Glottometrics

58/2025

International Quantitative Linguistics Association

Glottometrics

Vol. 58, 2025

Glottometrics is an open access scientific journal for the quantitative research of language and text published twice a year by International Quantitative Linguistics Association (IQLA). The journal was established in 2001 by a pioneer of Quantitative Linguistics Gabriel Altmann.

Manuscripts can be submitted by email to glottometrics@gmail.com. Submission guideline is available at https://glottometrics.iqla.org/.

International Quantitative Linguistics Association (IQLA)

Friedmanngasse 50
1160 Vienna
Austria

eISSN 2625-8226

Contents

How effective is OpenAI to write speeches for the US president?

Jacques Savoy[1*]

[1] University of Neuchatel, Neuchâtel, Switzerland

[*] Corresponding author's email: Jacques.Savoy@unine.ch

DOI: https://doi.org/10.53482/2025_58_422

ABSTRACT

Using large language models (LLMs), computers are able to generate a written text in response to a user request. As this pervasive technology can be applied in numerous contexts, this study analyses the written style of one LLM called GPT developed by OpenAI by comparing its generated speeches with those of the recent US presidents. To achieve this objective, the *State of the Union* (SOTU) addresses written by Reagan to Biden are contrasted to those produced by both GPT-3.5 and GPT-4.o versions. Compared to US presidents, GPT tends to overuse the lemma "we" and produce shorter messages with, on average, longer sentences. Moreover, GPT opts for an optimistic tone, choosing more often for political (e.g., president, Congress), symbolic (e.g., freedom), and abstract terms (e.g., freedom). Even when imposing an author's style to GPT, the resulting speech remains distinct from addresses written by the target author. Finally, the two GPT versions present distinct characteristics, but both appear overall dissimilar to true presidential messages.

Keywords: political speeches, large language models, stylometry, ChatGPT, authorship.

1 Introduction

With the development of large language models (LLMs) (Zhao *et al*., 2023), generative AI demonstrates its capability to generate a short text in response to a user request. Currently, such applications are freely available and can help users produce various types of writing (e.g., e-mail, CV, short letter, etc.). From this perspective, this study investigates the writing style of GPT developed by OpenAI when asked to generate *State of the Union* addresses for a president. Annually expressed in front of Congress, these speeches explain the world situation and political agenda of the occupant of the White House. The main objective is to inform and persuade the audience that the propositions and actions of the president are the most appropriate. To reach such an objective, the style and rhetoric play an important role in reinforcing the president's words.

Based on recent developments in automated text analysis designed by communication and psychological scholars (Jordan, 2022), this study analyses the style and rhetoric of six US presidents (Reagan, Clinton, Bush, Obama, Trump, and Biden) as well as that of two GPT versions (GPT-3.5 and GPT-4.o). In this study, rhetoric is defined as the art of effective and persuasive speaking, and the way to adopt a

tone to motivate an audience. An author's style is evaluated through studying frequent forms employed to support his/her communication objective (Biber & Conrad, 2009).

To author a SOTU speech, a chief ghostwriter collaborates more or less closely with the president[1]. Could we employ GPT to achieve a similar objective and expect that it could adopt a political tone and style of the current occupant of the White House? In the end, can we still discriminate between the generated address and the real one? If so, what are the stylistic characteristics that differ between the two speeches? Moreover, what are the rhetoric features that can be pertinent to discriminate between the addresses written by several presidents (Reagan, Clinton, Bush, Obama, Trump, and Biden)? Additionally, can we observe distinct aspects between the two GPT versions and, if so, which one is the best to write a political message?

To address these questions, this article is organised as follows. The first section presents some related work, while Section 3 describes the corpus used in our experiments. Section 4 analyses some stylistic features by comparing those in both GPT versions to those occurring in speeches written by US presidents. Additional experiments focusing on psychological and emotional characteristics are depicted in Section 5, while the next evaluates the global similarity between each president and the two GPT versions. Finally, a conclusion reports the main findings of this study.

2 State of the Art

Numerous studies have been published on authorship attribution and on recognising author demographics characteristics (e.g., gender, age, social status, native language, etc.) (Kreuz, 2023). Other stylometry studies have additionally been performed on the detection of plagiarism or fake documents, the identification of suspects in criminology (Olsson, 2018), the determination of text genre, and even the dating of a document. To resolve these questions, various natural language processing models have been applied by scientists from different domains such as computer science (Savoy, 2020), (Karsdorp *et al*., 2021), linguistics (Crystal, 2019), (Yule, 2020), psychology (Pennebaker *et al*., 2014), (Jordan, 2022) and communication studies (Hart *et al*., 2013), (Hart, 2020).

The main objective of this study is to analyse the style and rhetoric of true political speeches and to compare them with those automatically generated by GPT. This emerging technology is based on LLM (large language model) technology grounded on a deep learning architecture (Goodfellow *et al*., 2016), which is based on a sequence of transformers with an attention mechanism (Vaswani *et al.* 2017). The most important notion to understand LLM is the following: given a short sequence of tokens (e.g., words or punctuation symbols), the computer is able to automatically supply the next token. More

[1] For example, at https://www.youtube.com/watch?v=zFbaesLEa4g, Obama's ghostwriter, J. Favreau, comments his job.

precisely, knowing four tokens, the model must first determine the list of possible next tokens to complete the given sequence (Wolfram, 2023). For example, after the chain "the president of the" the computer, based on the training documents, can define a list of the next occurring token, such as United, Philippines, Senate, US, USA, UK, republic, Ukraine, and so forth.

From this list, and depending on some parameters, the system can then select the most probable token (in our case, "United") or based on a uniform distribution, one over the top *k* ranked tokens (e.g., "Senate"), or randomly depending on their respective probabilities of occurrence in the training texts (e.g., "US"). This non-deterministic process guarantees that the same request will produce distinct messages. Common to all LLMs, GPT may include hallucinations in its answers (namely, incorrect information). In our previous example, the sequence "the president of the UK" should be replaced by "the Prime Minister of the UK"). Moreover, the specification of the sources exploited to produce the text remains unknown[2].

As previously mentioned, the main target application of such LLMs is to generate a short text in the context of a dialogue. To analyse such automatically generated texts, different studies expose the effectiveness of several learning strategies capable of discriminating between answers generated by GPT-3.5 and answers written by human beings (Guo *et al.*, 2023). Based on a classifier trained on a given domain (e.g., RoBERTa), the recognition rate is rather high (around 95% to 98%). Such effectiveness is also obtained when the target language is not English (e.g., French (Antoun *et al.*, 2023)), or when it is Japanese (Mizumoto *et al*., 2024). Such a high degree could be reduced when faced with a new and unknown domain or when substituting tokens by misspelled words (in such cases, the achieved accuracy rate varies from 28% to 60%). Of course, the message must include at least 1,000 letters to allow the detection system to reach such a small error rate.

With a similar objective, the CLEF-PAN 2019 international evaluation campaign evaluated different systems to automatically detect whether a set of tweets was generated by bots or by humans (Daelemans *et al*., 2019). In this case as well, the effectiveness was rather high (between 93% to 95% for the best approaches). However, the tweets written by bots were not produced by a LLM, but corresponded to messages either containing a well-known citation, a passage of the Bible, or text corresponding to a predefined pattern (e.g., list of positions available in a large company).

3 Corpus Overview

To ground our conclusions on a solid basis, the same text genre has been selected: namely, written speeches given in the same context, to achieve similar objectives, and written in the same time period.

[2] The training sample employed by GPT is not precisely known and one might assume that many presidential speeches have been included.

To compare the style of recent US presidents with messages created by a machine, we queried the GPT API (Application Programming Interface) to generate the *State of the Union* (SOTU) addresses for six presidents, namely Reagan, Clinton, Bush, Obama, Trump, and Biden. For each US leader, only the SOTU addresses were taken into consideration. In addition, two versions of GPT were used, namely version 3.5 and 4.o (or 4.omni). As shown in Table 1, the number of SOTU speeches varied from three (Biden) to eight (Clinton, Bush, Obama).

Table 1: Some statistics on our American corpus.

	Presidency	Number	Tokens	Types	Mean length
Reagan-GPT-3.5		70	29,381	1,074	414.7
Clinton-GPT-3.5		80	42,125	1,385	528.0
Bush-GPT-3.5		70	35,756	1,254	504.9
Obama-GPT-3.5		80	32,224	1,340	539.7
Trump-GPT-3.5		40	19,616	1,033	484.5
Biden-GPT-3.5		30	15,282	977	489.3
Reagan-GPT-4.o		70	45,651	1,221	643.1
Clinton-GPT-4.o		80	55,085	1,275	680.8
Bush-GPT-4.o		70	45,665	1,277	643.9
Obama-GPT-4.o		80	52,557	1,414	649.2
Trump-GPT-4.o		40	25,049	1,027	614.4
Biden-GPT-4.o		30	19,879	941	640.1
R. Reagan	1981–1989	7	32,490	3,384	3,975.4
B. Clinton	1993–2000	8	59,705	3,835	6,520.5
W.G. Bush	2001–2008	8	40,532	3,514	4,349.5
B. Obama	2009–2016	8	53,777	3,902	6,021.0
D. Trump	2017–2020	4	22,189	3,200	3,973.8
J. Biden	2021–2024	3	25,598	2,912	5,778.0

To help both GPT versions in their generative process[3], the true SOTU address of the corresponding year was included in the prompt. In addition, a short list of possible topics was inserted (e.g., "deregulation, free market, reduced taxes, small government, education, middle-class, security, …"). Finally, the prompt[4] specified the president's name and year to obtain a message written according to the style of a specified leader. For example, for 1982, the prompt included the following sentences:

[3] The training sample used by GPT is unknown but one can assume that many presidential speeches have been included. However, those messages, if appearing in the training set, are employed to define the occurrence probability of a token, given the four previous ones, and not to identify a presidential style.

[4] All the prompts are available at https://drive.switch.ch/index.php/s/pzkraoobJWu7xqP. Moreover, the parameters have been fixed as follows: temperature=0.5, frequency_penalty=0, presence_penality=0, top_p=0.4, max_tokens=32768.

> "I'm Ronald Reagan, President of the United States of America. I need to write my SOTU speech. Can you write a SOTU speech to be presented in the front of the Congress in January 1982.…"

As GPT generates relatively short messages, ten different versions for each speech have been generated for both versions. As shown in the Appendix, this limit of ten seems problematic for OpenAI, particularly when generating political speeches.

Table 1 depicts a general overview of our political corpus. The third column indicates the number of speeches. The total number of tokens (labelled "Tokens") and the number of distinct words (labelled "Types") are reported in the next columns. These values are computed without counting the numbers and the punctuation symbols.

The last column shows the mean number of tokens per speech. The average size of the GPT versions is roughly ten times smaller than the real ones. When comparing both GPT versions, the overall mean length is 493.5 for GPT-3.5 and 645.3 with GPT-4.o, a significant difference (bilateral *t*-test, significance level 1%). In total, this corpus contains 652,561 tokens, with 418,270 created by both GPT versions and 234,291 belonging to true SOTU addresses.

4 Stylometric Analysis

As a first stylometric measure, one can focus on the language complexity that all political leaders tend to reduce. For example, L. B. Johnson (presidency: 1963–1969) specifies to his ghostwriters, "I want four-letter words, and I want four sentences to the paragraph." (Sherrill, 1967). The complexity of the language could be measured by the mean number of letters per words. In this case, the larger the mean, the higher the language complexity.

As an additional characteristic, we count the percentage of words composed of six letters or more, defined as big words (BW) in the English language. We observe, for example, that depending of the length of words, some are easier to understand than others. It is the difference between "ads" and "advertisements", for example, or "desks" and "furniture". Such a relationship between complexity and word length is clearly established:

> "One finding of cognitive science is that words have the most powerful effect on our minds when they are simple. The technical term is basic level. Basic-level words tend to be short. … Basic-level words are easily remembered; those messages will be best re-called that use basic-level language." (Lakoff & Wehling, 2012)

Finally, we evaluate the mean sentence length (MSL). It has been observed that long sentences tend to render the speech more complex to understand. Table 2 depicts these three measurements individually for each president, and globally for both GPT versions. Moreover, in the last row, the average over the

six presidents is shown by concatenating all their SOTU addresses. In this table, the largest values are presented in bold and the smallest in italics.

According to values shown in Table 2, GPT-3.5 presents the language with the highest complexity on the three measurements. On the other hand, Biden presents both the smallest mean of letters per word and the smallest MSL. Between the two GPT versions, we observe that version 4.o clearly reduces the mean word size and the percentage of BW. Both values are still higher than the mean value over the six presidents (4.85 vs. 4.44, and 37.8% vs. 28.7%). The MSL of GPT-4.o corresponds clearly to possible presidential speech (19.95 vs. 19.45).

Table 2: Statistics on three language complexity measurements.

	Mean word length	Big words	Mean sentence length
GPT-3.5	**5.07**	**40.11**%	**21.56**
GPT-4.o	4.85†	37.80%†	19.95†
Reagan	4.49†‡	29.34%†‡	21.45‡
Clinton	4.34†‡	26.79%†‡	21.33‡
Bush	4.50†‡	30.13%†‡	20.07†
Obama	4.31†‡	*25.94*%†‡	19.72†
Trump	4.55†‡	30.74%†‡	17.73†‡
Biden	*4.29*†‡	25.95%†‡	*15.72*†‡
Presidents	4.44†‡	28.70%†‡	19.45†‡

To statistically determine whether a given mean could be viewed as different than that produced by GPT-3.5, a bilateral *t*-test (Conover, 1990) has been applied with the null hypothesis H_0 specifying that both population means are equal. For example, in Table 2 GPT-3.5 produces an average word length of 5.07 letters. Reagan pronounces on average 4.49 characters per word. This difference (5.07 – 4.49 = 0.58) must be viewed as statistically significant (significance level $\alpha = 1\%$), and this statistical significance is indicated by a single cross (†). Moreover, GPT-4.o presents a mean value of 4.85. This difference, compared with Reagan's mean, is also statistically significant (significance level $\alpha = 1\%$), and is denoted by a double cross (‡). With the BW values, the proportion test (Conover, 1990) has been applied instead of the *t*-test with the same significance level.

When comparing the two GPT versions, Table 2 shows that for the three measurements, GPT-4.o results in a lower language complexity, and the differences are always statistically significant compared to GPT-3.5. The newest version presents a reduced language complexity, closer but not similar to true presidents. As displayed in Table 2, the differences with GPT-3.5 are always statistically significant, as well as with the mean over all presidents. When comparing with GPT-4.o, the differences are usually always statistically significant.

When analysing a written style, the words can be divided into content and function terms with nouns, main verbs, adjectives and adverbs belonging to the first class. Function (or glue) words corresponding to pronouns, articles, prepositions, auxiliary verbs and conjunctions are more frequent and tend to reflect some stylistic characteristics. In particular, some stylistic and psychological traits of the author can be derived by analysing the relative frequencies of pronouns (Pennebaker, 2011; Kacewicz *et al*., 2014).

In this regard, the occurrence frequencies of personal and impersonal pronouns (e.g., it, that) (denoted *Ipron*[5]) are displayed in Table 3. The last row shows the percentage of pronouns when concatenating all presidential speeches and can be viewed as a mean usage for a president in power. As for the previous table, the largest values appear in bold and the smallest in italics. In addition, the proportion test has been applied with significant difference ($\alpha = 1\%$), denoted by † over GPT-3.5 or by ‡ over GPT-4.o.

Table 3: Frequency of occurrence of pronouns.

	Self	*We*	*You*	*She/he*	*They*	*Ipron*
GPT-3.5	1.05%	6.93%	0.56%	0.00%	0.77%	3.59%
GPT-4.o	*0.68%*†	**8.30%**†	*0.46%*†	*0.00%*	*0.58%*†	*3.57%*
Reagan	1.03%‡	4.25%†‡	0.50%	0.23%†‡	0.84%‡	4.42%†‡
Clinton	1.55%†‡	4.45%†‡	0.77%†‡	0.33%†‡	**1.31**%†‡	4.67%†‡
Bush	0.96%‡	4.11%†‡	0.64%†‡	0.29%†‡	1.18%†‡	3.69%
Obama	1.32%†‡	4.28%†‡	0.55%‡	0.41%†‡	1.12%†‡	**5.64**%†‡
Trump	1.16%‡	4.17%†‡	0.80%†‡	**0.90**%†‡	0.93%†‡	3.73%
Biden	**1.98**%†‡	*3.33%*†‡	**1.37**%†‡	0.64%‡	1.26%†‡	4.65%†‡
Presidents	1.20%†‡	4.22%†‡	0.67%†‡	0.41%†‡	1.04%†‡	4.51%†‡

With the *Self* (I, me, mine, myself) category, GPT-4.o displays the smallest proportion of I-words while version 3.5 exposes a value close to that of some presidents (e.g., Reagan, Bush, or Trump). For a leader in an electoral campaign, a large proportion of *Self* corresponds to an efficient and successful communication strategy. After all, an election is the process of choosing between two candidates (e.g., US, Canada, France) (Labbé & Monière, 2008), (Savoy, 2018).

The use of we-words (we, us, our, ourselves) appear as a way to move from an individual point of view to a collective one, with a solidarity aspect. From a political communication point of view, this is a significant characteristic. The lemma 'we' is common to all political leaders in power. This pronoun has the advantage of being ambiguous; we are never sure who is behind the 'we'. Is it the president

[5] The term indicating a category is displayed in italics.

and his cabinet, the Congress, or more generally, the president and the people listening to the speech? In this last case, the speaker also wants to establish a relationship with the audience, usually to involve them in the proposed solution. As shown in Table 3, this pronoun is the most frequently employed by all presidents. Both versions of GPT overused it, and the proportion differences with all presidents are significant.

As shown in Table 3, GPT avoids using other personal pronouns. For GPT-4.o, those percentages are the lowest over all rows. We can explain these low rates by the difficulty of establishing the right reference between the referent and the pronoun. This is also true of the impersonal pronouns employed less frequently by the two GPT versions. Another finding is the absence of the third singular personal pronouns with GPT. More precisely, the word 'she' never appears under GPT's pen.

When analysing the differences between presidents, we observe that Biden employs the lemma 'we' less frequently, but presents the highest intensity in the categories of *Self* and *You*. This choice denotes the willingness to establish a relationship between the speaker and the audience. These differences characterise Biden's voice as distinct from those of the other occupants of the White House.

When evaluating two or three personal pronouns, some psychological traits about the author can be perceived (Kacewicz *et al*., 2014). People with higher status consistently use fewer first-person singular pronouns, and they use more first-person plural and second-person pronouns. The power language[6] is associated with attentional biases; higher status is linked with other-focus, whereas lower rank is linked with self-focus (Kacewicz *et al*., 2014), (Pennebaker, 2011). According to this perspective, both GPT versions appear to adopt a high leader status with a high frequency of *We* and *You*, and a low percentage of *Self* (e.g., GPT4.o: 8.3% + 0.46% – 0.68% = 8.08%). Among presidents, the combined frequency of the categories *We* + *You* - *Self* indicates that Trump (3.81%) and Bush (3.79%) embrace a higher social status than the other presidents, with the lowest value associated with Biden (2.71%).

5 Psychological and Emotional Analysis

A psychological and emotional analysis of political speeches can be grounded on LIWC[7]. This text-based analysis system is built around several wordlists according to syntactical, emotional or psychological categories. The main hypothesis is to assume that the words serve as guides to the way the author thinks, acts, or feels (Jordan, 2022). In LIWC, categories may match grammatical categories such as personal pronouns, as well as broader ones (e.g., verbs), or more specific ones (verbs in the past tense, auxiliary verbs). On a semantics level, the LIWC defines positive emotions (*Posemo*) (e.g.,

[6] The power language is used by people higher in power and status (e.g., your boss).

[7] Linguistic Inquiry & Word Count (Tausczik & Pennebaker, 2010).

happy, hope, peace), or negative ones (*Negemo*) (e.g., fear, blam*[8]). With these categories, the emotional aspect (optimism or pessimism) of a speaker can be evaluated. Presidents (or prime ministers) tend to voice positive words more frequently to appeal to the audience and to persuade the public. In particular, populist leaders more often employ emotional terms to incite strong sentiments in the population, usually to obtain a larger media coverage (Obradović *et al*., 2020), (Hart, 2020), (Savoy & Wehling, 2022).

The category *Cogproc* contains terms related to self-reflection (e.g., think, refer*) and causal words (e.g., cause, understand). This measure corroborates with an active thinking and narrative tone (Tausczik & Pennebaker, 2010). Under *Achieve* (e.g., plan, win, lead*, etc.), we evaluate the confidence of the author to resolve or to propose a solution to a problem in a successful way.

As a second approach, Hart *et al*. (2013) have developed the DICTION system, which groups different wordlists specifically created to analyse political messages. For example, in the *Familiarity* category (e.g., a, at, to, with, etc.), we see words that occur in everyday expressions, and that correspond to terms which are easily understood (Ogden, 1968). Such an enumeration corresponds to a stopword list applied by search engines to ignore terms without a clear meaning (Dolamic & Savoy, 2010). When opting for a high level of familiarity, the speaker wants to address his or her message to the entire population using a simple tone. To reinforce this characteristic, the orator could present a lower mean number of letters per words and write short sentences (see Table 2).

More specific to political text analysis, the category *Symbolism* contains terms related to the country (e.g., nation, America), ideology (e.g., democracy, freedom, peace), or generally political concepts and institutions (e.g., law, government). Those expressions are related on an abstract level and are usually employed to express an ideal view of the situation. Additionally, the *Politics* category (e.g., power, republican, majority, federal, etc.) contains concrete terms related to political institutions and parties in the US.

Table 4: Semantic categories over the US presidents and both GPT versions.

