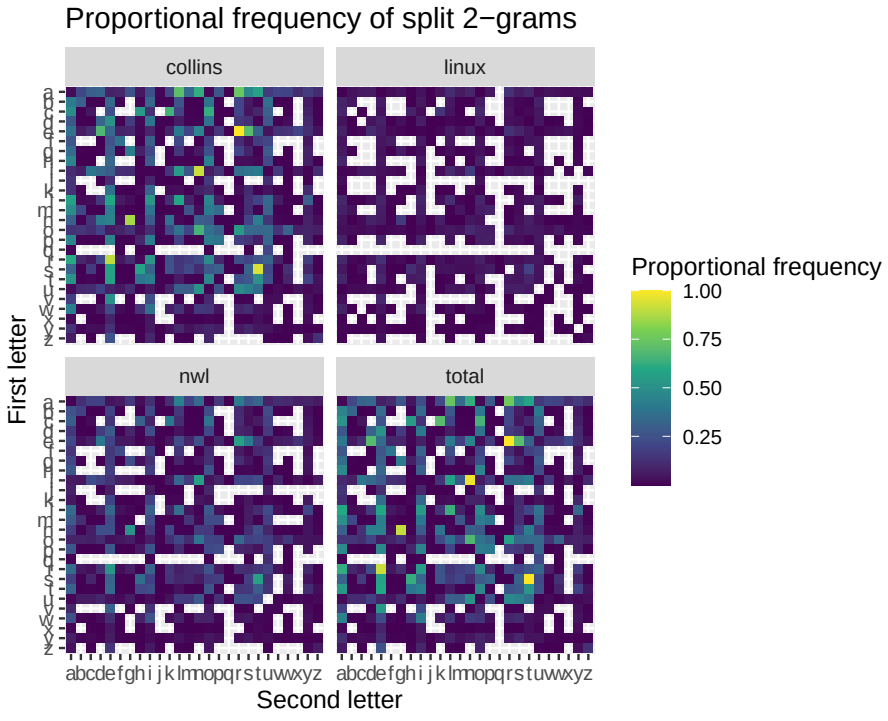

Figure 6: Split pairs are predominated by a few major players: es, ng, re, st, etc.

Positioning Most of the aforementioned high-frequency splits, specifically "er", "ng", "re", and "st", all appear as part of English bound morphemes; that is, units of meaning that can only be used by attaching them to more self-contained concept-words. For example, the word "unlocking" consists of three morphemes: the free morpheme "lock", prefixed by the bound morpheme "un-" and suffixed by "-ing". These bound morphemes, when interpreted in the correct order, transform the utterance's meaning to "the act of removing the status of 'locked' from something". However, we cannot confirm the usage of these substrings in their morphemic sense without first determining where in a given word pair they appear (e.g., STRO(NG/KE)).

We annotate every split by assorting them into one of three positional values. If the split occurs at the beginning of the word, as in (BA/CO)RN, both BA and CO are registered as two instances of a *prefix*-type split; if at the end, as in "FO(ND/XY)", as *suffix*-type splits; and as *middle*-type splits otherwise. Table 2 below depicts the 10 most-frequent splits at each position, across all corpora.

Many of the top splits in the prefix and suffix columns correspond to common English bound morphemes, which may explain their frequency: given some other word as a free morpheme, prepending or appending these splits (potentially along with extra common letters) can often create a new word with a modified meaning. Of particular significance are the "ng" and "on" splits, which Table 1 notes is the most common pair across all three corpora. Prefixed with a verb and an "i", these two suffixes decline the verb into a gerund and a noun, respectively: e.g., "activati**on**", "activati**ng**". Since many verbs are able to take on both of these forms using these suffixes, the frequency of the splits (especially as a pair) is tied directly to the number of verbs in a corpus. Similarly, suffixes like "-er" and "-st" are often found in comparative and superlative adjectival phrases ("funni**er**", "funnie**st**"). Although transformations like these differ in resultant length and thus cannot form pairs with each other, each can attach to many adjective stems and pair with a wider breadth of other words, whether other declensions of the same stem, e.g., funni(er/ly), or the same declension of a different stem, e.g., fu(nn/zz)iest. In effect, each highly-productive[4] bound morpheme serves as something as a multiplier on the distributional effects of different parts of speech.

Table 2: Most common splits by word position.

Rank	prefix (875010 total)	middle (1428766 total)	suffix (340726 total)
1	co (16759)	in (21543)	es (16544)
2	re (16740)	ck (19792)	ed (15443)
3	ba (11992)	ar (19450)	er (15269)
4	ca (11639)	ll (18567)	ng (15161)
5	bu (11179)	il (16974)	on (8717)
6	wa (11159)	tt (16750)	ly (8398)
7	pa (11154)	at (16473)	ts (8202)
8	ra (10910)	er (16388)	al (7453)
9	ma (10803)	st (16239)	st (6921)
10	bo (10595)	an (15413)	ve (5996)

[4]Meaning that the morpheme attaches readily to a broad number of words. Conversely, the plural suffix "-ren" (as in "children") would be a low-productive morpheme.

Again comparing the frequency distributions of splits, this time by position type instead of by corpus, against Zipf's Law yields another logarithmic correlation (note the log-scale in Figure 7). That is, the observed distribution overperforms compared to the Zipf model before a particular inflection point, after which it begins to underperform with increasing error. Here, sheer combinatorics produces something of a visual confounding effect: middle splits outnumber both prefixes and suffixes by an order of magnitude, and encompass almost two hundred more 2-grams. As a result, we do not observe the same x-intercept as in Figure **??**, in which all three distributions seem to intersect their ideal Zipf curves (dashed lines) at the same rank. We again observe that prefixes and suffixes have this intersection at roughly the same point, around rank 255, but the middle-type splits have their inflection point around rank 445. By drawing dotted lines at the intersections of these distributions with the Zipf model, we see that the two intersections occur approximately 190 ranks apart and in both cases separate roughly 100 of the least frequent splits into a 'tail' of sorts.

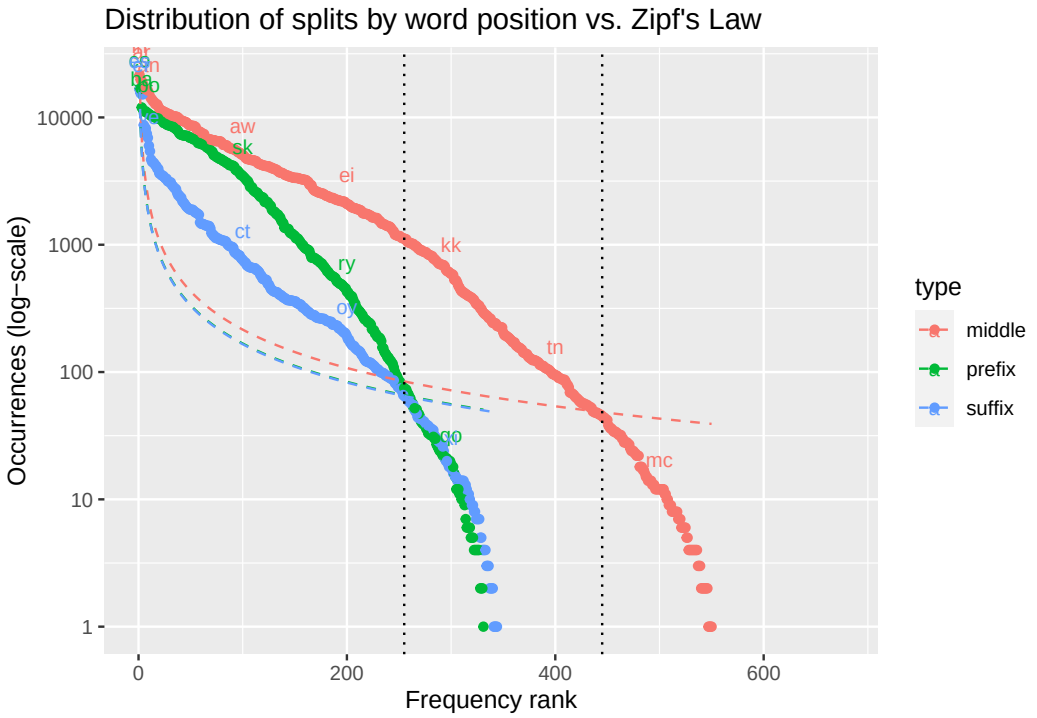

Figure 7: After the high-performing morphemes at the top of the rankings, frequency appears to fall logarithmically with rank.