	Posemo	*Negemo*	*Cogproc*	*Achieve*	*Familiarity*	*Symbolism*	*Politics*
GPT-3.5	**7.34**%	1.27%	8.12%	**5.51**%	20.06%	**5.24**%	3.94%
GPT-4.o	7.21%†	*0.99*%†	*7.23*%†	4.36%†	*20.01*%	5.37%	**5.40**%†
R. Reagan	4.86%†‡	1.88%†‡	8.93%†‡	2.73%†‡	**22.87**%†‡	3.84%†‡	4.18%‡
B. Clinton	4.20%†‡	1.62%†‡	9.72%†‡	3.08%†‡	22.60%†‡	3.52%†‡	3.37%†‡
W.G. Bush	4.99%†‡	**3.09**%†‡	8.46%‡	2.91%†‡	21.93%†‡	4.10%†‡	4.19%†‡
B. Obama	3.66%†‡	1.73%†‡	**10.31**%†‡	2.86%†‡	22.17%†‡	*3.16*%†‡	*3.10*%†‡
D. Trump	4.29%†‡	2.34%†‡	7.63%†	2.48%†‡	20.83%†‡	4.43%†‡	3.91%‡
J. Biden	*3.33*%†‡	1.74%†‡	9.11%†‡	*2.09*%†‡	21.32%†‡	3.50%†‡	3.31%†‡
Presidents	4.33%†‡	2.11%†‡	9.06%†‡	2.78%†‡	22.04%†‡	3.76%†‡	3.69%†‡

[8] When generating an entry in a wordlist, we use the symbol '*' to denote any sequence of letters.

The percentages of each category achieved by the six presidents and the two GPT versions are reported in Table 4. In the first two columns, both GPT versions employ more positive emotions and less negative ones compared to true presidents. Moreover, the differences with the US leaders are always statistically significant. Between presidents, Bush presents the highest percentages in both positive and negative feelings. In particular, he obtains the highest negative score with terms related to the war in Iraq and terrorists. One may be surprised to not see Trump with the highest percentage of negative terms. This study is based on written speeches, certainly authored by ghostwriters and not the president himself. With Trump, we observe significant differences between his written messages and his spontaneous language (e.g., interviews, press conferences, tweets) (Savoy & Wehren, 2022).

With terms occurring in the *Cogproc* category, GPT-3.5 portrays a percentage similar to Bush. Meanwhile, GPT-4.o, with the lowest value, is similar to Trump's percentage. In this regard, Obama clearly shows the highest value. For the categories *Achieve* and *Familiarity*, the differences are always significant with all of the presidents. GPT more often uses terms in the *Achieve* class and less words appearing in the *Familiarity* one. This finding confirms the presence of a complex formulation and longer words under GPT's pen. Moreover, GPT opts for a tone which underlies accomplished or fulfilled tasks.

Both GPT versions employ more terms belonging to the *Symbolism* category, and the difference with the true presidents is always significant. Moreover, the difference in percentage between GPT-3.5 and GPT-4.o is not significant. When generating political texts, GPT favors words related to abstract ideas (e.g., freedom) and national references (e.g., America). Between presidents, Obama uses these terms less often.

When inspecting the percentages of terms appearing in the *Politics* category, the two GPT versions expose significant differences in their usage. The newest model displays the highest value, more frequently referencing concrete terms related to political institutions (e.g., Congress, state, president). The differences with the presidents are always significant.

Instead of focusing on a single percentage related to a given wordlist, the LIWC system proposes a combination of several categories to generate four composite measurements, namely emotional tone, confidence (or clout), analytical thinking, and authenticity. The resulting numbers are standardised scores based on some LIWC categories, and their values range from 1 to 100 (Pennebaker *et al*., 2014; Jordan *et al*., 2019). The computed values obtained with our corpus are depicted in Table 5, which shows the largest values in bold and the smallest in italics. Moreover, a bilateral *t*-test has been applied because the values correspond to the means over all of the SOTU addresses written by each president or GPT model.

The emotional *Tone* (Monzani *et al*., 2021) combines both positive and negative dimensions (see also Table 4). Values larger than 50 indicate an overall positive tone, while numbers below this threshold

are associated with an overall negative sentiment. As shown in Table 5, both GPT versions focus exclusively on a positive timbre. The differences with the true presidential allocutions are significant. In the latter case, both positive and negative terms can be observed. In majority, however, the positive ones dominate, in part because they must convince the citizens that they have the capacity to solve current problems, and that their actions are the most appropriate for the country. Moreover, they are pleased that they have the power. Finally, between presidents, Biden displays the lowest positive emotional tone (during his term, he was confronted with the COVID-19 pandemic and the war in Ukraine).

Table 5: Composite summary measurements (LIWC).

	Tone	*Clout*	*Analytical*	*Authenticity*
GPT-3.5	**96.8**	95.5	81.1	15.31
GPT-4.o	**98.3**†	**97.3**†	79.0†	*9.7*†
Reagan	78.6†‡	85.3†‡	81.8‡	31.1†‡
Clinton	73.7†‡	89.3†‡	79.6†‡	32.6†‡
Bush	60.8†‡	89.3†‡	**84.1**†‡	22.8†‡
Obama	62.0†‡	83.7†‡	*71.7*†‡	37.1†‡
Trump	62.3†‡	89.7†‡	80.2†‡	30.0†‡
Biden	*56.2*†‡	*78.2*†‡	73.8†‡	**40.0**†‡
Presidents	66.8†‡	86.6†‡	78.9†	31.5†‡

The *Clout* (or confidence) category is used to determine the person's relative status in a social hierarchy. A leader must have a high status reflected by a higher usage of the pronouns 'we' and 'you' (see Table 3). On the contrary, a person of lower status tends to employ more I-words and impersonal pronouns (e.g., it, one) (Kacewitz *et al*., 2014; Pennebaker, 2011). People with a high social status present higher authoritative language and have a tone of higher certainty. As depicted in Table 5, both GPT versions expose a high value in this dimension. For both *Tone* and *Clout*, Biden shows the lowest value among US presidents.

The *Analytical* thinking measure has been shown to be associated with a greater academic level (Markowitz, 2023). This tone is grounded on a larger cognitive elaboration, leading to the impression of conveying more competence. An analytical language appears logical and formal, employs more articles and prepositions, and focuses more on noun phrases (Pennebaker *et al.*, 2014; Jordan *et al*., 2022). Opting for a highly analytical tone, the speaker takes the risk of appearing too distant, impersonal, and lacking an emotional aspect. On the other hand, a more intuitive and personal person writes more often with pronouns, negations, auxiliary verbs, conjunctions and some adverbs (e.g., so, very) (Pennebaker *et al*., 2014). Among presidents, Bush presents the highest analytical thinking, while Obama expresses the lowest.

The *Authenticity* measurement (Pennebaker *et al*., 2014) is related to the way a leader is able to communicate in a spontaneous way (Markowitz *et al.*, 2023), a pitch usually viewed as an honest one. Adopting this characteristic, the language is more concrete and presents more self-references in a natural way. Leaders adopting this tone appear to be closer or more connected to people's interests (Hart, 2023). However, this attitude does not imply that the speaker tells the truth (Pennebaker, 2011). As displayed in Table 5, Biden presents the highest value, while both GPT versions depict the lowest values. All presidents expose a significantly higher score than both GPT versions.

From data depicted in Table 5, GPT has a highly positive emotional tone, adopts a high-power language, and lacks authenticity. Only in analytical thinking could GPT be viewed as a true president. Biden's image appears to be clearly distinct from that of other presidents, with a more negative tone that is both low in language power and analytical thinking, but that could be viewed as honest.

6 Intertextual Distance

To evaluate more globally the similarity between all presidents and both GPT versions, an intertextual distance between all pairs of texts can be computed (Labbé, 2007). The computation of this measure between Text A and Text B is defined according to the entire vocabulary. Equation 1 specifies this measure with n_A indicating the length of Text A (in number of tokens), and $tf_{i,A}$ denoting the absolute frequency of the *i*th term (for $i = 1, 2, \ldots, m$). The value *m* represents the vocabulary length. Usually, both texts do not have the same length, so we may assume that Text B is the longest. To reduce the longest text to the size of the smallest, each of the term frequencies (in our case $tf_{i,B}$) is multiplied by the ratio of the two text lengths, as indicated in the second part of Equation 1.

$$(1) \qquad D(A,B) = \sum_{i=1}^{m} \left| tf_{i,A} - \widehat{tf_{i,B}} \right| / (2 \cdot n_A) \qquad \text{with } \widehat{tf_{i,B}} = tf_{i,B} \cdot n_A / n_B$$

Having six presidents, and for each president the two GPT versions, we have, in total, 18 texts. Directly displaying the 18 x 18 matrix containing these distances is of limited interest. Knowing that this matrix is symmetric and that the distance to itself is nil, we still have in total ((18 x 18) – 18) / 2 = 153 values. To achieve a better picture than a list of values or a dendrogram, such distance matrices can be represented by a tree-based visualisation *approximately* respecting the real distances between all nodes (Baayen, 2008; Paradis, 2011). We adopt this new representation, of which the result is displayed in Figure 1. Additionally, the string '35' has been added after each president's name to indicate speeches generated by GPT-3.5. A similar denomination has been applied for GPT-4.o.

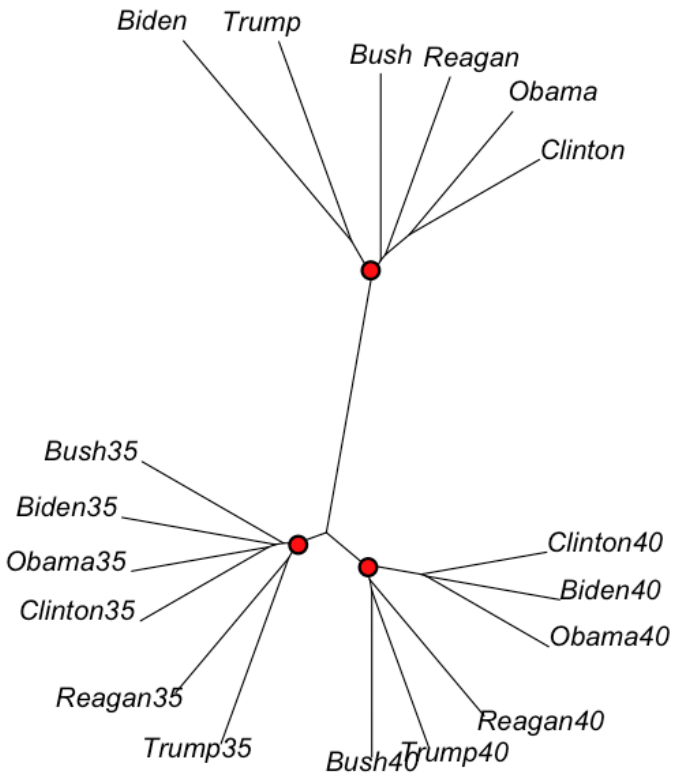


Figure 1. Overall distance between presidents and GPT versions.

Overall, this figure illustrates the large difference between the true addresses (appearing on the top part) and the other GPT speeches (depicted in the bottom part). To obtain a better understanding of this picture, the starting point of each cluster is indicated by a red dot. The two GPT versions clearly form two distinct subtrees, and the distance between them is smaller than with the set of true speeches.

With GPT-4.o, two subgroups can be defined: one with the Republican presidents (Bush 40, Trump 40, and Reagan 40), and a second with the Democrats (Clinton 40, Biden 40, and Obama 40). Moreover, the true presidents cluster displays a greater distance between each member than in the other two groups. Finally, the last two US presidents (Trump and Biden) are displayed with some distance from the four others.

7 Conclusion

Some experiments performed in this study demonstrate that both GPT models can generate political speeches sharing some similarities with real *State of the Union* (SOTU) addresses. In addition, the newest version (GPT-4.o) exposes distinct characteristics compared to GPT-3.5. For example, the messages generated by GPT-4.o are significantly longer: on average, 645 tokens vs. 493 for GPT-3.5.

The two models share some common features, such as a higher language complexity compared to true presidents. In this regard, GPT generates longer words (the mean is 4.96 letters per word), with a higher

percentage of big terms (on average, 39%), and longer sentences (20.76). Among presidents, Biden tends to present the lowest language complexity, with the shortest words and sentences.

When focusing on personal pronouns, both GPT versions opt for a large percentage of we-words (we, us, our) with few other pronouns (e.g., the third singular pronouns occur very rarely). Even if the increased frequency of we-words is a characteristic of political leaders in power, GPT employs them more often than true presidents. Between presidents, Biden presents a distinct figure with a relatively high number of I-words and second-person pronouns.

When inspecting emotional terms, both GPT models employ almost only positive terms (on average, 7.3%), leading to an optimist tone. True presidents also favour positive sentiments (on average 4.3%), along with some negative ones (2.1%). Among presidents, Bush writes with the highest number of emotional terms (on average, 4.99% are positive, 3.09% negative). This feature can be explained by the war in Iraq and against terrorists. Again, Biden uses the lowest percentage of positive terms (3.33%), and a low number of negative ones (1.74%).

When considering other categories, the two GPT versions opt for a larger percentage of *Achieve* (on average, 4.9%), *Symbolism* (5.3%), and *Politics* (4.7%) terms. This can be explained by the wish to anchor the speech in political parlance (e.g., nation, Congress, America) and to underline the results or actions already planned (e.g., win, plan). For the presidents, the average percentages are significantly lower (*Achieve:* 2.8%, *Symbolism:* 3.8%, *Politics*: 3.7%).

When considering other psychological measurements, both GPT models expose a clear language, belonging to a high-status person (*Clout*), but with a low value in authenticity. The resulting tone could appear authoritative and distant. Among presidents, Biden opts for a less optimistic and less confident tone that could also appears as being more honest.

Finally, by computing a global intertextual distance between each president and the corresponding messages generated by both GPT versions, three separate clusters are displayed: one for each GPT model, and one for the true presidents. Based on the language, the difference between machine-based speeches and real ones appears clearly, with GPT favouring a more complex language, opting for an optimistic feeling, and a more authoritative tone. Based on current technology, a LLM producing political messages can still be identified (when the text is rather long, namely more than 2,000 words). With some improvements over existing models, the risk is increasing that computers could generate speeches that can no more be discriminated from real political leaders. At that time, this technology could represent a real threat for all nations.

Acknowledgments

The author wants to thank the anonymous referees for their helpful suggestions and remarks on a previous version of this article.

References

Antoun, W., Mouilleron, V., Sagot, B., Seddah, D. (2023). Towards a robust detection of language model-generated text: Is ChatGPT that easy to detect? *CORIA-TALN Conference*, Paris June 2023, pp. 1–14.

Baayen, H.R. (2008). *Analyzing linguistic data. A practical introduction using R*. Cambridge: Cambridge University Press.

Biber, D., Conrad, S. (2009). *Register, genre, and style*. Cambridge: Cambridge University Press.

Conover, W.J. (1990). *Practical nonparametric statistics*. New York: John Wiley and Sons.

Crystal, D. (2019). *The Cambridge encyclopedia of the English language*. 3rd Ed. Cambridge: Cambridge University Press.

Daelemans, W., Kestemont, M., Manjavacas, E., Potthast, M., Rangel, F., Rosso, P., Specht, G., Stamatatos, E., Stein, B., Tschuggnall, M., Wiegmann, M, Zangerle, E. (2019). Overview of PAN 2019: Bots and gender profiling, celebrity profiling, cross-domain authorship attribution and style change detection. In: F. Crestani, M. Braschler, J. Savoy, A. Rauber, H. Müller, D. Losada, G. Heinatz Bürki, L. Cappelo, and N. Ferro (Eds.). *Experimental IR Meets Multilinguality, Multimodality*, pp. 402–416. Lecture Notes in Computer Science #11696. Cham: Springer.

Dolamic, L., Savoy, J. (2010). When stopword lists make the difference. *Journal of the American Society for Information Sciences and Technology*, 61(1), pp. 200-203.

Goodfellow, I., Bengio, Y., Courville, A. (2016). *Deep learning*. Boston: The MIT Press.

Guo, B., Zhang, X., Wang, Z., Jiang, M., Nie, J., Ding, Y., Yue, J., Wu, Y. (2023). *How close if ChatGPT to human experts? Comparison corpus, evaluation and detection*. arXiv:2301.07597.

Hart, R.P. (2020). *Trump and us. What he says and why people listen*. Cambridge: Cambridge University Press.

Hart, R.P. (2023). *American eloquence: Language and leadership in the twentieth century*. New York: Columbia University Press.

Hart, R.P., Childers, J.P., Lind, C.J. (2013). *Political tone. How leaders talk and why*. Chicago: The University of Chicago Press.

Jiao, W., Wang, W., Huang, J.T., Wang, X., Tu, Z. (2023). *Is ChatGPT a good translator? Yes with GPT-4 as the engine*. arXiv: 2301.08745. https://arxiv.org/pdf/2301.08745.pdf

Jordan, K. (2022). Language analysis in political psychology. In: Dehghani, M., Boyd, R.L. (Eds.). *Handbook of Language Analysis in Psychology*, pp. 159-172. New York: Guilford Publications.

Jordan, K.N., Sterling, J., Pennebaker, J.W., Boyd, R.L. (2019). Examining long-term trends in politics and culture through language of political leaders and cultural institutions. *Proc. National Academy of Science*, 116(9), pp. 3476–3481.

Kacewicz, E., Pennebaker, J. W., Davis, M., Jeon, M., Graesser, A. C. (2014). Pronoun use reflects standings in social hierarchies. *Journal of Language and Social Psychology*, 33(2), pp. 125–143.

Karsdorp, F., Kestemont, M., Riddell, A. (2021). *Humanities data analysis. Case studies with Python*. Princeton: Princeton University Press.

Kreuz, R. (2023). *Linguistics fingerprints. How language creates and reveals identity*. Prometheus Books, Guilford, (CT).

Labbé, D. (2007). Experiments on authorship attribution by intertextual distance in English. *Journal of Quantitative Linguistics*, 14(1), pp. 33–80.

Labbé, D., Monière, D. (2008). Je est-il un autre? *Proceedings JADT*, pp. 647-656.

Lakoff, G., Wehling, E. (2012). *The little blue book: The essential guide to thinking and talking democratic*. New York: Free Press.

Markowitz, D.M. (2023). Analytic thinking as revealed by function words: What does language really measure? *Applied Cognitive Psychology*, 37(3), pp. 1–8.

Markowitz, D.M., Kouchaki, M., Gino, F., Hancock, J.T., Boyd, R.L. (2023). Authentic first impressions relate to interpersonal, social, and entrepreneurial success. *Social Psychological and Personality Science*, 14(2), pp. 107–116.

Mizumoto, A., Yasuda, S., Tamura, Y. (2024). Identifying ChatGPT-generated texts in EFL students' writing: Through comparative analysis of linguistic fingerprints. *Applied Corpus Linguistics*, 4, in press.

Monzani, D., Vergani, L., Pizzoli, S. F. M., Marton, G., Pravettoni, G. (2021). Emotional tone, analytical thinking, and somatosensory processes of a sample of Italian tweets during the first phases of the COVID-19 pandemic: Observational study. *Journal of Medical Internet Research*, *23*(10), e29820–e29820.

Obradović, S., Power, S.A., Sheehy-Skeffington, J. (2020). Understanding the psychological appeal of populism. *Current Opinion in Psychology*, 35(10), 125–131.

Ogden, C.K. (1968). *Basic English: International Second Language*. New York: Harcourt.

Olsson, J. (2018). *More Wordcrime. Solving Crime Through Forensic Linguistics.* London: Bloomsbury.

Paradis, E. (2011). *Analysis of Phylogenetics and Evolution with R.* New York: Springer.

Pennebaker, J.W. (2011). *The Secret Life of Pronouns. What our Words Say About Us*. New York: Bloomsbury Press.

Pennebaker, J.W., Chung, C.K., Frazee, J., Lavergne, G.M., Beaver, D.I. (2014). When small words foretell academic success: The case of college admissions essays. *PLoS ONE* 9, doi.org/10.1371/journal.pone.0115844

Savoy, J., (2018). Trump's and Clinton's Style and Rhetoric During the 2016 Presidential Election. *Journal of Quantitative Linguistics*, 25(2), pp. 168-189

Savoy, J. (2020). *Machine learning methods for stylometry. Authorship attribution and author profiling*. Cham: Springer.

Savoy J., Wehren M. (2022). Trump's and Biden's styles during the 2020 US presidential election. *Digital Scholarship in the Humanities*, 37(1), pp. 229-241.

Sherrill, R. (1967). *TheaAccidental president*. New York: Grossman.

Tausczik, Y.R., and Pennebaker, J.W. (2010). The psychological meaning of words: LIWC and computerized text analysis methods. *Journal of Language and Social Psychology*, 29(1), pp. 24-54.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I. (2017). Attention is all you need. In: *Advances in Neural Information Processing Systems*, 30.