Discussion

When discussing the data and observations covered in this paper, it is important to remember that our corpora only possess enough data to suggest the *existence* of a word, and speak nothing to its usage. Even though we used the Split Decisions game as a jumping-off point for our examination, a plurality of these words would be ineligible in such a puzzle not due to the properties of its letters, but because of its guessability and fun. Many words, in

particular those that appear in the Scrabble wordlists, are esoteric in the extreme and would likely be considered too rare to be reasonably guessed by the average crossword player. Or, a word pair could contain an extremely vulgar or offensive phrase. Or – and this is likely the greatest disqualifier – the split pair would be too generic, and too many words of the correct length could feasibly serve as fill. For example, 1,084 word pairs across the three corpora are seven-letter words ending in a ER/NG split: baggier/bagging, bulkier/bulking, tastier/tasting... With limited possibility for cluing available on the puzzle page, these clues would not be entertaining to solve, eliminating a large proportion of our vocabulary from real-life use.

That being said, our exploration does lay groundwork for a future study, one better-equipped to handle the quantitative aspects of written and spoken English, in what sort of psycholinguistic models are used to guess pairs of words based on Split Decisions-style cluing. Through insights acquired both through writing this paper and through experience of playing Split Decisions puzzles, a few major constraint systems that transparently underly English speech and orthography, as well as some transformative ortho-phonological effects that can affect one side of a split pair, but not the other.

Instruction of all following linguistic theories and axioms is accredited to the University of Toronto Department of Linguistics c.2018–2020. Teaching staff include but are not limited to: Marisa Brook, Susana Béjar, Keir Moulton, Nathan Sanders, Guillaume Thomas, and Peter Jurgec.

Orthotactics

Like all writing systems, written English is underlain by a complex system of rules and heuristics that, in part, dictates its aesthetic. To best understand these systems, it is often useful to encounter an example that violates these rules, and then to interrogate one's reactions to the failure as a native speaker. Take, for example, the Welsh loanword *cwtch*, meaning something small and intimate. To the anglophone eye, *cwtch* simply "looks wrong", for any number of articulable reasons. One such reason may be the absence of one of the "aeiou" sequence English speakers usually associate with vowel sounds. Another could be the substring *cw*, which is extremely uncommon in English orthography and otherwise unattested as the first two letters of a word (or indeed outside of compound words)[5].

In many cases, being familiar with these English orthography rules, however subtle, can supplement solver's ability to deduce possible words. For example, if a word with the following 'template' appears in a puzzle:

$$-\binom{EE}{TR}---$$

There are only select few letters that can reasonably appear as the first letter of the fill, with the remainder being logically eliminated by one or both of the splits. The most likely letter altogether, as a keen reader has likely deduced by now, is *s*; many common English words of similar length begin with "see-" or "str-". Many other letters are quickly eliminated: the first letter is unlikely to be a vowel, as three-vowel sequences at the beginning of words

[5]Other words containing *cw* in Collins and NWL include: *colicweed(s)*, *cwtches*, *cwtching*. *cwtched*, *rusticwork(s)*, *plasticware(s)*.

is rare.[6] Other continuant consonants, like *f*, *z*, *h*, or *r*, conflict instead with the TR split and are thus incompatible, despite no string of letters being ostensibly impossible to write. Conversely, we observe that the first letter at the right boundary of the fill *must* be a vowel: assuming that the first letter is *s*, the consonant cluster *str* is as long as either written or spoken English will allow at the beginning of a syllable. But, the vowel must not be *e* to avoid an *eee* substring... This deduction, powered entirely by one's intuition about English spelling, continues until a solution is found, most likely (SEEING, STRING). Take inventory of which of these propositions you found yourself inherently agreeing with, even in absence of some objective reference for which substrings are or are not attested in English words.

This sort of constraint-based reasoning about spelling is generally referred to as *orthotactics*. In general, having a solid understanding of English orthotactics benefits the solver of a Split Decisions puzzle, as it helps reduce uncertainty about missing letters. This effect compounds with the crossword-like nature of the puzzle: as each word pair is solved, it also introduces letters into one or more crossing pairs, which further propagates constraints in the fill.

However, to introduce greater variety and complexity, puzzle setters can also rely on special categories of words whose properties allow them to violate these orthotactic rules more frequently. One of these categories is loanwords like the aforementioned *cwtch* (from Welsh) or *bazaar* (from Persian), both of which visibly violate English's rules about vowel count. This is especially true of loanwords from other languages that use the Latin alphabet: French, Italian, Finnish, etc. Latin-scripted words are more likely to be taken wholesale without modification, whereas the transfer from another writing system into English often affords opportunity for ortho-phonological repair which reifies the spelling and pronunciation of a word to something more normatively English. Another such category would be compound words, which contain more than one free morpheme concatenated together. A common word pair, found in several of Piscop (2021)'s puzzles, is B(AR/ED)ROOM, consisting of two compound words that describe rooms of a house by their contents. Morpheme boundaries can also cause categorical ambiguity for solvers by pairing words that differ in the transparency of their compounding: although most English speakers readily identify PREWAR as consisting of two morphemes – *pre-*, meaning 'before', and *war* – but less readily recognize the same *pre-* morpheme in the function word PREFER[7]. As a result, the PRE(FE/WA)R pair provides a novel dimension of challenge to new solvers.

Our results showed that bound morphemes, especially suffixes like *-er* and *-ing*, were predominantly frequent across all three corpora. As previously mentioned, a great many of these declensions produce pairings that are not particularly novel to solve, e.g., SCARI(ER/NG). However, we can again see that combinations of words in which said suffix functions differently in each can create challenging fills, specifically by forcing solvers to consider the clued substring both in its grammatical and non-grammatical context. An example was provided at the beginning of this section in S(EE/TR)ING: the -ING suffix simultaneously functions in its common gerund form (in SEEING) and as a simple segment of a non-verb word (STRING).

[6] Examples: *aeons*, *aeolian*, *aioli*, *ouija*, *oeuvre*.

[7] Etymology Online (link): from Latinate *pre-* and a Proto-Indo-European (PIE) root which arrived in English as *bear* (i.e., to carry); roughly "to choose to bear first".

Phonotactics

Similar to the orthotactics of written language, spoken or signed languages also operate under a matrix of rules and constraints known as *phonotactics*, which defines what sorts of utterances are permissible as potential words. This concept is explored in Andrew Clement's 1998 children's novel *Frindle*, in which a young boy rises to celebrity by coining the titular term as a novel synonym for "pen". Indeed, *frindle* appears to be both ortho- and phonotactically felicitous to English speakers, with any markedness (i.e., speaker hesitance) being brought about primarily by a lack of previous attestation for the word. A facetious term for this phenomenon, coined by David Cohen of *The Simpsons*, is *cromulent*: "*embiggens* is a perfectly cromulent word!" Indeed, the word "cromulent" is so cromulent that, despite appearing only once on the animated sitcom, native English speakers are usually able to readily understand what is meant by the declined form *cromulence* (i.e., the status of being cromulent).

Note that, in writing systems like that of English, where the written form of a word at least partly attempts to reflect its pronunciation, there is a strong interaction between the phonotactics of the spoken language and the orthotactics of its written variant; we have referred to this interaction as *ortho-phonotactics* at varying points through this paper. Ortho-phonotactic repair is frequently seen when adapting loanwords from languages that use phonemes or other linguistic features unfamiliar to English: consider the breadth of pronunciations and spellings of the holiday *Hannukkah*.