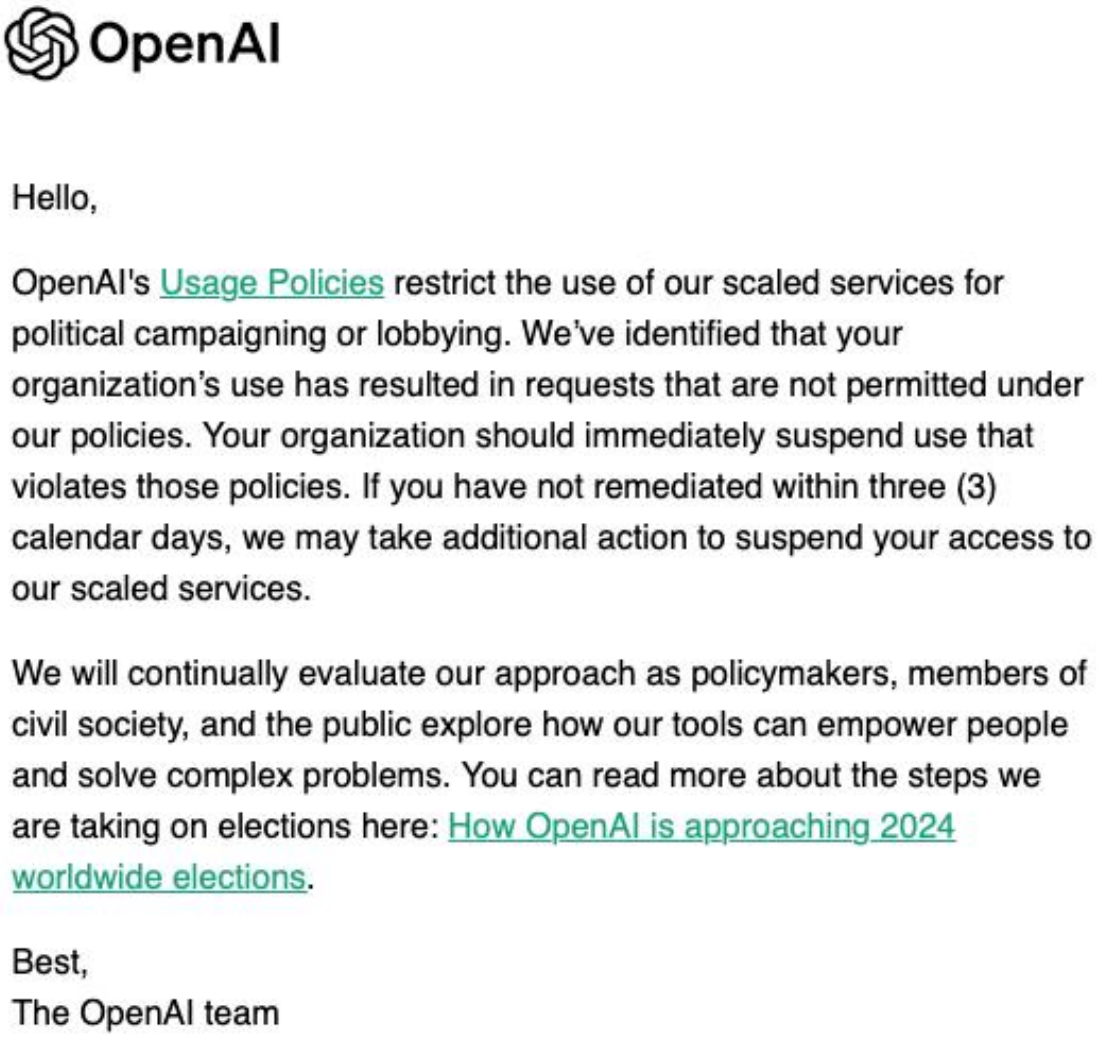
Wolfram, S. (2023). *What is GPT-4 doing… and what does it work?*. Orlando: Wolfram Research Inc., Champaign (IL).

Yule, G. (2020). *The study of language*. 7th ed., Cambridge: Cambridge University Press.

Zhao, W., Zhou, K., Li, J., Tang, T., Wang, X, Hou, Y. Min, Y., Zhang, B., Zhang, J., Dong, Z., Du, Y., Yang, C., Chen, Y., Chen, Z., Jiang, J., Ren, R., Li, Y., Tang, X., Peiyu, P., Nie, J.Y., Wen, J.R. (2023). A survey of large language models. arXiv: 2303.18223v11.

Appendix

Figure A.1. Warning received from OpenAI when generating political speeches



Comparative Statistical Analysis of Word Frequencies in Human-Written and AI-Generated Texts

Anna Kudryavtseva[1*] (0009-0004-8577-9929), Artyom Kovalevskii[1,2,3] (0000-0001-5808-3134)

[1] Novosibirsk State University, Novosibirsk, Russia
[2] Novosibirsk State Technical University, Novosibirsk, Russia
[3] Sobolev Institute of Mathematics, Novosibirsk, Russia
[*] Corresponding author's email: a.kudryavtseva@g.nsu.ru

DOI: https://doi.org/10.53482/2025_58_423

ABSTRACT

We classify texts using relative word frequencies. The task is to distinguish human-written texts from those generated by a computer using modern algorithms. We study two essay datasets, each containing an equal number of human-written and computer-generated essays. Studying Zipf diagrams shows that the generated texts have a significantly smaller vocabulary compared to human ones. However, the relative frequency of rare words (not included in the 1000 most common) does not allow us to confidently classify the texts. As additional features, we used the relative frequencies of the four most frequent words, as well as the ratio of the number of hapax legomena to the total number of different words. This feature allows to significantly improve the classification. Using these six features allows us to fairly confidently determine whether the text is computer-generated.

Keywords: Large Language Model, Zipf's Law, rare words.

1 Introduction

The advent of Large Language Models (LLMs) such as GPT-3/4 has opened entirely new possibilities in Artificial Intelligence (AI) text generation and radically changed the content of research in the field of AI.

Among the new research challenges arising from this event, one of the main ones is the problem of detecting texts created by AI, which is topical in various fields — from school and university education to information security. Intensive research in this direction is underway. In particular, such AI detection software tools as GPTZero (2023) and ZeroGPT (2024) have become widely known. Unfortunately, detecting LLM-generated texts is an intricate challenge, and until now the reliability of such software is debatable. In particular, in a study conducted by Weber-Wulff et al. (2023), researchers evaluated 14 detection tools, including GPTZero, and found that "all scored below 80% precision and only 5 above 70%."

A comprehensive review of different methods for the detection of AI-generated text is given by Wu et al. (2024). In this review, the detector techniques are divided into a few groups: watermarking techniques, statistics-based detectors, neural-based detectors, and human-assisted methods.

Recently, it was announced that a method had been developed to recognize machine-generated texts with a high degree of reliability (Hans et al., 2024). It is claimed that over a wide range of document types the method, called *Binoculars*, detects over 90% of generated samples from ChatGPT (and other LLMs) at a false positive rate of 0.01%, despite not being trained on any ChatGPT data. *Binoculars* belongs to neural-based detectors, it uses two LLMs, one is an "observer" LLM and another is a "performer" LLM.

One of possible approaches to distinguishing between human- and machine-generated texts can be based on a statistical analysis of the text vocabulary, in particular on investigating the usage of rarest and most frequent words.

Zipf (1949) and Mandelbrot (1965) showed that human texts approximately follow a power law of decreasing frequencies

$$f_r \simeq \frac{c}{(r+b)^a}, \tag{1}$$

where f_r is the relative frequency of word with rank r,

a is the Zipf exponent,

b is the Mandelbrot shift,

c is the normalizing constant.

However, Mandelbrot demonstrated that texts generated by a simple random algorithm also satisfy this law.

The distinction between human and machine texts may be found in the parameters of the law. It is known that these parameters vary over a fairly wide range, depending on the author, and are not constant for the entire language.

Piantadosi (2014) analyzed deviations of human language in the frequency distribution from the Zipf — Mandelbrot law and concluded that human language has a highly complex, reliable structure in the frequency distribution over and above this classic law.

Santis et al. (2024) studied the frequency distribution of words in novels and in texts generated by computer algorithms, but did not find a universal criterion for distinguishing them: "We have planned to go in depth on these interesting questions while maintaining the general claim that concerns the characterization of texts generated by machines with respect to some methodologies made available by the complexity sciences."

Two companion papers Abebe et al. (2022) and Abebe et al. (2023) explore the potential of the Heaps diagram (a process of counts of different words in a text) to analyze text homogeneity and find places where two different texts connect.

In addition to Zipf's law, studies highlight how temporal and structural factors shape word distributions. Altmann et al. (2009) demonstrate that word usage exhibits bursty patterns — clusters of high frequency followed by lulls — deviating from Poisson randomness and aligning with stretched exponential models. This variability is context-dependent, reflecting semantic and pragmatic influences.

Further complexity arises from the interplay of word properties and sentence structure. Popescu et al. (2009) reveals that relative word frequencies correlate with inherent linguistic features: shorter words and polysemous terms (e.g., "run") tend to occur more frequently, while morphological complexity reduces usage rates. Critically, positional dynamics in sentences also govern frequency—low-frequency words disproportionately occupy informationally salient positions, such as sentence-final slots, due to their role in conveying new or emphatic content.

Beyond these intrinsic and syntactic factors, variability across texts introduces additional stochasticity: Gerlach and Altmann (2014) demonstrate that vocabulary size exhibits Taylor's law (Taylor, 1961), where fluctuations in word diversity persist even for long texts, scaling linearly with the mean due to topic-driven heterogeneity. This quenched disorder — rooted in topical variations rather than pure randomness — renders vocabulary growth non-self-averaging, meaning lexical richness cannot be disentangled from contextual or discourse-level shifts.

These findings underscore that word frequency is not merely a function of statistical ubiquity but is mediated by syntactic roles, semantic richness, discourse structure, and systemic variability across textual domains.

Thus, while Zipf's law describes the global distribution of word frequencies, the interplay of burstiness, lexical properties, and positional constraints reveals finer-grained linguistic mechanisms that transcend frequency alone.

In the present paper, we study the statistical characteristics of AI-generated texts and compare them with those of human-written essays. For this purpose, two datasets are analyzed.

The human essays of the first dataset are taken from a project by Morgan (2012) aimed at developing an automated scoring algorithm for student-written essays. They are available from Kaggle (https://www.kaggle.com), a data science competition platform, and contain responses to a single prompt written by students.

The essays are analyzed and compared along with the essays generated by an LLM from the same prompt.

For this purpose, we used one of the most powerful freely available LLM (NousResearch, 2023). The generated essays can be found on the GitHub page https://github.com/kudrann/ai-human-data.

The second dataset is collected by Verma et al. (2023) who have been developing *Ghostbuster*, a system for detection of AI-generated texts. The dataset includes high school and university level essays taken from the IvyPanda web site (https://ivypanda.com/essays/) as well as LLM-generated essays prepared by Ghostbuster developers. They used ChatGPT to first generate a prompt corresponding to each human essay and then generate a corresponding essay that responds to that prompt. The full dataset can be found on their Github page https://github.com/vivek3141/ghostbuster-data.

2 Methodology

It is well known that in many natural languages the frequency of a word f is roughly inversely proportional to its number (rank) r in the list of the most frequent words, $f \sim 1/r$. This empirical relation is known as Zipf's law (Zipf, 1949). In fact, in many cases a generalized version of this relation known as the Zipf — Mandelbrot (ZM) law (1) works better (Mandelbrot, 1965). As an example, the frequency of words in the classic novel "Dracula" by Bram Stoker is shown in Figure 1 using the log-log scale.

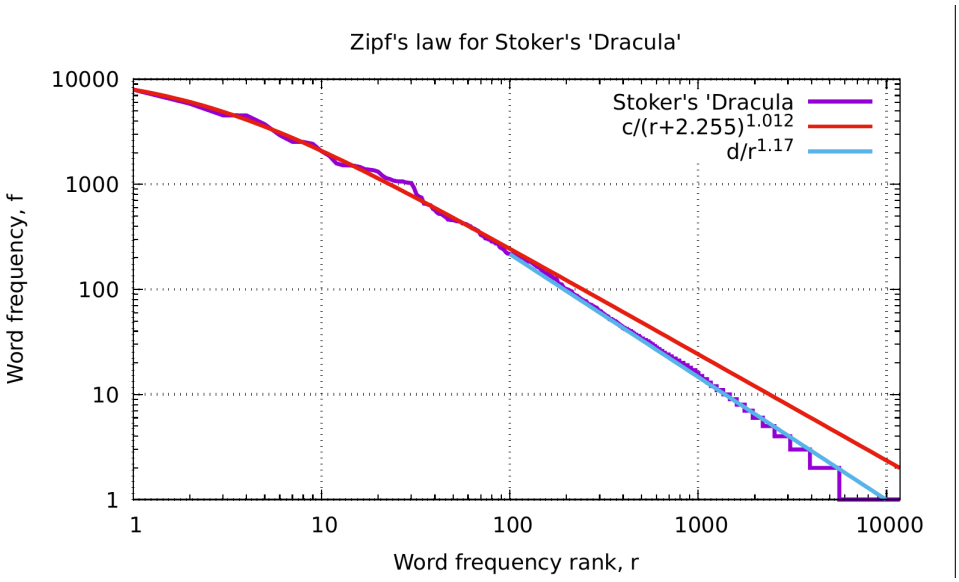


Figure 1: Word frequency distribution in the novel "Dracula" by B. Stoker.

The frequency distribution of the words is seen to be in close agreement with the ZM law at $a = 1.012$ and $b = 2.255$ for words whose rank is less than approximately 110. At the same time, the distribution of rare words deflects noticeably and cannot be described by the function with the same values of a and b. In fact, it can be better fitted by the function $\sim 1/r^a$ with $a = 1.17$ (Figure 1).

It can be expected that the distribution of rare words is specific to different authors and may be considered as an important characteristics of an author's style. In particular, one can assume that human-written and AI-generated texts can differ in statistical properties of distribution for rare words. So, we pay special attention to analyzing their usage in the essays.

For text analysis, a Python code was written using the text processing library *collections*. After preprocessing (removing punctuation and capitalization, splitting into separate words), it allows us to construct word frequency distributions, determine the parameters of the distributions, study the scatter in the frequency of rare words, and so on. The results of its work are presented below.

3 Results

3.1 The First Dataset

The essays in the dataset of Morgan (2012) were written as a response to the following prompt.

> *More and more people use computers, but not everyone agrees that this benefits society. Those who support advances in technology believe that computers have a positive effect on people. They teach hand-eye coordination, give people the ability to learn about faraway places and people, and even allow people to talk online with other people. Others have different ideas. Some experts are concerned that people are spending too much time on their computers and less time exercising, enjoying nature, and interacting with family and friends. Write a letter to your local newspaper in which you state your opinion on the effects computers have on people. Persuade the readers to agree with you.*

800 essays are selected from this database for statistical analysis. They are compared with the same number of essays generated by *Nous-Hermes-Llama2* (NousResearch, 2023), one of the most powerful freely available LLM, containing 13 billion parameters. Using the LMStudio application (LMStudio, 2024), the model quantized to 8 bits was installed on a compute cluster with 8 Nvidia GeForce GTX 1080 graphics processing units (GPUs) and 11.264 GB of video memory on each GPU. 800 essays were generated with the temperature value (a parameter that determines the degree of difference between the generated essays) $T = 0.7$. The average length of essay is 284 words, the average generation time is 21 s.

An example of the generated essay is given below:

> *Dear Editor,*
>
> *I am writing this letter to express my thoughts on the impact of computers on society.*

As technology advances and more people become reliant on computers, it is essential to consider both the benefits and drawbacks of this development.

On one hand, computers have undoubtedly made our lives easier in many ways. They provide access to a wealth of information, allowing us to learn about any topic instantly, communicate with people across the globe, and perform tasks more efficiently. In addition, they help develop important skills such as hand-eye coordination and problem-solving.

However, there are also concerns that excessive computer use can lead to negative consequences. People may spend too much time in front of screens, neglecting their physical health, social interactions, and relationships with family and friends. Moreover, the widespread use of computers has led to job losses in some sectors, causing economic hardships for many individuals.

In conclusion, while computers have revolutionized our lives in numerous ways, it is crucial that we strike a balance between embracing technology and maintaining our physical, mental, and social well-being. By being mindful of the potential drawbacks and taking steps to mitigate them, we can ensure that computers continue to benefit society positively.

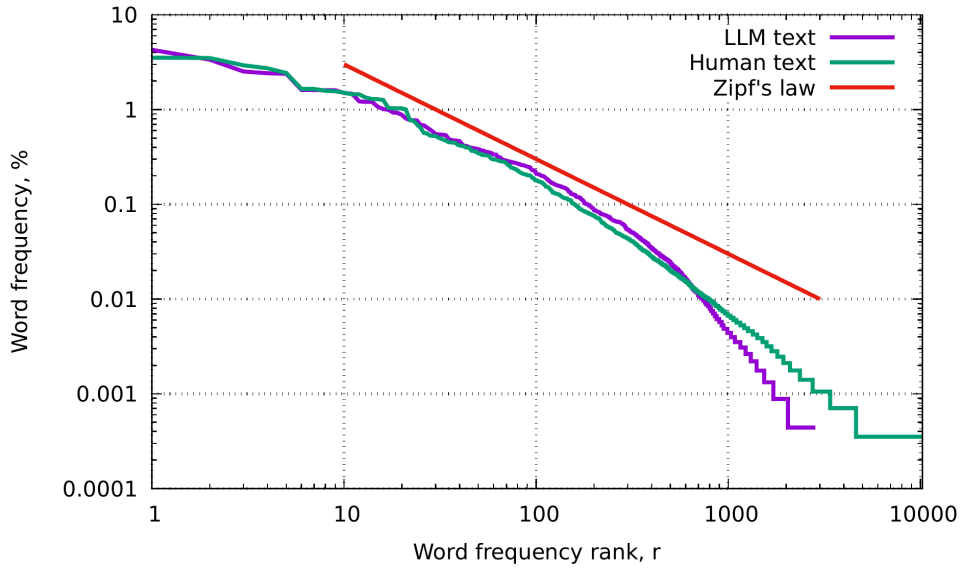


Figure 2: Word frequency distributions in the human-written and AI-generated essays. Dataset #1.

The word frequency distributions for the human-written and AI-generated essays are compared in Figure 2.

The word frequencies of both human-written and AI-generated essays deviate significantly from Zipf's law, especially if one looks at the "tails" of the distributions.

It is worth noting that there are also some distinctions in the list of the most frequent words — see Table 1.

Table 1: The most frequent words in human-written and AI-generated essays. Dataset #1.

Rank	1	2	3	4	5	6	7	8	9	10
					Humans					
Word	the	to	and	you	are	computers	on	people	of	that
Frequency	10029	9934	8308	7755	6916	4695	4682	4478	4419	4268
Percentage	3.53	3.50	2.93	2.73	2.44	1.65	1.65	1.58	1.56	1.50
					LLM					
Word	and	to	the	of	computers	have	that	in	on	with
Frequency	9707	7631	5730	5482	5418	3670	3642	3638	3632	3398
Percentage	4.27	3.36	2.52	2.41	2.38	1.62	1.60	1.60	1.60	1.50

In order to find the best fits of these distributions to the ZM law (1) we estimate the a and b constants by solving a nonlinear least-squares problem with the Levenberg—Marquardt (damped least-squares) algorithm (Gill et al., 1981, pp. 136-137). As a result, the constants are found to be $a = 1.235$, $b = 7.551$ for the human-written essays and $a = 1.035$, $b = 4.17$ for the AI-generated ones — see Figure 3 and Figure 4.

Thus, the parameters of the ZM fittings differ noticeably for two distributions. Moreover, in both cases the distributions are in reasonable agreement with the ZM law for word ranks $r \lesssim 300$, though there is some visible deviation from the ZM law in the range $50 < r < 200$ for the LLM-generated essays.

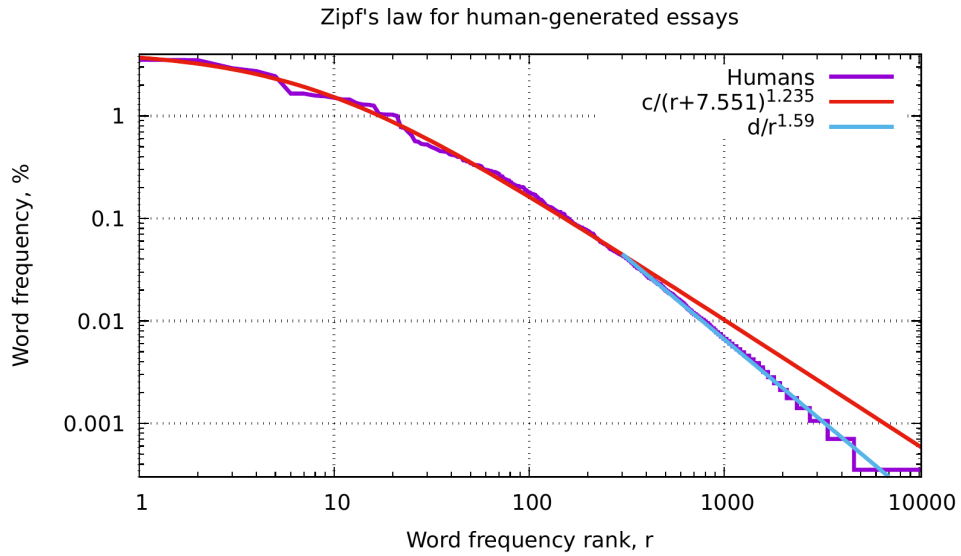


Figure 3: Word frequency distribution in the human-written essays compared with the ZM law. Dataset #1.