As previously mentioned, there is nothing inherently unpronounceable about many complex clusters of consonants like, say, /tln/. Instead, this notion of pronounceability is largely defined by the languages one speaks, and thus varies widely among speakers. For the purposes of an English-language puzzle, however, taking advantage of these expectations – what sorts of substrings are expected to appear in a word and where – allows puzzle setters to wield a surprisingly granular amount of control over the difficulty of a puzzle. For example, the split ZJ is exceedingly rare as a 2-gram, causing most English speakers to readily identify it as indicating either a phoneme from a loanword or the nexus of a compound word. Indeed, the 2-gram appears in exactly one word in both the `collins` and `nwl` corpora: BIZJET, a shortened form of "business jet(plane)". Conversely, cluing a common 2-gram that usually appears as a cluster, but instead the two letters are separated by a morpheme boundary, can increase the guessing of difficulty by forcing solvers to consider the two letter-sounds separately. Example: FATHE(RS/AD). Notice how in one side of the split, the TH 2-gram represents the usual dental fricative sound, but in the other, it represents a /t/ and an /h/ separately.

Local phonological interactions

No phoneme is an island. Its articulation (i.e., exactly how it is pronounced out of a speaker's mouth) is directly impacted by its phonological context, and in turn that phoneme's presence affects the phonological context around it. Consider the difference in pronunciation of the final consonants in *booth* vs. *booths*. Although even very young children readily identify that the plural *-s* is pronounced /z/ when the preceding phoneme is voiced, and /s/ otherwise (**???**), the final consonant in *booth* is unvoiced unless the word is pluralized, in which case both the dental fricative (the 'th') and the /s/ change to their voiced variants. This phenomenon of two adjacent phonemes interacting with each other, with one or both mutating as a result, is generally referred to as a *local phonological interaction*. Although these interactions are

not as marked in English as, say, the vowel harmony system of the Uralic languages, their presence serves as another ortho-phonological avenue for controlling the guessability of word pairs. In terms of local interactions, we use the terms of art *regressive* and *progressive* to refer to interactions affecting preceding and succeeding phonemes, respectively.

One such regressive interaction in English is vowel nasalization: when a vowel either directly precedes or directly follows a nasal consonant, one of /n/, /m/, or /ŋ/[8], it is articulated with a retracted velum, allowing airflow through the nasal passage. This is why, for example, the words *bat* and *ban* do not slant-rhyme (share a common vowel sound). A similar regressive interaction happens before /r/, called *r-coloring*, which can significantly alter the pronunciation of a vowel (*bun*, *burn*). Regressive interactions can induce a great deal of difficulty at the left boundary of a split pair, by differentiating the pronunciation of a preceding vowel between the two sides. Consider, for example, the pair PRESE(NT/RV)ED, in particular the difference in pronunciation of the left-boundary *E*.

Figure 8 below depicts left boundaries: that is, the transition between the last letter of the common substring before a split, and the first letters of said split. For visual clarity, only those boundaries that occur more than 400 times are shown, and some of the more dominant declension suffixes have been removed. The thickness and opacity of lines is proportional to their frequency. Along the left edge, it can be seen that most common left boundaries start with a vowel, but the respective first letter of the split varies widely. However, a few major outliers can be seen. One is the connection between *i* and *e*; others stem from *o* to *o*, *u*, or *w*. Although less visible among other nearby letters, *n* receives many of these vowels from the left boundary.

Although, again, our corpora have not been curated for what sorts of words might be more viable in a real Split Decisions puzzle, these points tell an interesting narrative of ortho-phonological interaction. Indeed, consider the following sets of words:

- piece, piety, lies, lien
- poor, blood, foot, loon
- pour, lout, would, through, cough, counter
- bowl, cowl, town

Despite sharing the same two-vowel substrings, each of these words' phonetic vowels are articulated differently. Even for the last word in each set, which differs by its syllable-final nasal (here *n*), the vowel or diphthong is articulated subtly differently owing to the aforementioned nasalization effect. In short, a large set of common words share these common bivocalic left boundaries, but vary widely in their pronunciation. A future study may be interested in how this psychoacoustic difference between similarly-spelled words may affect the priming time of recalling these two words as a pair, and subsequently deducing that they serve as the unambiguous solution for a given Split Decisions puzzle. Such an analysis would also be interested in, for example, the usage frequency and commonality of different words, and perhaps would also employ lemmatization algorithms to differentiate functional vs. non-functional suffixes more directly.