As concerns rare words, both distributions decay much faster than their calculated ZM fittings. The

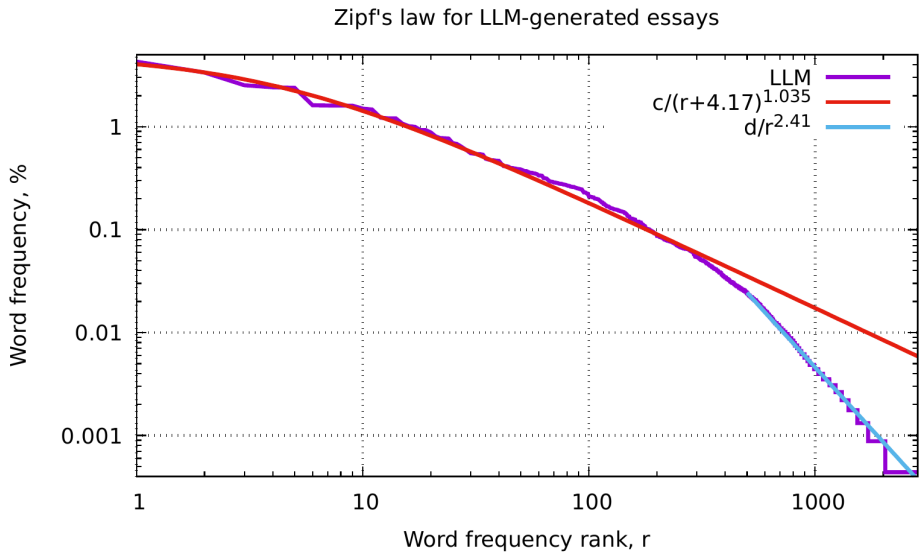


Figure 4: Word frequency distribution in the AI-generated essays compared with the ZM law. Dataset #1.

distribution tails can be better fitted by d/r^a functions with different values of a. As is seen in Figure 3 and Figure 4, the distribution for human-written essays at $r \geq 300$ are fitted well with $a = 1.59$ while the distribution for AI-generated essays at $r \geq 500$ better corresponds to $a = 2.41$. The exponents in the power law are significantly different. Also, it is seen that the transition to the power-law distributions happens for the AI-generated essays at a noticeably larger value of r than it does for the human-written ones ($r = 500$ instead of $r = 300$).

Peculiarities in the distribution tails prompted us to take a closer look at uncommon words occurring in the essays. In Figure 5 the proportion of uncommon ($r > 1000$) words is shown as a function of the essay number. It can be concluded that the average proportion of uncommon words in the human-written essays is much higher than in the AI-generated ones. Additionally, there are some essays composed by students in which the proportion is very high.

Our hypothesis is that the proportion of rare words remains stable for a homogeneous text of one author, but varies significantly between authors. The correlation coefficient between the proportion of rare words and the length of the essay in words is $corr = 0.64$. This confirms the difference between authors and also indicates that authors with a richer vocabulary write longer texts on average. The dependence of the proportion of rare words on errors and typos is analyzed in detail below.

In Table 2 the maximum proportion of uncommon words, their average proportion and the standard deviation are given for both sets of essays. One human-written essay contains 57.6% of uncommon words!

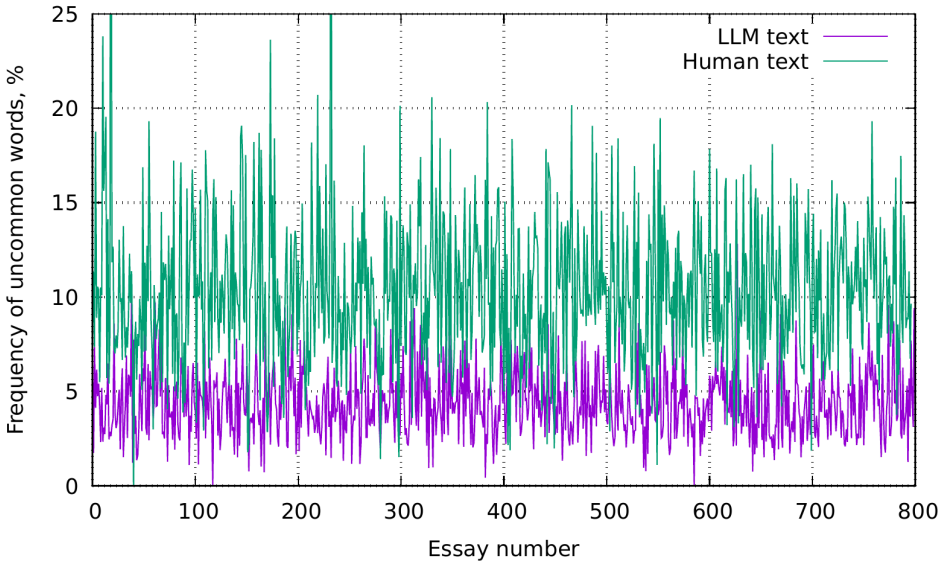


Figure 5: Frequency of uncommon words ($r > 1000$) in essays. Dataset #1.

Table 2: The proportion of uncommon words ($r > 1000$) in human-written and AI-generated essays. Dataset #1.

	Max. fraction	Mean value	Standard deviation
LLM	12.8%	4.4%	1.9%
Humans	57.6%	10.4%	4.3%

The "record-breaking" essay with 57.6 % of words not in the top 1000 looks like as follows.

> *I aegre waf the evansmant ov tnachnololage. The evansmant ov tnachnolige is being to halp fined a kohar froi alnsas. Tnanchnololage waf ont ot we wod not go to the moon. Tnachnologe evans as we maech at. The people are in tnacholege to the frchr fror the good ov live. Famas invanyor ues tnacholage leki lena orde dvanse and his fling mashine. Tnachologe is the grat.*

Spelling errors make this text virtually incomprehensible.

As can be seen from Fig. 5 and Table 2, the average proportion of uncommon words in the human-written essays is about 10%. There are some essays in which the proportion is noticeably higher, but the number of such essays does not seem high. To investigate the relationship between the number of uncommon words and that of orthographical mistakes, we analyzed 20 randomly chosen essays. There are mistakes and typos in all 20 essays. In 17 of them, their proportion does not exceed 4 %, there is also one essay each with 8, 9 and 11 % of mistakes and typos. The correlation between the number of words in an essay and the percentage of mistakes is weakly negative ($corr = -0.33$) and insignificant (p-value, statistical significance is $p = 0.16$). It is expected as poorly proficient students write shorter essays. The correlation between the percentages of mistakes and uncommon words is weakly positive ($corr = 0.30$)

and insignificant ($p = 0.20$). Thus, mistakes and typos contribute to the frequency of rare words, but their contribution is not decisive.

Thus, it can be concluded that the differences in statistical characteristics of human-written and AI-generated essays are caused, at least partially, by spelling errors inherent in humans.

3.2 Classifications of Texts of the First Dataset

We classify texts using C-Support Vector Classification (Pedregosa et al., 2011) with the parameter kernel='linear', see https://scikit-learn.org/dev/modules/generated/sklearn.svm.SVC.html.

Firstly, we use only one feature, namely x_1, which represents the fraction of uncommon words in the dataset ($r > 1000$). The model uses 75 % of the data for training and other 25 % for testing. The corresponding parts of the sets of human and AI-generated texts are selected at random.

The AI-generated texts are designated as the positive class, while the human texts are designated as the negative class. Thus, True Positive (TP) denotes the number of AI-generated texts correctly classified as AI-generated, False Positives (FP) — human-written texts incorrectly classified as AI-generated, True Negatives (TN) — human-written texts correctly classified as human-written and . False Negatives (FN) — AI-generated texts incorrectly classified as human-written.

The test is based on the pre-trained set and the test sample, which comprises 200 human texts and 200 AI-generated texts. In this test,

TP = 176, FP = 40, TN = 158, FN = 26, so that the accuracy = 0.835.

In order to obtain a more accurate classification, we use additional features of texts:

x_2 is the percentage of word "the",

x_3 is the percentage of word "and",

x_4 is the percentage of word "you",

x_5 is the percentage of word "are",

x_6 is the proportion of hapax legomena.

The features x_2, x_3, x_4, x_5 are selected on the base of Table 1 as words with the greatest differences in percentages.

The choice of feature x_6 is based on Figure 2. The last step (horizontal segment) of the relative frequency graph corresponds to hapax legomena. This step is significantly shorter in the set of AI-generated texts than in the human ones.

The Zipf parameter can be estimated by the inverse value, i.e. by dividing the number of different words by the number of hapax legomena. This estimate was proposed within the framework of the elementary probability model in Ohannessian and Dahleh (2012), its properties are studied in Chebunin and Kovalevskii (2019). In particular, the corresponding statistical test allows us to study the significance of differences in the number of hapax legomena. The correspondence of texts to the elementary probabilistic Zipf's model from the point of view of this statistics was studied in Fayzullaev and Kovalevskii (2024). Davis (2018) proposed and investigated an interesting and very precise relationship between the number of different words and the number of hapax legomena. Another interesting model for the number of hapax legomena was formulated by Milička (2009).

Using these 6 features, we have under the same approach for the same training and test sets of texts:

TP = 201, FP = 5, TN = 193, FN = 1, so we have 6 mistakes overall, and the accuracy = 0.985.

Our optimal linear classifier produces the following weights for the features (Table 3).

Table 3: Optimal linear classifier for dataset #1.

Feature	x_1	x_2	x_3	x_4	x_5	x_6
Importance	0.048	0.086	0.070	0.470	0.206	0.120

3.3 The Second Dataset

The analyzed texts consist of 1000 essays written by students and 1000 texts of approximately the same length generated by ChatGPT using prompts extracted from the students' essays. The word frequency distributions for the human-written and AI-generated essays are shown in Figure 6.

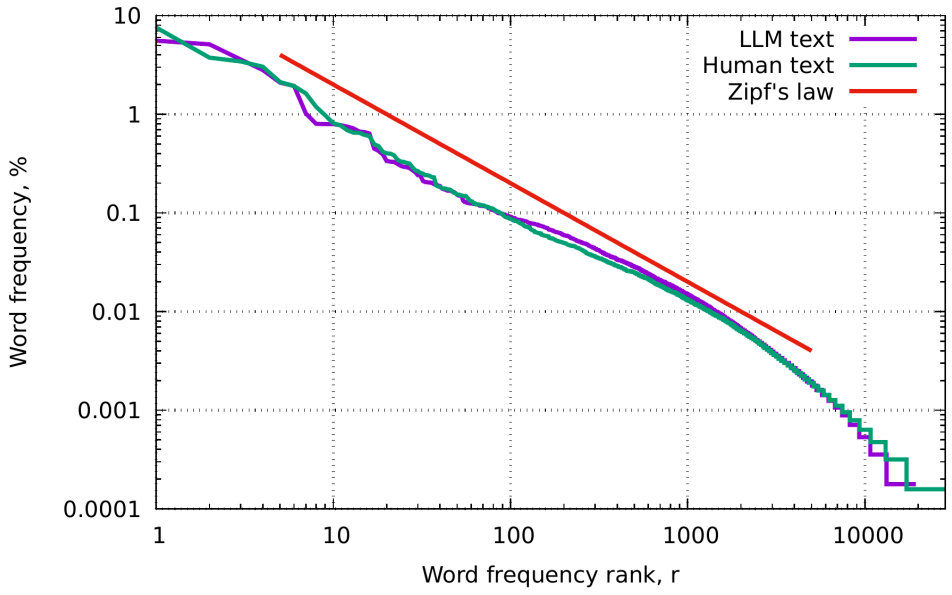


Figure 6: Word frequency distributions in the human-written and AI-generated essays. Dataset #2.

It can be seen that both distributions follow Zipf's law (not very precisely) up to $r \simeq 80 \div 100$. Their shapes for rarer words are very similar but clearly do not match the power-law distribution. It is worth noting that the distributions are much closer to each other than it was for the first dataset.

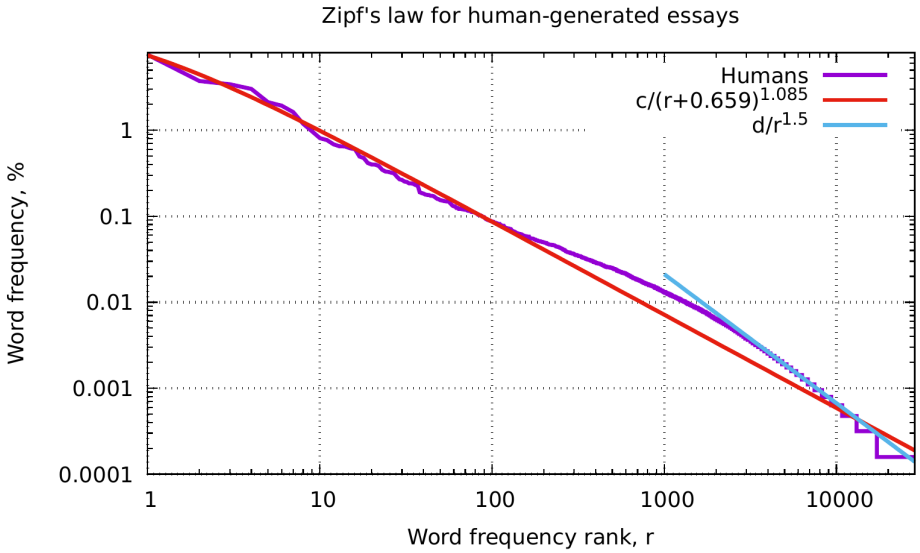


Figure 7: Word frequency distribution in the human-written essays compared with the ZM law. Dataset #2.

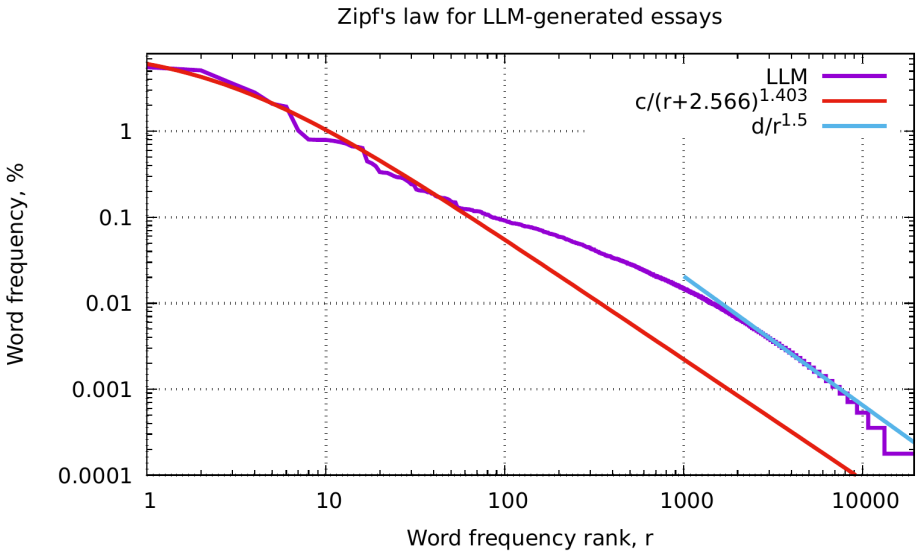


Figure 8: Word frequency distribution in the AI-generated essays compared with the ZM law. Dataset #2.

Least-square fitting of the word frequency distributions to the ZM law has been performed and the resulted best fits are compared with the distributions themselves for human-written and AI-generated essays in Figs. 7 and 8, respectively. The best fitting parameters are $a = 1.085$, $b = 0.659$ for the former curve and $a = 1.403$, $b = 2.566$ for the latter.

As can be seen, the deviations from the ZM law are particularly large for rare words, at $r > 150$ for the human-written essays and at $r > 60$ for the AI-generated ones. This is because the least-square fitting procedure primarily seeks to reduce errors at small values of r for which word frequencies are high. At the same time, the word frequency distributions for rare words follow the power law with the same exponent $d = 1.5$, much steeper than Zipf's law.

Frequencies of uncommon words (Figure 9) show that, for the second dataset, there are no such pronounced difference in their averaged and maximum fractions between human-written and AI-generated essays.

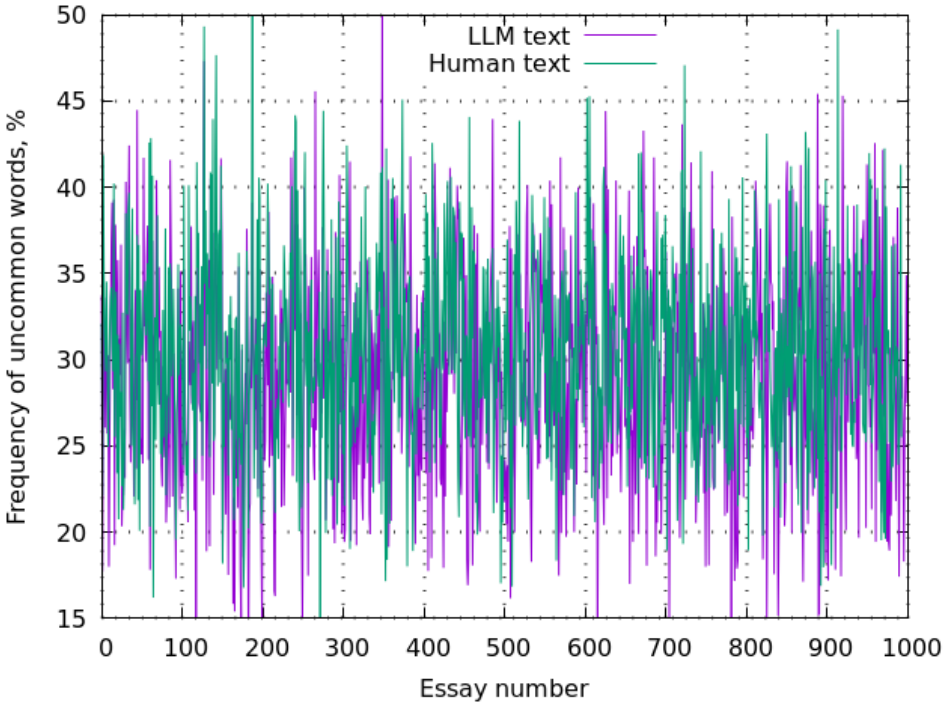


Figure 9: Frequency of uncommon words ($r > 1000$) in essays. Dataset #2.

The maximum and averaged fractions of uncommon words in human-written and AI-generated essays as well as the values of standard deviation are given in Table 4.

Table 4: The fraction of uncommon words ($r > 1000$) in human-written and AI-generated essays. Dataset #2.

	Max. fraction	Mean value	Standard deviation
LLM	50.7%	29%	6.2%
Humans	58.6%	30.4%	5.8%

The text classification based on the same one and six features as above has also been performed for the second dataset. We have for one feature:

TP = 127, FP = 119, TN = 129, FN = 124, the accuracy = 0.513, there are many mistakes.

For six features:

TP = 214, FP = 32, TN = 216, FN = 37, the accuracy = 0.862.

Overall, one can see that, in the second dataset, computer and human texts are not so easily distinguished. Using one feature leads to a large number of errors, and increasing the number of features allows us to significantly improve the accuracy.

The optimal linear classifier is shown in Table 5.

Table 5: Optimal linear classifier for dataset #2.

Feature	x_1	x_2	x_3	x_4	x_5	x_6
Importance	0.151	0.102	0.254	0.014	0.121	0.358

4 Conclusion

Computer-aided text generation is becoming increasingly common in essay writing. The present study contributes to the recognition of computer-aided text generation. This study is based on relative word frequencies and allows for the combination of the proposed methods with other methods for recognizing computer-aided text generation. A Python code has been developed for analyzing statistical features of word usage in different texts using the well-known *collections* library. Analysis of the entire text array reveals significant differences in the relative frequencies of the most common words, as well as in the total vocabulary size.

The relative frequency of rare words alone is not sufficient for confident recognition. A more accurate algorithm uses, in addition, the relative frequencies of the four most common words, as well as the ratio of hapax legomena to the total number of different words.

Dataset 1 shows a classification accuracy of 0.835 using only the relative frequency of rare words (r > 1000), and an accuracy of 0.985 using six features, with the greatest contribution to the classification coming from the personal pronoun “you”, the verb “are” and the proportion of hapax legomena. Dataset 2 shows an accuracy of 0.513 when using only the first feature, and an accuracy of 0.862 when using six features. Here, the largest weights are given to the proportion of hapax legomena, the relative frequency of the conjunction “and” and the relative frequency of rare words.

It is worth noting that, as follows from the investigation of Dataset 1, typos and orthographical mistakes common in human-written text can, to some extent, contribute to statistical features of word frequency distributions. Thus, a study comparing AI-generated texts with those written by real humans but contain no errors – whether originally or after correction – could shed even more light on the topic under study. However, it would require the use of an automatic spell-checking tool or the compilation of mistake-free

essay databases.

Acknowledgments

We are grateful to an anonymous reviewer for his remarks that helped us improve the paper. The work is supported by Program of Fundamental Scientific Research of the SB RAS, project FWNF-2022-0010.

References

Abebe, B., Chebunin, M., Kovalevskii, A. (2023). Text segmentation via processes that count the number of different words forward and backward. *Journal of Quantitative Linguistics*, *31*(1), pp. 1–18. https://doi.org/DOI: 10.1080/09296174.2023.2275342

Abebe, B., Chebunin, M., Kovalevskii, A., Zakrevskaya, N. (2022). Statistical tests for text homogeneity: Using forward and backward processes of numbers of different words. *Glottometrics*, *53*(1), pp. 42–58. https://doi.org/10. 53482/2022_53_401

Altmann, E. G., Pierrehumbert, J. B., Motter, A. E. (2009). Beyond Word Frequency: Bursts, Lulls, and Scaling in the Temporal Distributions of Words. *PLOS One*, *4*(11), e7678. https://doi.org/10.1371/journal.pone.0007678

Chebunin, M., Kovalevskii, A. (2019). Asymptotically Normal Estimators for Zipf's Law. *Sankhya A*, *81*(2), pp. 482–492. https://doi.org/10.1007/s13171-018-0135-9

Davis, V. (2018). Types, Tokens, and Hapaxes: A New Heap's Law. *Glottotheory*, *9*(2), pp. 113–129. https://doi.org/ 10.1515/glot-2018-0014

Fayzullaev, S., Kovalevskii, A. (2024). Hapax legomena via stochastic processes. *Glottometrics*, *56*, pp. 22–39. https://doi.org/10.53482/2024_56_415

Gerlach, M., Altmann, E. G. (2014). Scaling laws and fluctuations in the statistics of word frequencies. *New Journal of Physics*, *16*, 113010. https://doi.org/10.1088/1367-2630/16/11/113010

Gill, P. E., Murray, W., Wright, M. H. (1981). *Practical Optimization*. Academic Press.

GPTZero. (2023). *The Global Standard for AI Detection: Humans Deserve the Truth*. Retrieved October 10, 2024, from https://gptzero.me

Hans, A., Schwarzschild, A., Cherepanova, V., Kazemi, H., Saha, A., Goldblum, M., Geiping, J., Goldstein, T. (2024). *Spotting LLMs with Binoculars: Zero-Shot Detection of Machine-Generated Text*. Retrieved October 10, 2024, from https://arxiv.org/pdf/2401.12070

LMStudio. (2024). *LM Studio. Discover, Download, and Run Local LLMs*. Retrieved October 10, 2024, from https://lmstudio.ai

Mandelbrot, B. (1965). Information Theory and Psycholinguistics. In: Wolman, B. B., Nagel, E. (Eds.). *Scientific Psychology*, pp. 550–562. Basic Books.

Milička, J. (2009). Type-token & hapax-token relation: A combinatorial model. *Glottotheory*, *2*(1), pp. 99–110. https://doi.org/10.1515/glot-2009-0009

Morgan, J. (2012). *The Hewlett Foundation: Automated Essay Scoring*. Retrieved October 10, 2024, from https: //www.kaggle.com/competitions/asap-aes/data

NousResearch. (2023). *Model Card: Nous-Hermes-Llama2-13b*. Retrieved October 10, 2024, from https : / / huggingface.co/NousResearch/Nous-Hermes-Llama2-13b

Ohannessian, M. I., Dahleh, M. A. (2012). Rare probability estimation under regularly varying heavy tails. In: Mannor, S., Srebro, N., Williamson, R. C. (Eds.). *Proceedings of the 25th annual conference on learning theory*, pp. 21.1–21.24, Vol. 23). PMLR.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, pp. 2825–2830.

Piantadosi, S. T. (2014). Zipf's word frequency law in natural language: a critical review and future directions. *Psychonomic bulletin & review*, *25*(5), pp. 1112–1130. https://doi.org/10.3758/s13423-014-0585-6

Popescu, I.-I., Grzybek, G. A. P., Jayaram, B. D., Köhler, R., Krupa, V., Mačutek, J., Pustet, R., Uhlířová, L., Vidya, M. N. (2009). *Word Frequency Studies* (1st ed.). Mouton de Gruyter.

Santis, E. D., Martino, A., Rizzi, A. (2024). Human Versus Machine Intelligence: Assessing Natural Language Generation Models Through Complex Systems Theory. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *46*(7), pp. 4812–4829. https://doi.org/10.1109/TPAMI.2024.3358168

Taylor, L. M. (1961). Aggregation, variance and the mean. *Nature*, *189*, pp. 732–735. https://doi.org/https://doi.org/ 10.1038/189732a0

Verma, V., Fleisig, E., Tomlin, N., Klein, D. (2023). *Ghostbuster: Detecting Text Ghostwritten by Large Language Models*. Retrieved February 16, 2025, from https://arxiv.org/pdf/2305.15047

Weber-Wulff, D., Anohina-Naumeca, A., Bjelobaba, S., Foltýnek, T., Guerrero-Dib, J., Popoola, O., Šigut, P., Waddington, L. (2023). Testing of Detection Tools for AI-generated Text. *International Journal for Educational Integrity*, *19*(1), 26. https://doi.org/10.1007/s40979-023-00146-z

Wu, J., Yang, S., Zhan, R., Yuan, Y., Wong, D. F., Chao, L. S. (2024). *A Survey on LLM-Generated Text Detection: Necessity, Methods, and Future Directions*. Retrieved October 10, 2024, from https://arxiv.org/pdf/2310.14724

ZeroGPT. (2024). *Trusted GPT-4, ChatGPT and AI Detector tool by ZeroGPT*. Retrieved October 10, 2024, from https://www.zerogpt.com

Zipf, G. K. (1949). *Human Behavior and the Principle of Least Effort*. Addison-Wesley.

The distribution of syntactic dependency distances

Sonia Petrini[1] (0000-0002-0514-6223), Ramon Ferrer-i-Cancho[1*] (0000-0002-7820-923X)

[1] Quantitative, Mathematical and Computational Linguistics Research Group. Departament de Ciències de la Computació, Universitat Politècnica de Catalunya
[*] Corresponding author's email: rferrericancho@cs.upc.edu.

DOI: https://doi.org/10.53482/2025_58_424

ABSTRACT

The syntactic structure of a sentence can be represented as a graph, where vertices are words and edges indicate syntactic dependencies between them. In this setting, the distance between two linked words is defined as the difference between their positions. Here we wish to contribute to the characterization of the actual distribution of syntactic dependency distances, which has previously been argued to follow a power-law distribution. Here we propose a new model with two exponential regimes in which the probability decay is allowed to change after a break-point. This transition could mirror the transition from the processing of word chunks to higher-level structures. We find that a two-regime model – where the first regime follows either an exponential or a power-law decay – is the most likely one in all 20 languages we considered, independently of sentence length and annotation style. Moreover, the break-point exhibits low variation across languages and averages values of 4-5 words, suggesting that the amount of words that can be simultaneously processed abstracts from the specific language to a high degree. The probability decay slows down after the breakpoint, consistently with a universal chunk-and-pass mechanism. Finally, we give an account of the relation between the best estimated model and the closeness of syntactic dependencies as function of sentence length, according to a recently introduced optimality score.

Keywords: dependency syntax, dependency distance, exponential distribution, power-law distribution

1 Introduction

Language is one of the most complex and fascinating expressions of humans as social animals, stemming from our urge for communication and physical and cognitive limitations. The interaction between these two forces inevitably shapes language at many levels (Christiansen and Chater, 2016; Liu et al., 2017). Among them we here focus on syntax, namely the way in which words in a sentence compose into larger hierarchical structures, creating a parallel dimension to their plain linear arrangement. The hierarchical structure arises from the relations between words, modelled by means of a directed edge in the one-dimensional space of the network of a sentence (Figure 1). We call the resulting structure a syntactic dependency tree: each vertex is a word, and each word – besides the root – depends syntactically on its

head, to which it is connected by an edge. We define d as the absolute value of the difference between the positions of two syntactically related words (Ferrer-i-Cancho, 2004). Thus, consecutive words are at distance 1, words separated by an intermediate word are at distance 2 and so on. For instance, in Figure 1 "John" and "gave" are at distance 1, "gave" and "painting" are at distance 3, and so on.

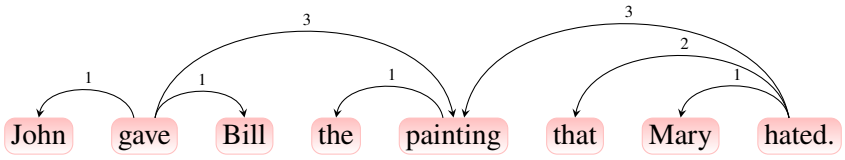


Figure 1: Example of syntactic dependency tree. Edges are labelled with the value of the syntactic dependency distance between the words they connect.

A well-established principle of Dependency Distance minimization (DDm) has been consistently found in languages, implying the preference for short dependencies (Ferrer-i-Cancho, 2004; Ferrer-i-Cancho et al., 2022; Futrell et al., 2015; Liu, 2008).

1.1 On the distribution of syntactic dependency distances

The large body of evidence in favor of DDm suggests that there are universal patterns underlying language structure, which are likely to reflect the functioning of the human brain rather than features of specific languages. Here we focus on the probability distribution of syntactic dependency distances as a window to that functioning (Liu et al., 2017). Ferrer-i-Cancho described the probability of a syntactic dependency as an exponentially decaying function of distance for sentences of fixed length in Czech and Romanian (Ferrer-i-Cancho, 2017; Ferrer-i-Cancho, 2004). However, he made an interesting observation concerning a change in the speed of the decay: the probability of observing a dependency at distance 4-5 or more is higher than expected, in the sense that the decay slows down, which apparently contradicts the DDm principle itself. Later on, Liu proposed a power-law behaviour to describe the distribution of dependency distances in a Chinese treebank, considering sentences of mixed length (Liu, 2007) that was later refined as a modified power law with an additional parameter (Liu, 2009). A later cross-linguistic study covering 30 languages identified a power-law distribution for long sentences, and an exponential trend in short ones (Lu and Liu, 2016). These approaches illustrate the complexity of the analysed problem. Nevertheless, all these distributions have a similar shape, characterized by the dominance of very short distances and a long tail (Jiang and Liu, 2015). The observed differences could hence derive from systematic discrepancies in sentence lengths, context, and annotation style, which all influence syntactic dependency distances (Ferrer-i-Cancho et al., 2022; Jiang and Liu, 2015). Moreover,

power-laws can emerge from mixing other distributions, for instance from differently parameterized exponentials (Stumpf and Porter, 2012). Hence the need – expressed in various studies (Ferrer-i-Cancho, 2004; Ferrer-i-Cancho and Liu, 2014; Jiang and Liu, 2015) – to find the common ground of these results, analyzing the distribution of dependency distances while accounting for all these factors: considering both mixed and fixed sentence lengths in a large enough parallel corpus, while also controlling for annotation style.

1.2 Exponential distributions in nature

An exponential distribution of syntactic dependency distances was predicted assuming a constraint on the average distance between syntactically related words that was justified in terms of cognitive economy (Ferrer-i-Cancho, 2004). At a lower cognitive level, the exponential distribution of projection distances between cortical areas has been justified in terms of a general principle of wiring economy in neural networks (Ercsey-Ravasz et al., 2013).

It is worth framing our proposal of a two-regime exponential distribution for syntactic dependency distances in a broader setting where a breakpoint may indicate a boundary between local and non-local dynamics. A double exponential distribution for the average distance traversed by foraging ants is a robust phenomenon where the breakpoint separates risk-averse from risk-prone trajectories (Campos et al., 2016). A hypothesis for the origins of the breakpoint in the distribution of syntactic dependency distances is elaborated below.

1.3 Short-term memory (STM) limitations

Short-term memory (also called working memory), refers to a system, or a set of processes, holding mental representations temporarily available for use in thought and action (Cowan, 2017). G. Miller's classic article set the grounds for research on a possible absolute constraint on the amount of information that can be temporarily stored in memory, and on the mechanisms enacted to cope with it (Miller, 1956). The estimated values of this maximum span vary: 7 ± 2 (Miller, 1956), $2 - 3$ (Lewis and Vasishth, 2005) or 4 ± 1 (Cowan, 2001). However, it is commonly argued that such variation reflects variation in the unit of measurement: Miller's 7 ± 2 (Miller, 1956) would correspond to the amount of information before being compressed while lower values would correspond to chunks or compressed information (Mathy and Feldman, 2012).

These considerations on STM are particularly relevant in the scope of linguistic communication: communicating requires constantly receiving and processing new inputs, without losing reference to the previous ones. To illustrate this, suppose a left-to-right incremental processing of the sentence in Figure 1. Let an open dependency be one in which only one of the two elements that compose it has already

appeared, and a closed dependency one in which both the head and the dependent have already been encountered. Then, in the context of dependency structure the success of communication depends on the ability to keep track of an open dependency while opening new ones, and without knowing *a priori* when it is going to be closed (Liu et al., 2017). Notice that dependencies represent relations between words, which are necessary for the speaker to convey a complex message building it from smaller units (encoding), and for the listener to recover such message by understanding the subjacent structure of the sequence of words (decoding). Thus, syntactic structure really reveals the way in which humans deal with physical limitations to be able to produce and process a potentially unbounded number of words. Christiansen and Chater provided an integrated framework to describe both the cognitive constraints affecting STM in language processing – what they call the "now-or-never bottleneck" – and the chunking strategy enacted to cope with them, which they refer to as "chunk-and-pass" mechanism (Christiansen and Chater, 2016). They collected a wide set of empirical results, describing the bottleneck as mainly arising from our short memory for auditory signals, the speed of new incoming linguistic input, and from memory limitations on sequence recalling tasks. According to the authors, to deal with these constraints the human cognitive system relies on a series of strategies. That is, as we receive new linguistic input, we eagerly process it by grouping units into chunks, and passing them at a more abstract level of representation; once a chunk has been integrated into the available knowledge hierarchy (Figure 1), a new one can be processed and again passed at higher representation levels. This model entails that chunking is required to store information for a longer time while a single word would be an easily forgotten piece of de-contextualized information, grouping words together produces a meaningful abstract image, which can be related to the following incoming concept. This mechanism would thus guarantee effective and efficient communication, profoundly shaping the structure of language itself.

1.4 Contribution

The primary aim of this work is to test the hypothesis that dependency distances in languages are distributed following two exponential regimes, modelled by means of a two-regime geometric distribution, and that the break-point between the regimes is similar across languages. The proposal of two regimes is motivated both empirically and theoretically. On one hand, it builds on the observations by Ferrer-i-Cancho concerning a change in probabilistic decay (Ferrer-i-Cancho, 2004). On the other hand, the existence of two different regimes would be consistent with the widely accepted idea that words are being chunked in order to be processed (Christiansen and Chater, 2016). Indeed, in a commentary on the work by Christiansen & Chater, Ferrer-i-Cancho had suggested a relation between his empirical observation and their processing framework, linking the chunking mechanism with the puzzling slowing down of probability decay in syntactic dependency distances after 4-5 words (Christiansen and Chater,

2016). Verifying this hypothesis opens the path for a deeper understanding of the distribution of syntactic dependency distances, and of how this could be influenced and shaped by universal constraints on memory. Concerning the first point, we believe our work will contribute to the existing literature on the distribution of dependency distances, finding a common ground to previous results by accounting for the effect of sentence length, context, and annotation style (Ferrer-i-Cancho, 2004; Ferrer-i-Cancho and Liu, 2014; Jiang and Liu, 2015). In fact, we consider both the syntactic structure of sentences with a specific length, and of various sentence lengths jointly, performing the analysis on a parallel corpus following two alternative syntactic dependency annotation schemes. The second point is related to one of the free parameters of our models, namely the break-point between the two regimes. If the change in probability is a mirror of the chunking mechanism enacted in language processing, the break-point we estimate could be a visible and direct statistical marker of the hypothesis advanced by Christiansen and Chater (2016). In particular, it may approximate the distance after which physical and cognitive limitations become too pressing, and the current chunk needs to be closed and encoded in memory, in order not to be overwritten by forthcoming information. Therefore, looking at the homogeneity of the estimated break-point values across languages could shed light on general cognitive patterns. Formally, we aim to verify the following two-fold hypothesis

- H_1. Syntactic dependency distances are distributed following two exponential regimes.
- H_2. The break-point between the two regimes exhibits low variation across languages and within a language.

Additionally, we further investigate the relation between the DDm principle and sentence length (Ferrer-i-Cancho and Gómez-Rodríguez, 2021), analysing how it is reflected in the shape of the distribution of syntactic dependency distances. We use Ω, a recently introduced optimality score, to quantify the intensity of DDm (Ferrer-i-Cancho et al., 2022).

1.5 Structure

The remainder of the article is organized as follows. In order to test H_1, we compare the fit of the proposed two-regime model against an ensemble of alternative distributions. Section 2 presents the definitions of the models for the distribution of syntactic dependency distances. Section 3 provides a detailed description of the data while Section 4 details the methodology. Section 5 reports the results of the model selection on sentences of languages from distinct families and investigates the relation between the best model and the optimality of syntactic dependency distances. Finally, section 6 discusses the findings, focusing on the verification of our hypotheses and on other general patterns while accounting for the observed cross-linguistic variability. Section 7 summarises the major conclusions of this article.

2 Models

We use $p(d)$ to refer to the probability that two linked words are at distance d. $d \in [1, n)$ in a sentence of n words. See Table 1 for a summary of the ensemble of models and Figure 2 for the shape of the models against an artificial random sample of their probability distributions (details on the generation of these samples are given in Appendix C). Here we present a series of well-known models (e.g., geometric distribution, right-truncated zeta distribution) and non-standard models for $p(d)$. The details of the derivation of the non-standard models are given in Appendix A.

Table 1: Models for the distribution of syntactic dependency distances. K is the number of free parameters. Refer to Appendix A for the derivation of the equations.

Model	Function	K	Definition
0	Null model	0	$\frac{1}{\binom{n}{2}}(n-d)$ if $d \in [1, n)$
0.0	Null model	1	$\frac{1}{\binom{d_{max}+1}{2}}(d_{max}+1-d)$ if $d \in [1, d_{max}]$
0.1	Extended Null model	0	$\sum_{n=min(n)}^{max(n)} \frac{n-d}{\binom{n}{2}}\, p(n)$ if $d \in [1, max(n))$
1	Geometric	1	$q(1-q)^{d-1}$ if $d \geq 1$
2	Right-truncated geometric	2	$\frac{q(1-q)^{d-1}}{1-(1-q)^{d_{max}}}$ if $d \in [1, d_{max}]$
3	Two-regime geometric	3	$\begin{cases} c_1(1-q_1)^{d-1} & \text{if } d \in [1, d_{max}] \\ c_2(1-q_2)^{d-1} & \text{if } d \geq d^* \end{cases}$
4	Two-regime - right-truncated geometric	4	$\begin{cases} c_1(1-q_1)^{d-1} & \text{if } d \in [1, d_{max}] \\ c_2(1-q_2)^{d-1} & \text{if } d \in [d^*, d_{max}] \end{cases}$
5	Right-truncated zeta distribution	2	$\frac{d^{-\gamma}}{H(d_{max},\gamma)}$ if $d \geq 1$
6	Two-regime zeta-geometric	3	$\begin{cases} c_1 d^{-\gamma} & \text{if } d \in [1, d_{max}] \\ c_2(1-q)^{d-1} & \text{if } d \geq d^* \end{cases}$
7	Two-regime - right-truncated zeta-geometric	4	$\begin{cases} c_1 d^{-\gamma} & \text{if } d \in [1, d_{max}] \\ c_2(1-q)^{d-1} & \text{if } d \in [d^*, d_{max}] \end{cases}$

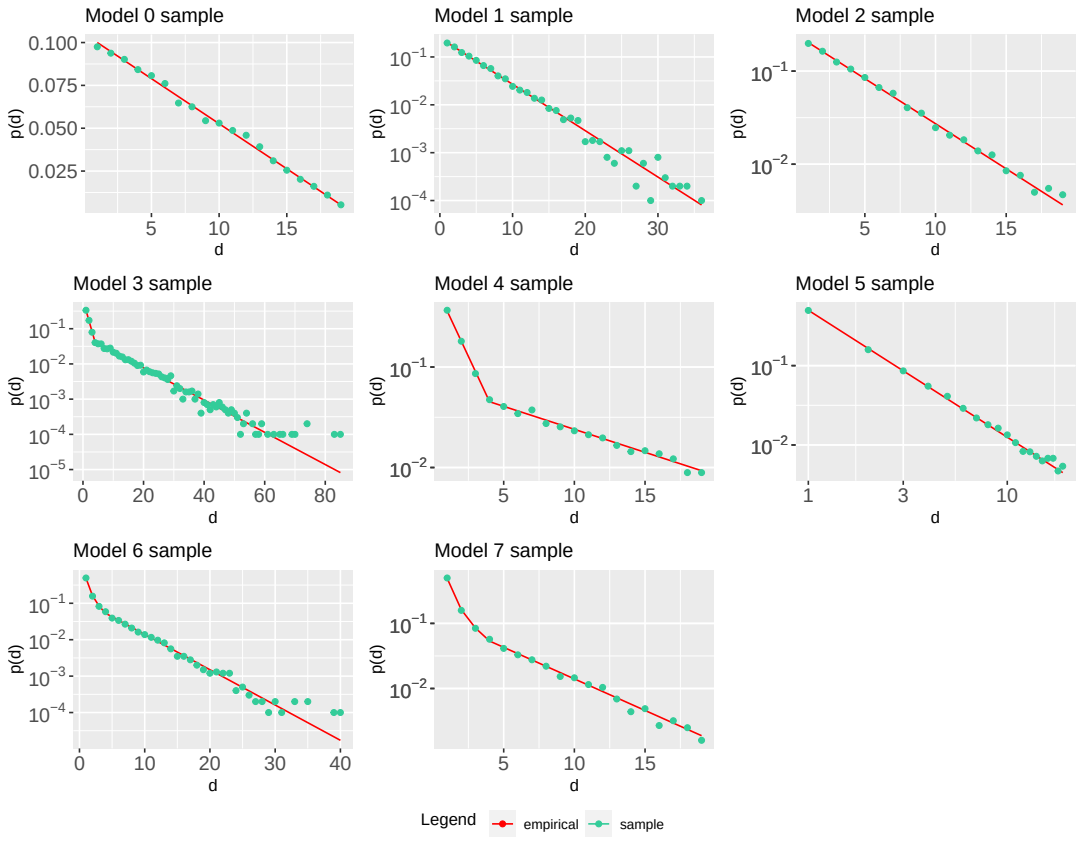


Figure 2: $p(d)$, the probability of d in a model versus a random sample of itself. The random sample has size 10^4. $n = 20$ ($d_{max} = 19$) for the right-truncated models. Thus Model 0 is the same as Model 0.0 here. $d^* = 4$ for the two-regime models. For the equations of the models refer to Table 1, while for the complete list of parameter values refer to Table 17.