[8]The velar nasal; appears at the end of "hang" or "thinking".

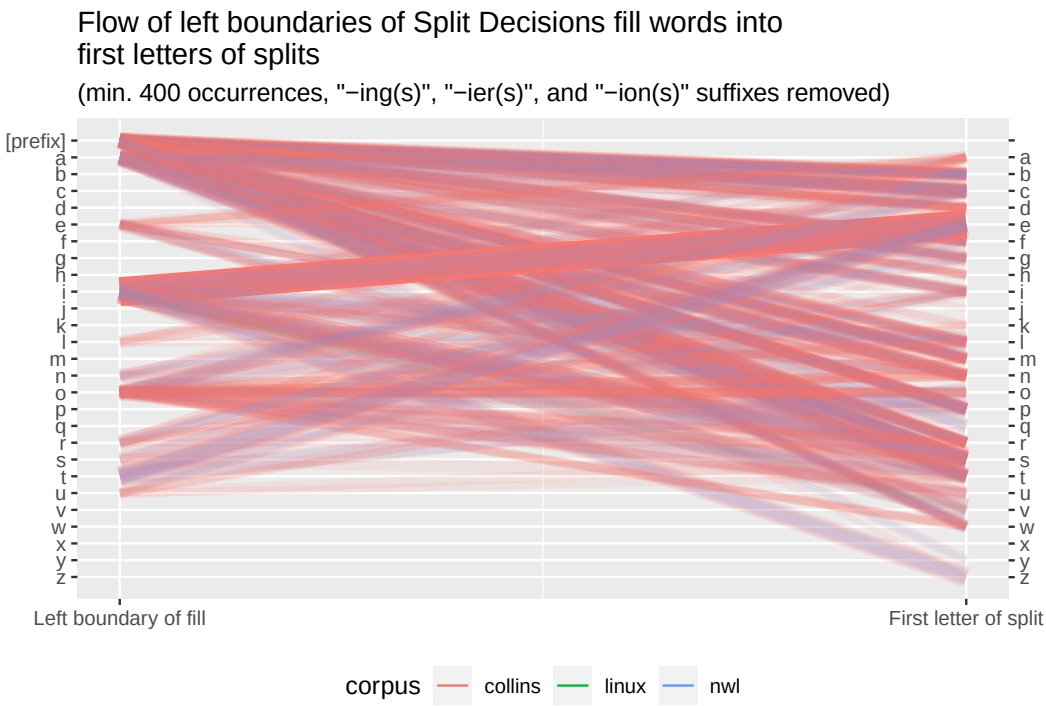

Figure 8: Even excluding some of the most common declension suffixes, vowels are extremely common on both sides of a left split boundary.

Appendix

Link to Python preprocessing script

Found in the GitHub repo (link).

References

Collins Official Scrabble® Words. 2019. Harper Collins.

NASPA Dictionary Committee. 2020. *NASPA Word List, 2020 Edition*. NASPA.

Piscop, Fred. 2021. *Split Decisions*. Split Decisions.

R Core Team. 2021. *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. https://www.R-project.org/.

Wickham, Hadley. 2016. *Ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York. https://ggplot2.tidyverse.org.

Wickham, Hadley, Mara Averick, Jennifer Bryan, Winston Chang, Lucy D'Agostino McGowan, Romain François, Garrett Grolemund, et al. 2019. "Welcome to the tidyverse." *Journal of Open Source Software* 4 (43): 1686. https://doi.org/10.21105/joss.01686.

Xie, Yihui. 2021. *Knitr: A General-Purpose Package for Dynamic Report Generation in R*. https://yihui.org/knitr/.

Zipf, George. 1949. *Human Behavior and the Principle of Least Effort*.