The first model that we consider is Model 0, the null model obtained when a real sentence is shuffled at random or, equivalently, when there is no word order constraint (and all the $n!$ word orderings are equally likely). Then (Ferrer-i-Cancho, 2004)

$$p(d) = \begin{cases} \frac{1}{\binom{n}{2}}(n-d) & \text{if } d \in [1, n) \\ 0 & \text{otherwise.} \end{cases} \quad (1)$$

The formulation of Model 0 in 1 assumes that the maximum distance is $n - 1$ and that sentence length is unique, two assumptions that are too restrictive for our model selection setting. First, we do not know if actual maximum value of d is $n - 1$ or a lower value that is unknown to us (but could be set by some memory limitations of the human brain). Second, we are interested in the best model by fixing sentence length (where sentence length is unique) and also when considering jointly all sentences of any length for a given language (where sentence length varies). Thus, for fitting purposes, we distinguish between two specifications of Model 0. In the first one, Model 0.0, we relax the first assumption and give the model the freedom to select a maximum distance that does not need to be $n-1$, the theoretical maximum

value of d. Accordingly, Model 0.0 is defined as

$$p(d) = \begin{cases} \frac{1}{\binom{d_{max}+1}{2}}(d_{max} + 1 - d) & \text{if } d \in [1, d_{max}] \\ 0 & \text{otherwise} \end{cases}$$

where d_{max} is the only free parameter. The second specification of Model 0, Model 0.1 adapts the initial Model 0 to sentences of mixed lengths. Suppose that $p(n)$ is the proportion of sentences having length n, and $\min(n)$ and $max(n)$ are the minimum and maximum observed values of n in the sample. Then Model 0.1 is defined as

$$p(d) = \begin{cases} \sum_{n=min(n)}^{max(n)} \frac{n-d}{\binom{n}{2}} p(n) & \text{if } d \in [1, max(n)) \\ 0 & \text{otherwise.} \end{cases}$$

The following models follow the same design principle of Model 0.0 and, for the sake of simplicity, do not introduce n into the definition of the model as Model 0 or Model 0.1.

Given that distances are discrete, an exponential decay can be modeled with a geometric curve. Thus, Model 1 is the displaced geometric distribution, defined as

$$p(d) = \begin{cases} q(1-q)^{d-1} & \text{if } d \geq 1 \\ 0 & \text{otherwise,} \end{cases} \tag{2}$$

where q is the only free parameter. When $d \geq n$, the displaced geometric assumes that $p(d) > 0$ while in a real sentence $p(d) = 0$. For this reason, we also consider Model 2, that is a right-truncated version in which non-zero probability mass is restricted to $d \in [1, d_{max}]$, i.e.

$$p(d) = \begin{cases} \frac{q(1-q)^{d-1}}{1-(1-q)^{d_{max}}} & \text{if } d \in [1, d_{max}) \\ 0 & \text{otherwise,} \end{cases}$$

The two-regime models are obtained by splitting the range of variation of d into two overlapping regimes, one for $1 \leq d \leq d^*$ and another for $d \geq d^*$, where $p'(d)$ and $p''(d)$, the probability mass in the first and in the second regime respectively, satisfy $p'(d^*) = p''(d^*)$. Accordingly, Model 3 is a generalization of Model 1 that consists of two regimes, and is defined as

$$p(d) = \begin{cases} c_1(1-q_1)^{d-1} & \text{if } d \in [1, d^*] \\ c_2(1-q_2)^{d-1} & \text{if } d \geq d^* \\ 0 & \text{otherwise,} \end{cases}$$

where c_1 and c_2 are normalization factors defined as

$$c_1 = \frac{q_1 q_2}{q_2 + (1-q_1)^{d^*-1}(q_1 - q_2)} \tag{3}$$

$$c_2 = \tau c_1$$

$$\tau = \frac{(1-q_1)^{d^*-1}}{(1-q_2)^{d^*-1}}. \tag{4}$$

Thus, the only free parameters of Model 3 are q_1, q_2 and d^*.

Model 4 is a generalization of Model 3 by right truncation, that is

$$p(d) = \begin{cases} c_1(1-q_1)^{d-1} & \text{if } d \in [1, d^*] \\ c_2(1-q_2)^{d-1} & \text{if } d \in [d^*, d_{max}], \\ 0 & \text{otherwise,} \end{cases}$$

where c_1 and c_2 are normalization factors defined as

$$c_1 = \frac{q_1 q_2}{q_2 + (1-q_1)^{d^*-1}(q_1 - q_2 - q_1(1-q_2)^{d_{max}-d^*+1})}. \tag{5}$$

and $c_2 = \tau c_1$ with τ defined as in 4. The only free parameters of Model 4 are q_1, q_2, d^* and d_{max}.

Next, following previous on syntactic dependency distances (Liu, 2007), we also consider Model 5, a power-law model that is a right-truncated zeta distribution with parameters γ and d_{max} (Wimmer and Altmann, 1999), that is defined as follows

$$p(d) = \begin{cases} \frac{d^{-\gamma}}{H(d_{max},\gamma)} & \text{if } d \geq 1 \\ 0 & \text{otherwise,} \end{cases}$$

where

$$H(d_{max}, \gamma) = \sum_{k=1}^{d_{max}} \frac{1}{k^\gamma}$$

is the generalized harmonic number of order γ of d_{max}. Finally, we introduce Models 6 and 7, that are also composed of two regimes, the first one distributed as a right-truncated power-law and the second one as a geometric curve. Model 6 is defined as

$$p(d) = \begin{cases} c_1 d^{-\gamma} & \text{if } d \in [1, d^*] \\ c_2(1-q)^{d-1} & \text{if } d \geq d^* \\ 0 & \text{otherwise,} \end{cases}$$

where c_1 and c_2 are normalization factors defined as

$$c_1 = \frac{q}{qH(d^*,\gamma) + d^{*-\gamma}(1-q)} \tag{6}$$

$$c_2 = \tau c_1$$

$$\tau = \frac{d^{*-\gamma}}{(1-q)^{d^*-1}}. \tag{7}$$

Model 7, the right-truncated version of Model 6, is defined as

$$p(d) = \begin{cases} c_1 d^{-\gamma} & \text{if } d \in [1, d^*] \\ c_2(1-q)^{d-1} & \text{if } d \in [d^*, d_{max}] \\ 0 & \text{otherwise,} \end{cases}$$

where

$$c_1 = \frac{q}{qH(d^*,\gamma) + d^{*-\gamma}(1-q-(1-q)^{d_{max}-d^*+1})}, \tag{8}$$

and $c_2 = \tau c_1$ with τ defined as in 7. The only free parameters of Model 6 are γ, d^* and q. Model 7 adds a third free parameter that is d_{max}.

2.1 Speed of decay

When plotted in log-linear scale, an exponential curve becomes a line. For a geometric model (2), the slope of that line is $\log(1-q)$ since

$$\begin{aligned} \log p(d) &= \log q(1-q)^{d-1} \\ &= d\log(1-q) + \log\frac{q}{1-q}. \end{aligned}$$

That slope conveys information about the speed of probability decay. Such slope is a decreasing function of q (Figure 3), meaning that as q increases the slope becomes more negative, and probability decays faster. In light of this fact, we consider parameters q (Models 1 and 2) as well as q_1 and q_2 (Models 3-4) to account for the speed of exponential decay in the two regimes of Models 3-4, and we refer to them as "slope parameters" for simplicity.

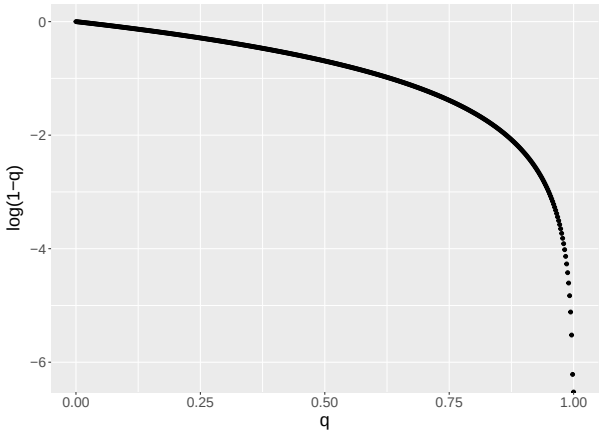


Figure 3: Slope of a geometric curve in log-linear scale as a function of its parameter q for $q \in [0, 1)$.

3 Material

Table 2: The languages, their linguistic family and their writing system.

Language	Family	Writing system
Arabic	Afro-Asiatic	Arabic
Chinese	Sino-Tibetan	Han
Czech	Indo-European	Latin
English	Indo-European	Latin
Finnish	Uralic	Latin
French	Indo-European	Latin
German	Indo-European	Latin
Hindi	Indo-European	Devanagari
Icelandic	Indo-European	Latin
Indonesian	Austronesian	Latin
Italian	Indo-European	Latin
Japanese	Japonic	Japanese
Korean	Koreanic	Hangul
Polish	Indo-European	Latin
Portuguese	Indo-European	Latin
Russian	Indo-European	Cyrillic
Spanish	Indo-European	Latin
Swedish	Indo-European	Latin
Thai	Kra-Dai	Thai
Turkish	Turkic	Latin

We extract syntactic dependency distances from a parallel subset of 20 languages from the Universal Dependencies collection (Nivre et al., 2017). See Table 2 for the languages, their linguistic family and their writing system. This subset is parallel in the sense that it contains the same sentences translated into every language. We use version 2.6, available here. Parallelism is crucial for robust cross-linguistic comparisons, as context can largely influence various aspects of language, including dependency structure. Another factor that shall be considered is annotation style, as there is no univocal way to generate syntactic dependency trees starting from a sentence. For this reason, we compare two different annotation styles: Universal Dependencies (Nivre et al., 2017) and the alternative Surface Syntactic Universal Dependencies (Gerdes et al., 2018). We refer to the two resulting versions of the collection as PUD and PSUD. See Table 3 and Table 4 for a summary of the main statistical features of PUD and PSUD respectively. It can be seen that mean dependency distance values ($\mathrm{mean}(d)$) are smaller in PSUD.

Table 3: Summary of PUD collection. $\#s$ stands for number of sentences, $\#d$ stands for number of distances.

Language	$\#s$	$\#d$	$\min(d)$	$\text{mean}(d)$	$\max(d)$	$\min(n)$	$\text{mean}(n)$	$\max(n)$
Arabic	995	17514	1	2.30	30	3	18.60	50
Czech	995	14976	1	2.39	29	3	16.05	44
German	995	17544	1	3.11	42	4	18.63	50
English	995	17711	1	2.53	31	4	18.80	56
Finnish	995	12465	1	2.24	21	3	13.53	39
French	995	21165	1	2.52	36	4	22.27	54
Hindi	995	20517	1	3.30	42	4	21.62	58
Indonesian	995	16311	1	2.26	27	3	17.39	47
Icelandic	995	15860	1	2.32	34	3	16.94	52
Italian	995	20413	1	2.48	35	3	21.52	60
Japanese	995	24703	1	2.97	65	4	25.83	70
Korean	995	13978	1	2.75	37	3	15.05	43
Polish	995	14720	1	2.23	27	3	15.79	39
Portuguese	995	19808	1	2.53	34	4	20.91	58
Russian	995	15369	1	2.27	32	3	16.45	47
Spanish	995	19986	1	2.50	32	3	21.09	58
Swedish	995	16119	1	2.47	31	4	17.20	49
Thai	995	21034	1	2.44	38	4	22.14	63
Turkish	995	13727	1	2.91	34	3	14.80	37
Chinese	995	17501	1	3.09	39	3	18.59	49

Table 4: Summary of PSUD collection. $\#s$ stands for number of sentences, $\#d$ stands for number of distances.

Language	$\#s$	$\#d$	$\min(d)$	$\text{mean}(d)$	$\max(d)$	$\min(n)$	$\text{mean}(n)$	$\max(n)$
Arabic	995	17514	1	2.05	30	3	18.60	50
Czech	995	14976	1	2.11	29	3	16.05	44
German	995	17544	1	2.82	38	4	18.63	50
English	995	17711	1	2.12	31	4	18.80	56
Finnish	995	12465	1	2.04	22	3	13.53	39
French	995	21165	1	2.13	35	4	22.27	54
Hindi	995	20517	1	3.04	38	4	21.62	58
Indonesian	995	16311	1	2.00	27	3	17.39	47
Icelandic	995	15860	1	1.92	34	3	16.94	52
Italian	995	20413	1	2.10	35	3	21.52	60
Japanese	995	24703	1	2.73	67	4	25.83	70
Korean	995	13978	1	2.70	38	3	15.05	43
Polish	995	14720	1	2.00	27	3	15.79	39
Portuguese	995	19808	1	2.13	34	4	20.91	58
Russian	995	15369	1	2.05	32	3	16.45	47
Spanish	995	19986	1	2.13	31	3	21.09	58
Swedish	995	16119	1	2.07	31	4	17.20	49
Thai	995	21034	1	2.20	39	4	22.14	63
Turkish	995	13727	1	2.86	33	3	14.80	37
Chinese	995	17501	1	2.99	39	3	18.59	49

4 Methodology

The code for this work was written both in `R` and `python`, and is available here.

4.1 Model selection

We here describe the model selection procedure implemented to test H_1. This methodology is validated with the help of artificially generated random samples from a given distribution (Appendix C).

Optimal parameters for each model are estimated by maximum likelihood. Then, the best model is selected according to Information Criteria (Anderson and Burnham, 2004). In real languages (this section), models are compared through Akaike Information Criterion (AIC). In artificially generated random samples (Appendix C), the best model is better selected through Bayes Information Criterion (BIC) because the true data generating process is known. BIC differs from AIC by relying on the assumption that the real distribution is among the tested ones (Wagenmakers and Farrell, 2004). For a given model, we use the following definitions of these scores (Anderson and Burnham, 2004)

$$\begin{aligned} AIC &= -2\mathcal{L} + 2K\frac{K}{N-K-1} \\ BIC &= -2\mathcal{L} + K\log N, \end{aligned} \tag{9}$$

where K is the number of parameters of the model and N is the sample size. With respect to AIC, the criterion proposed by Schwarz (BIC) applies a stronger penalty for the number of parameters.

Given that both AIC and BIC are measures of information loss, the best model for a sample is the one minimizing the selected score. We aim to find the best model for a sample of N distances $\{d_1, d_2, ..., d_i, ..., d_N\}$, where $\min(d)$ and $\max(d)$ are, respectively, the minimum and maximum observed distances, and $f(d)$ is the frequency of distance d in the sample. Then the sample size is

$$N = \sum_{i=1}^{max(d)} f(d_i) = \sum_{d=1}^{max(d)} f(d).$$

The log-likelihood functions of the models are summarized in Table 13. See Appendix B for a derivation of the log-likelihood functions for each model.

4.1.1 Parameter estimation

Maximum likelihood estimation (MLE) algorithms require one to specify the range of variation of the parameters as well as proper initial values. It is well-known that MLE methods are highly sensitive to the choice of the starting values, as they may incur local optima when minimizing the minus log-likelihood function (Myung, 2003). Here we explain the criteria used to select the initial value and the range of variation of the parameters, which are summarised in Table 5 and Table 6 respectively. Let x_{init} be the initial value of parameter x. Also, let $\max_i(d)$ be the $i-th$ largest value of d in the sample, so that $\max_1(d) = max(d)$. Similarly, let $\min_i(d)$ be the $i-th$ smallest value of d in the sample, so that $\min_1(d) = min(d)$.

Table 5: The initial values of the parameters for maximum likelihood estimation. Here Model 0 refers to model 0.0.

Model	d_{max}	q	q_1	q_2	d^*	γ
0	$\max(d)$	-	-	-	-	-
1	-	q_{init}	-	-	-	-
2	$\max(d)$	q_{init}	-	-	-	-
3	-	-	q_{1init}	q_{2init}	5	-
4	$\max(d)$	-	q_{1init}	q_{2init}	5	-
5	$\max(d)$	-	-	-	-	γ_{init}
6	-	q_{init}	-	-	5	γ_{init}
7	$\max(d)$	q_{init}	-	-	5	γ_{init}

The rationale for the choices in Table 5 and Table 6 is as follows

- d_{max}. The maximum observed distance is both the starting point and smallest admissible value, while there is no upper bound.

- q. In the geometric models (Models 1 and 2), the initial value for q, q_{1init}, is the maximum likelihood estimator, i.e. the inverse of the mean observed distance $q_{init} = 1/mean(d)$. The bounds are set so that $q \in (0, 1)$ to avoid values out of the domain of the log-likelihood function. In Models 6 and 7, the initial value of q for the second regime is set to the maximum likelihood estimator $1/mean(d)$ of an ideal geometric distribution, but restricting the mean to distances greater than d^*.

- q_1 and q_2. These two parameters are both initialized by first running a linear regression on $\log p(d)$ and d, for $d \leq d^*$ in the case of q_{1init}, and for $d \geq d^*$ in the case of q_{2init}. Then, the respective slopes β_1 and β_2 are used to compute the initial values via $q_{1init} = 1 - e^{\beta_1}$ and $q_{2init} = 1 - e^{\beta_2}$. Notice that, as the tail of the distribution is noisy, the estimated slope sometimes results in a 0 or even a positive value for values of d^* very close to $\max(d)$. When that happened, the corresponding q_{2init} was set to its lower bound. As in q, the bounds are set so that $q_1, q_2 \in (0, 1)$.

- d^*. The initial value is 5, as suggested by the visual inspection of the plots. The parameter is bounded to vary between $\min_2(d)$ and $\max_2(d)$, based on the minimum requirement on the size of the two regimes (section 4.1.3). Indeed, by setting d^* to either $\min_1(d)$ or to $\max_1(d)$, one of the two regimes would only be composed by one isolated observation, from which no trend can be inferred. Incidentally, the DDm principle, predicts that $\min_2(d) = 2$ if n is large enough (Ferrer-i-Cancho, 2004).

- γ. For Model 5, the initial value of the MLE estimator of the exponent of a continuous power-law (Newman, 2005):

$$\gamma_{init} = 1 + N \left[\sum_{i=1}^{N} \frac{d_i}{min(d)} \right]^{-1} .$$

For Models 6 and 7 (where only the first regime follows a zeta distribution), γ_{init} is computed over the distances up to d^*.

Table 6: The lower (low) and upper (up) bounds of the parameters for maximum likelihood estimation. $\epsilon = 10^{-3}$. Here Model 0 refers to Model 0.0.

	d_{max}		q		q_1		q_2		d^*		γ	
Model	**low**	**up**	**low**	**up**	**low**	**up**	**low**	**up**	**low**	**up**	**low**	**up**
0	$\max(d)$	∞	-	-	-	-	-	-	-	-	-	-
1	-	-	ϵ	$1-\epsilon$	-	-	-	-	-	-	-	-
2	$\max(d)$	∞	ϵ	$1-\epsilon$	-	-	-	-	-	-	-	-
3	-	-	-	-	ϵ	$1-\epsilon$	ϵ	$1-\epsilon$	$\min_2(d)$	$\max_2(d)$	-	-
4	$\max(d)$	∞	-	-	ϵ	$1-\epsilon$	ϵ	$1-\epsilon$	$\min_2(d)$	$\max_2(d)$	-	-
5	$\max(d)$	∞	-	-	-	-	-	-	-	-	0	∞
6	-	-	ϵ	$1-\epsilon$	-	-	-	-	$\min_2(d)$	$\max_2(d)$	0	∞
7	$\max(d)$	∞	ϵ	$1-\epsilon$	-	-	-	-	$\min_2(d)$	$\max_2(d)$	0	∞

4.1.2 Maximum likelihood estimation (MLE)

We considered two MLE methods in `R`: `mle()` from `stats2` and `mle2()` from the `bbmle` package (Bolker, 2007). The base `R` implementation, `mle()`, may explore values out of the specified bounds thus resulting in errors. Where this is the case, we resort to the enhanced, more robust version of the optimizer, `mle2()`, which is able to return a result even if the algorithm does not reach convergence. Both `mle2a()` and `mle2()` optimize on a continuous space. Hence, for the discrete parameters, i.e. d^* and d_{max}, we retrieved their most likely value by exhaustively exploring all values included between their theoretical bounds. In this way, we also decrease the complexity of MLE by reducing the number of parameters to be optimized through the call to `mle()` or `mle2()`. Thus, for each value of d^* (and d_{max} in the right-truncated models) we optimized the remaining parameters, and finally selected the parameters combination resulting in the highest log-likelihood.

4.1.3 Requirements for two-regime models

In order to fit a double-regime model to a data sample, we need $N \geq 3$. In fact, at least two points are needed in order to infer a speed of probability decay within a regime, meaning that each regime has to contain at least 2 distinct observations. Given that the value assigned to the break-point is common to the two regimes, this results in a requirement of $N \geq 3$. See Figure 4 for an example of this scenario, displaying the distribution of syntactic dependency distances for sentences of 4 words in Italian, annotated according to SUD. Notice that this requirement directly implies that sentences with $n < 4$ are excluded from the model selection procedure.

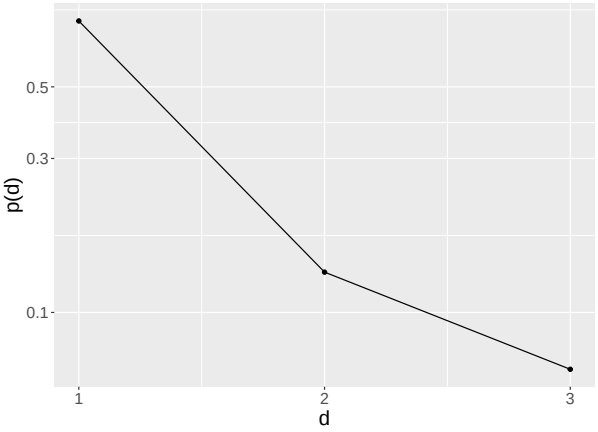


Figure 4: Syntactic dependency distance distribution of sentences with 4 words in Italian, annotated according to SUD. Only three unique values of d have been observed.

4.1.4 Representativeness

When performing model selection on sentences of specific lengths from a certain language, we obtain a set of best models, one for each sentence length. To summarize this information and obtain a single best model for each language, we consider the most frequent best model within that language. However, this raises the concern of whether all best models are equally reliable, as some of them are estimated on a single sentence. For instance, very long sentences, which are normally rare, are thus likely to be underrepresented in the data. On the other hand, setting a single specific threshold on the minimum number of sentences required for a sentence length to be included in the voting procedure would mistakenly hide important aspects of the analysis. In fact, the suitable threshold should depend on sentence length itself. Consider a very long sentence, composed of 50 words, and a very short one, of only 4 words. While – keeping fixed the syntactic structure – the first one could appear with $50!$ different re-orderings, the second one could only be written in $4!$ possible ways. Thus, a single sentence observed for $n = 4$ is much more representative (as the expected variability in dependency distance is lower) for the whole length category than a single one observed for $n = 50$. For this reason, we report the most frequent best models both when no threshold is set (Table 8) and for increasing representativeness threshold (Figure 7).

4.2 The Ω optimality score

Ω is a recently introduced optimality score for the closeness of syntactic dependency distances, which integrated normalization with respect to both a minimum and a random baseline (Ferrer-i-Cancho et al., 2022). The score is defined as

$$\Omega = \frac{D_{rla} - D}{D_{rla} - D_{min}},$$

where D is the observed sum of dependency distances in a sentence, D_{rla} is the expected sum of dependency distances in a uniformly random linear arrangement of its words (Ferrer-i-Cancho, 2004, 2019), i.e.

$$D_{rla} = \frac{n^2 - 1}{3} \tag{10}$$

and D_{min} is the sum of dependency distances in a minimum linear arrangement of the words (Esteban and Ferrer-i-Cancho, 2017; Shiloach, 1979). Both baselines assume that the network structure is fixed. D and D_{min} are computed using the `python` interface of the Linear Arrangement Library (Alemany-Puig et al., 2021).

Positive values of Ω indicate that syntactic dependency distances in the sentence are shorter than one would expect from picking uniformly at random among all the possible $n!$ orderings. The maximum, $\Omega = 1$, is reached when $D = D_{min}$. Conversely, a negative value indicates that distances are being maximized, as they are higher than expected in a random shuffling of words in a sentence. When word order is random, Ω will take values tending to 0. $\langle\Omega\rangle$ is the average value of Ω over individual sentences.

5 Results

We fit the models introduced in the Section 2 to a parallel collection of texts from 20 languages called PUD, that has been annotated with syntactic dependencies as in Figure 1. To control for annotation style we consider two variants, PUD with the original annotation style (Nivre et al., 2017) and PSUD, that follows the alternative SUD annotation style (Gerdes et al., 2018). Refer to section 3 for further details on the data, and to section 4.1 for a complete description of the model selection procedure.

This section is organized as follows. First, we report on the best models (Section 5.1), the break-points of the two-regime models (Section 5.2) and the relationship between slope parameters (q_1 and q_2) for each language (Section 5.3), both by considering fixed and mixed sentence lengths. We define representativeness threshold, shortly representativeness, as the minimum number of distinct sentences with a certain length for such length to be included in model selection (a further justification of this threshold is found in Section 4.1.4. Section 5.1 investigates the robustness of conclusions with respect to sample representativeness. Detailed tables of the estimated parameters, Akaike Information Criterion (AIC) scores, and AIC differences for both collections can be found in Appendix D. Second, we will investigate the relationship between the best model and the degree of optimality of syntactic dependency distances on sentences of fixed length (Section 5.4. Notice that we often refer jointly to Models 3 and 4 (6 and 7) as 3-4 (6-7), given that they model the same probability distribution with or without a right-truncation point.

5.1 Model selection

The best model to describe syntactic dependency distances independent of sentence length is composed of two regimes in every language and collection (Table 7). Models 3-4 dominate over Models 6-7, with 13/20 languages in PUD and 11/20 in PSUD having Model 3 or 4 as the best one. We find overall agreement between the two annotation styles, both in terms of best model and in terms of right-truncation. The exceptions to this agreement are Indonesian and Japanese – for which PUD yields an exponential decay in the first regime, while PSUD identifies a power-law one – and Chinese, English, and Italian, where the best model in PUD and PSUD differs by right truncation. In Figure 5, we show how the best models in PUD are able to accurately capture the bulk of the distribution, with some variability left in the tail. The equivalent figure for PSUD can be found in Appendix D.

Table 7: Best model for the distribution of syntactic dependency distances in sentences of mixed lengths for every language and collection. Models 3-4 are marked with pink and Models 6-7 with blue to ease visualization.

Language	PUD	PSUD	Language	PUD	PSUD
Arabic	7	7	Italian	4	3
Chinese	6	7	Japanese	4	7
Czech	3	3	Korean	7	7
English	3	4	Polish	3	3
Finnish	6	6	Portuguese	3	3
French	4	4	Russian	3	3
German	3	3	Spanish	4	4
Hindi	7	7	Swedish	3	3
Icelandic	3	3	Thai	6	6
Indonesian	3	7	Turkish	7	7

Table 8: Most voted best model for the distribution of syntactic dependency distances in sentences of fixed lengths, for every language and collection. The most voted best model is computed aggregating models by type, thus counting together the occurrences in which Models 3-4 (Models 6-7) are the best. Models 3-4 are marked with pink, Model 5 with yellow, and Models 6-7 with blue to ease visualization.

Language	PUD	PSUD	Language	PUD	PSUD
Arabic	5	5	Italian	3-4	3-4
Chinese	5	5	Japanese	3-4	6-7
Czech	3-4	3-4	Korean	6-7	6-7
English	3-4	3-4	Polish	3-4	5
Finnish	6-7	6-7	Portuguese	3-4	3-4
French	3-4	3-4	Russian	3-4	5
German	3-4	3-4	Spanish	3-4	3-4
Hindi	6-7	6-7	Swedish	3-4	3-4
Icelandic	3-4	3-4	Thai	5	5
Indonesian	3-4	5	Turkish	6-7	6-7

The best model for sentences of fixed lengths shows some variability for short and long sentences (Figure 6). Nevertheless, a double regime model is the most frequent best one across sentence lengths

in 17/20 languages in PUD (including a tie between Model 5 and Models 6-7 in Chinese), and in 14/20 languages in PSUD (Table 8). Within the languages for which a two-regime model is the best one, Models 3-4 win in 13/17 languages in PUD, and in 9/14 in PSUD. Once again, we find high consistency between annotation styles, with the exceptions of Indonesian, Polish, and Russian, for which PSUD yields Model 5 as the most frequent best one (while PUD yields models 3-4), and Japanese, for which PUD and PSUD differ in the type of two-regime model. Finally, Model 5 is the most frequent best one in both collections for Arabic, Chinese, and Thai. However, Figure 7 shows how the most voted best model ceases to be Model 5 in some instances of both PUD and PSUD when the representativeness of a sentence length is taken into account. The only languages in which Model 5 is consistently the most frequent best one even after imposing an arbitrary high threshold are Thai, Indonesian, and Arabic in PSUD. Arabic shows a border-line behaviour in PUD, with Model 5 being consistently the most voted only up to a certain threshold value. Finally, a comparison of the actual distribution against the best model in sentences of fixed characteristic length is shown in Appendix E.

5.2 The break-point

When looking at languages globally, meaning considering jointly sentences of any length, we find that the break-point d^* always takes small values – ranging between 2 and 7 – and has a quite small standard deviation (Figure 8 and Table 9), meaning that its value is similar across languages. This is especially true for Models 3-4 and the PSUD collection: out of 11 languages having Models 3-4 as the best ones in this collection, 9 have an estimated break-point at $d^* = 4$ (Figure 8). In PUD these models have an average d^* value of 5, but with some more variability across languages. In both types of two regime models median and mean values are virtually the same, independently of annotation style, providing additional evidence for the low variance of d^* (Table 9). Checking the distribution of d^* within a language allows us to verify whether global values (found when mixing sentence lengths), namely the bars in Figure 8, are good approximations of the break-points actually observed in real sentences of any fixed length. We display the distribution of d^* across sentence lengths for each language in the same figure as a violin plot. Once again the median is very close to the mean in almost every combination of two-regime model and annotation style – with the exception of Models 6-7 in PSUD – further supporting H_2 (Table 10). Then, notice that where Models 3-4 are the best we observe relatively narrow distributions, skewed towards low values and showing one or a few modes (Figure 8). In particular, the global value of d^* is virtually

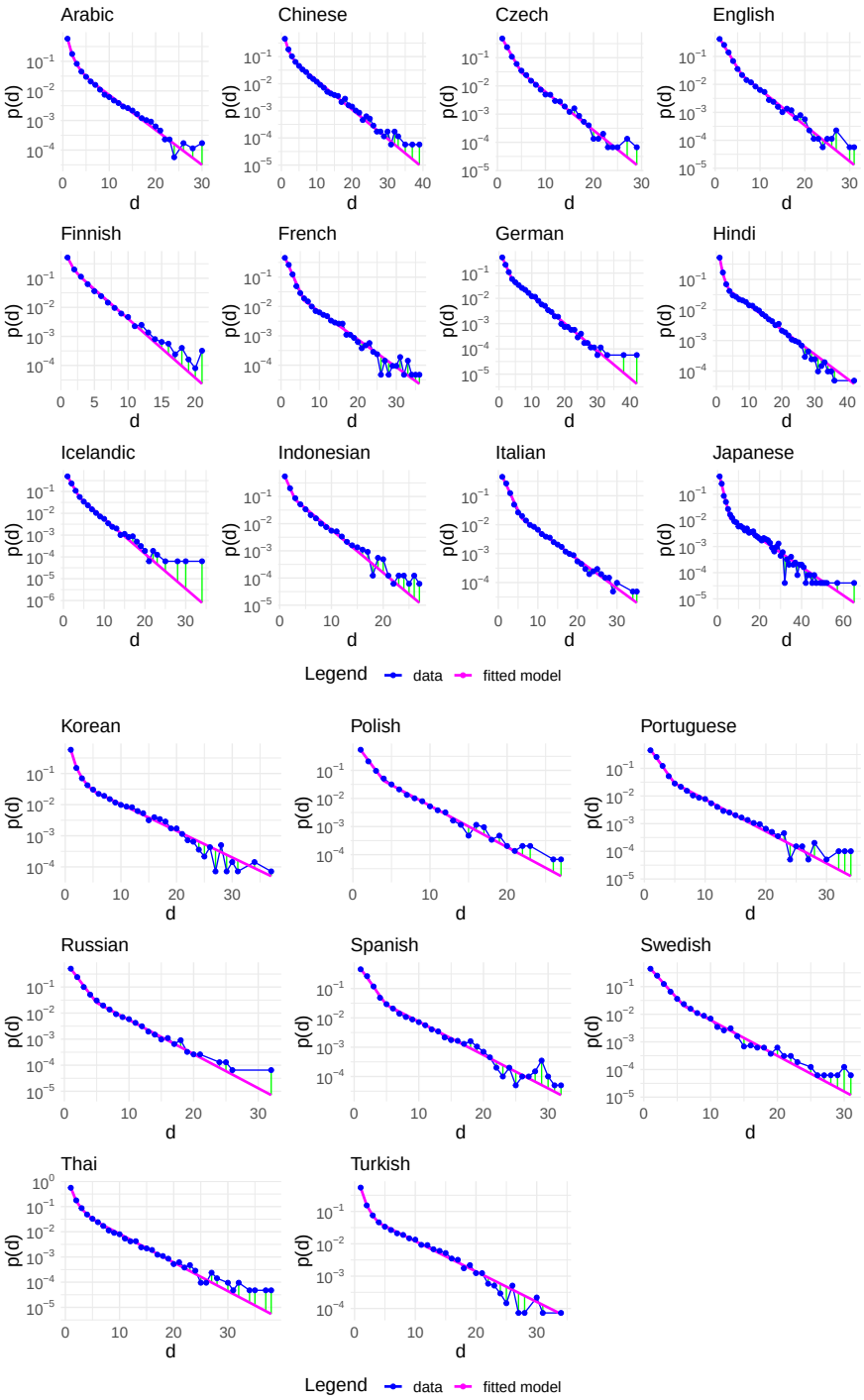
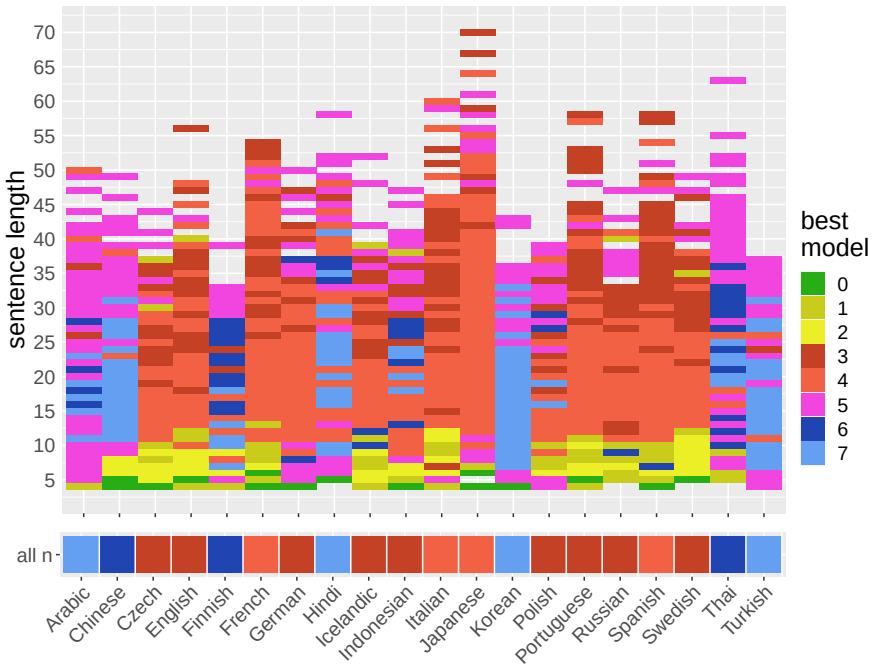
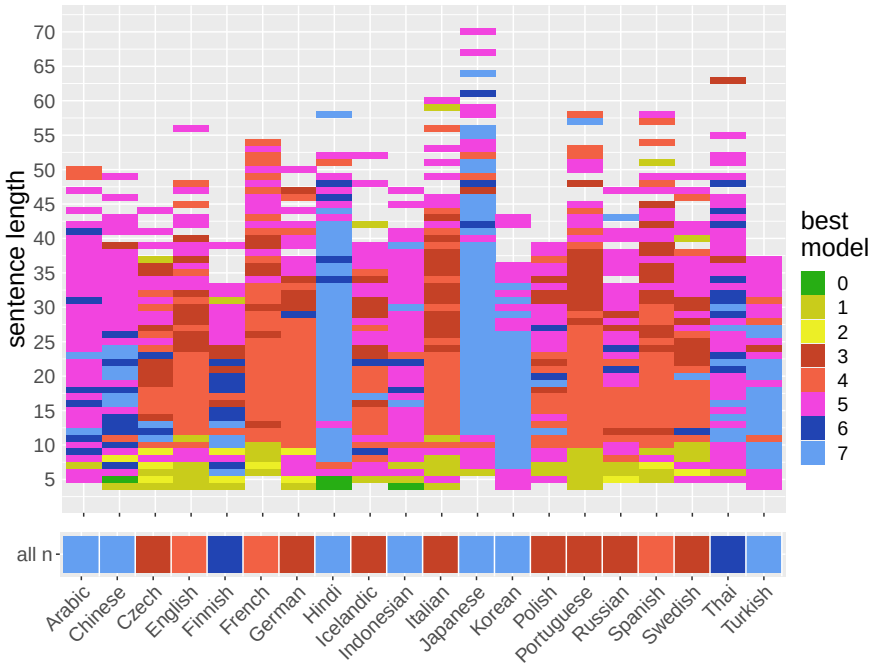


Figure 5: $p(d)$, the probability that a dependency link is formed between words at distance d according to the data and the best model for every language in PUD.



(a) PUD collection



(b) PSUD collection

Figure 6: Distribution of best model for each sentence length on top, with reference to the best model on mixed sentence lengths at the bottom. (a) PUD collection. (b) PSUD collection. In both (a) and (b) the empty tiles mark lengths for which no sentence was observed, or on which model selection was not performed given the minimum requirement to fit a double-regime model, described in section 4.1.3. Here Model 0 refers to Model 0.0.

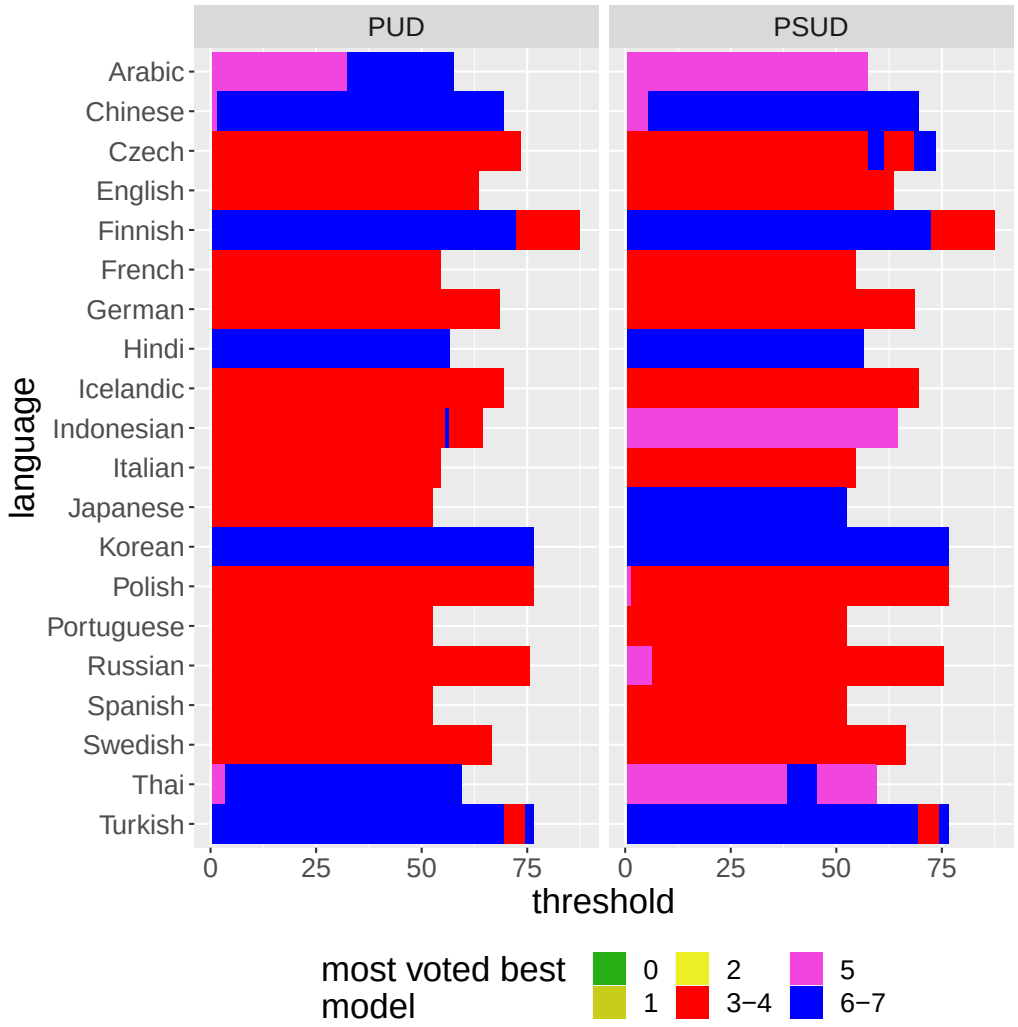


Figure 7: Most voted best model type across sentence lengths for increasing representativeness threshold. When no threshold is set (1 minimum sentence), we get the scenario displayed in Figure 6. Ties are counted in favour of models without two regimes. Here Model 0 refers to Model 0.0.

always found in correspondence of one of these modal values, confirming its representativeness for the whole language. Considering that sentences can reach up to a minimum of 37 (Turkish) and a maximum of 70 words (Japanese) (Table 3) the observed variation ranges in Models 3-4 are quite small, with values going up to roughly $d^* = 13$. On the other hand, within languages for which Models 6-7 are the best when mixing sentence lengths, the distribution of d^* across different sentence lengths is generally flatter, especially in PSUD. Even where values are centered around a mode, this does not correspond with the break-point estimated globally, with the exception of Hindi. Thus, it appears like the global break-points estimated in Models 3-4 are good approximations of the values observed within the language, while estimates of d^* in Models 6-7 are less reliable as representations of the actual break-point if there is any.

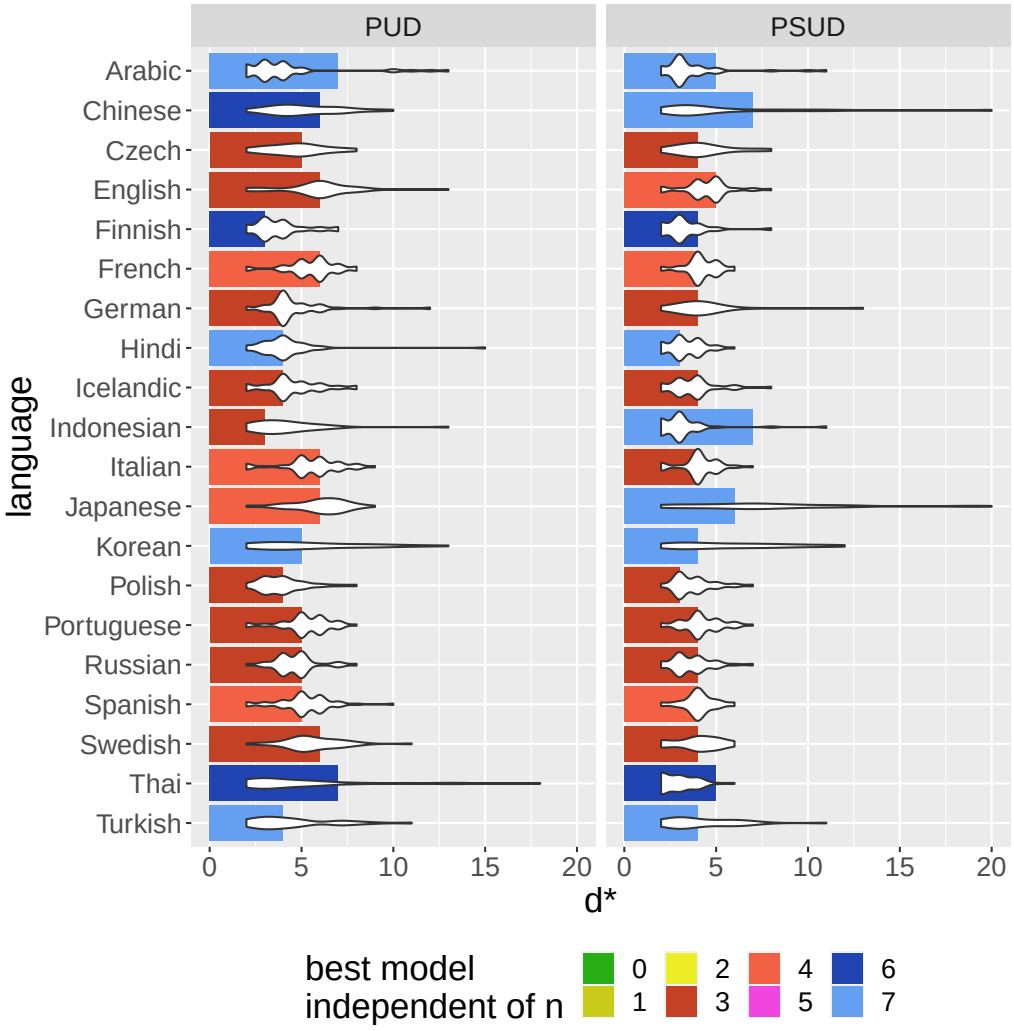


Figure 8: Value of d^* for mixed sentence lengths (bars) in each language and collection, and its distribution across fixed sentence lengths (violin plots), color-coded by best model independent of sentence length (namely the best model estimated on sentences of mixed lengths). Model 0 refers to Model 0.1 in the context of mixed sentence lengths.

Table 9: Summary statistics of the d^* parameter, by annotation style and type of two-regime model, estimated from model selection on sentences of mixed lengths. The summary is computed over languages where Models 3-4 are the best, where Models 6-7 are the best, and over all languages where a double-regime model is the best (Models 3-4-6-7). Thus, sample size is measured in number of languages. *s* stands for sample size, *sd* stands for standard deviation.

	Models	*s*	**Min.**	**1st Qu.**	**Median**	**Mean**	**3rd Qu.**	**Max.**	*sd*
PUD	3-4	13	3.00	4.00	5.00	5.00	6.00	6.00	1.00
	6-7	7	3.00	4.00	5.00	5.14	6.50	7.00	1.57
	3-4-6-7	20	3.00	4.00	5.00	5.05	6.00	7.00	1.19
PSUD	3-4	11	3.00	4.00	4.00	4.00	4.00	5.00	0.45
	6-7	9	3.00	4.00	5.00	5.00	6.00	7.00	1.41
	3-4-6-7	20	3.00	4.00	4.00	4.45	5.00	7.00	1.10

Table 10: Summary statistics of d^* parameter, by collection and type of two-regime model, estimated from model selection on sentences of fixed lengths. The summary is computed over sentence lengths and languages where Models 3-4 are the best, where Models 6-7 are the best, and over all languages and sentence lengths where a double-regime model is the best (Models 3-4-6-7). Thus, sample size is measured in number of distinct sentence lengths. *s* stands for sample size, *sd* stands for standard deviation.

	Models	*s*	**Min.**	**1st Qu.**	**Median**	**Mean**	**3rd Qu.**	**Max.**	*sd*
PUD	3-4	431	2.00	4.00	5.00	5.37	6.00	13.00	1.43
	6-7	134	2.00	4.00	6.00	6.28	7.00	18.00	3.03
	3-4-6-7	565	2.00	4.00	5.00	5.59	6.00	18.00	1.97
PSUD	3-4	297	3.00	4.00	4.00	4.32	5.00	13.00	1.11
	6-7	190	2.00	3.00	5.00	6.21	8.00	20.00	3.73
	3-4-6-7	487	2.00	4.00	4.00	5.06	5.00	20.00	2.65

5.3 Speed of decay

Recall that q_1 and q_2 are the slope parameters of Models 3-4, which quantify the speed of probability decay. For each language in which a two-regime model is the best, we consider q_1, q_2, and their ratio q_1/q_2, where the latter quantity is computed to establish which slope is steeper. It has been suggested that the probability decay is slower in the 2nd regime (Ferrer-i-Cancho, 2017; Ferrer-i-Cancho, 2004). When Models 6-7 are the best models, we estimate q_1 of the first regime by fitting the corresponding double exponential model (Model 3 or 4). When we refer to a slope, we refer to its absolute value.

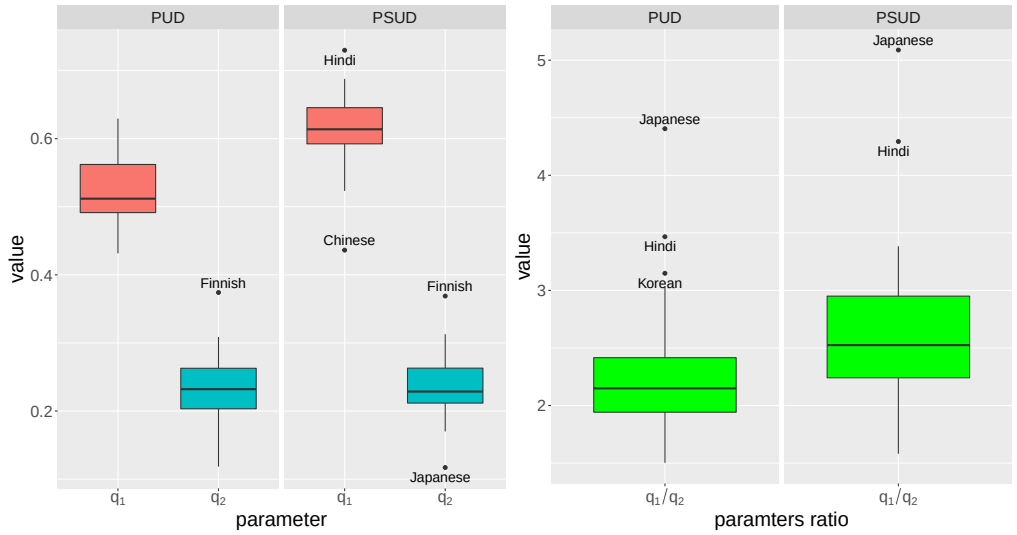
Where the best model has two regimes, the estimated slope parameters for each regime are fairly similar across languages (Figure 9 and Table 11). In addition, notice that the ratio q_1/q_2 is larger than 1 for every language and annotation style, and that q_1 and q_2 have a quite small standard deviation (Table 11). Standard deviation values are practically the same for the two parameters, but q_2 takes much lower values, meaning that it is relatively more variable than q_1. Moreover – as in the case of the break-point parameter – median and mean values are virtually the same, for both q_1 and q_2 and independently of annotation style. The slope estimated in the first regime in PUD is significantly lower than the one estimated in PSUD (Figure 9 (a)). Moreover, the estimated slopes show a clear pattern, with probability in the first regime consistently decaying faster compared to the second one. This pattern holds for the overwhelming majority of sentence lengths within a language, with a few exceptions found for very short sentences (Figure 10).

Table 11: Summary statistics of q_1 and q_2 parameters and their ratio (q_1/q_2) for model selection on sentences of mixed lengths, by annotation style (referred to as collection). Statistics are computed over all sentence lengths and languages for which a double-regime model is the best. *sd* stands for standard deviation.

	Collection	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	*sd*
q_1	PUD	0.43	0.49	0.51	0.52	0.56	0.63	0.05
	PSUD	0.44	0.59	0.61	0.61	0.65	0.73	0.06
q_2	PUD	0.12	0.20	0.23	0.24	0.26	0.37	0.06
	PSUD	0.12	0.21	0.23	0.24	0.26	0.37	0.05
q_1/q_2	PUD	1.50	1.94	2.15	2.32	2.42	4.40	0.70
	PSUD	1.58	2.24	2.52	2.75	2.95	5.09	0.79

5.4 The best model versus the optimality of syntactic dependency distances

Ω is a new closeness score for syntactic dependency distances. The higher its value, the closer the syntactically related words. Refer to section 4.2 for further details on its properties and computation.



(a) q_1 and q_2 parameters distribution (b) Ratio $\frac{q_1}{q_2}$ distribution

Figure 9: Distribution of slope parameters q_1 and q_2 and their ratio. Isolated points are labelled with the corresponding language.

The score takes positive values when syntactic dependency distances are minimized, negative values when they go against minimization, and values around 0 when there is no pressure in either direction (Ferrer-i-Cancho et al., 2022). Let $\langle\Omega\rangle$ be the average value of Ω over all sentences with a given length in a language. See Figure 11 and Figure 12 for the best model for each sentence length (a) and the corresponding value of $\langle\Omega\rangle$ (b), for PUD and PSUD respectively. First, in sentences of a very few words, the best model is either Model 0 or one with a single regime, and the values of the optimality score signal the coexistence of the three possible systems: anti-DDm (orange tiles), no bias (white tiles), and pro-DDm (purple tiles). Given the definition of the score, we expect that, under the assumption that Model 0 is the real distribution, $\langle\Omega\rangle$ will take values around 0, as both situations underlie random word ordering. This expectation is met in 6/8 instances, as displayed in Table 12, and as suggested by the correspondence between white tiles in (b) and green tiles in (a). The two exceptions are Korean in PSUD and Polish in PUD, for which the best model is Model 5. Then, for sentences longer than 5-6 words, $\langle\Omega\rangle$ indicates that distances in syntactic structures are always minimized, which is mirrored in the disappearance of Model 0 and the predominance of the single regime models. Finally, as pressure for minimization further increases with sentence length, these simpler models are progressively replaced by the models with two regimes.

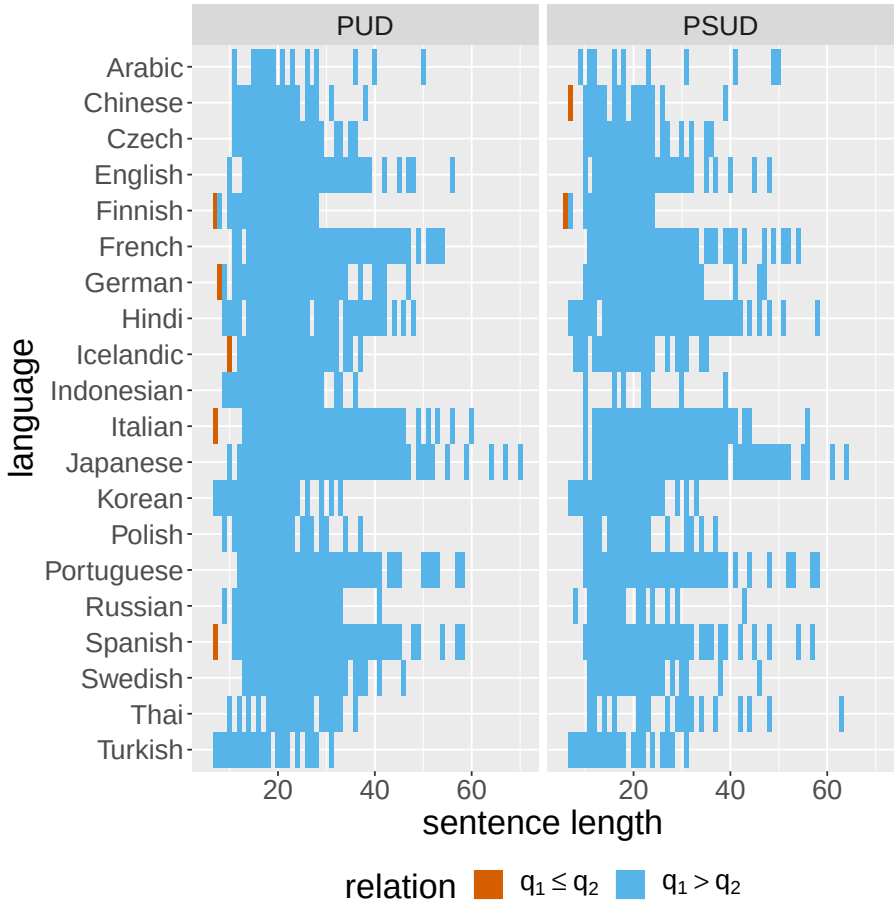
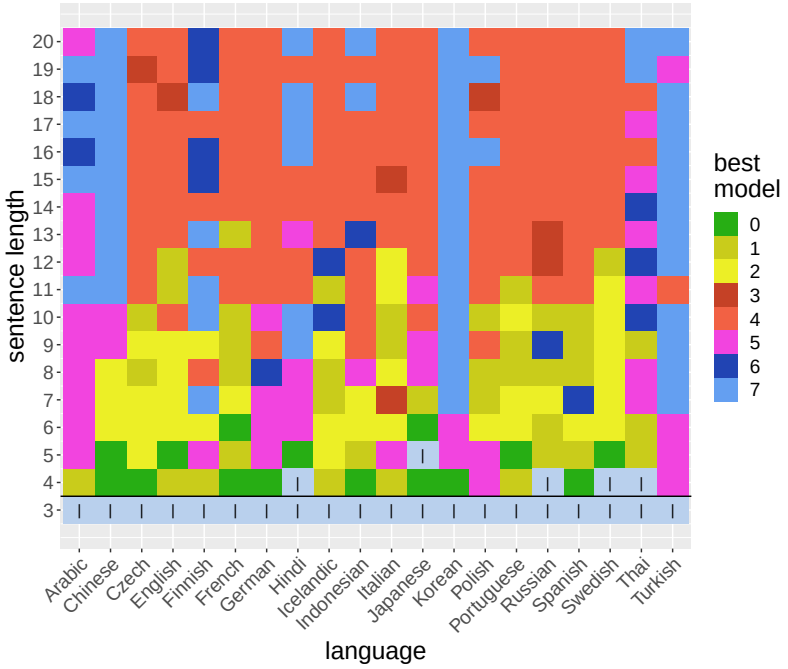


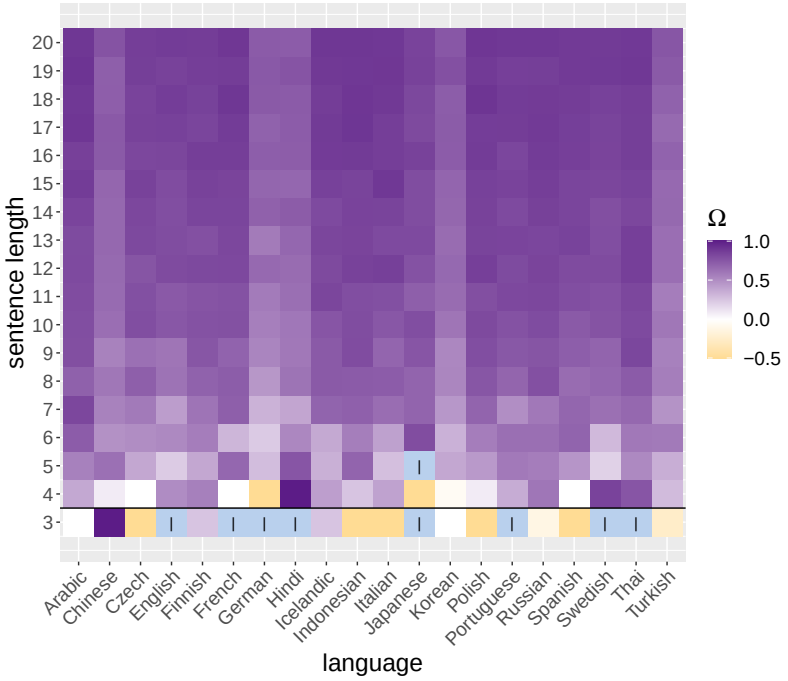
Figure 10: Relation between slope parameters q_1 and q_2 estimated from model selection on fixed sentence lengths. Lengths for which $q_1 \leq q_2$ are colored in red, while those for which $q_1 > q_2$ are colored in blue. Where the best model was 6 (7), the first slope was approximated by fitting Model 3 (4) with the original value of d^*. The empty tiles indicate lengths for which no sentence was observed, a two-regime model was not the best one, or on which model selection was not performed given the minimum requirement on the number of observed distance values to fit a double-regime model, described in section 4.1.3.

Table 12: Estimated best model on fixed sentence in collections, languages, and sentence lengths for which $|\langle\Omega\rangle - \epsilon| \leq 0$, with $\epsilon = 0.1$. $\langle\Omega\rangle$ is the average value of Ω over all sentences with a given length in a language.

Collection	Language	n	$\langle\Omega\rangle$	Best model
PUD	Korean	4	−0.05	0
	Czech	4	0.00	0
	French	4	0.00	0
	Spanish	4	0.00	0
	Polish	4	0.08	5
	Chinese	4	0.08	0
PSUD	Korean	4	−0.10	5
	Hindi	4	0.00	0

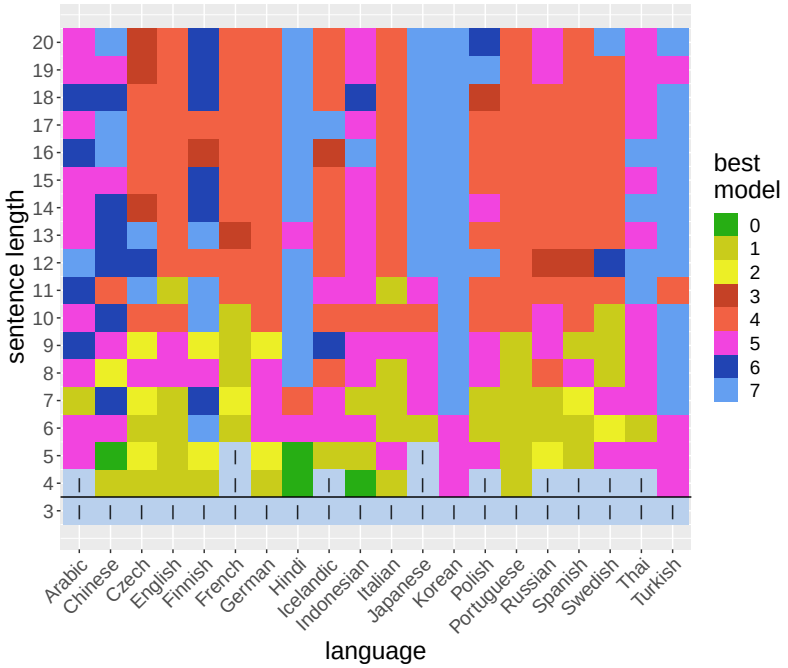


(a) Best model for every language and sentence length.

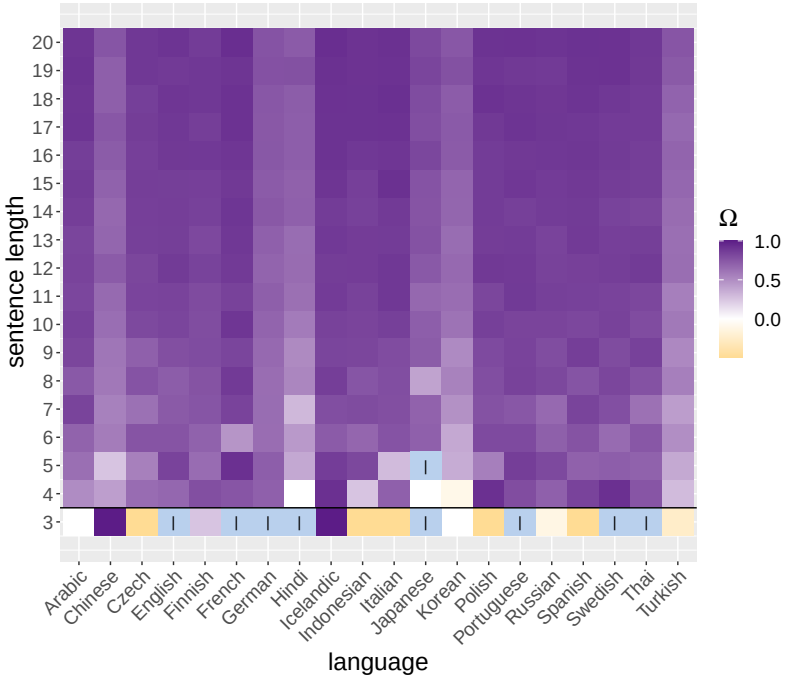


(b) $\langle\Omega\rangle$ for each language and sentence length: orange signals negative values, white signals values around 0, and purple signals positive values.

Figure 11: Relation between Ω score and best model in PUD. The barred gray cells indicate the sentence lengths which have not been observed, or that were excluded from model selection according to the representativeness threshold. Sentence lengths are cut at $n = 20$ to ease visualization. Model 0 refers to Model 0.0. In (b), orange signals negative values, white signals values around 0, and purple signals positive values.



(a) Best model for every language and sentence length.



(b) $\langle\Omega\rangle$ for each language and sentence length: orange signals negative values, white signals values around 0, and purple signals positive values.

Figure 12: Relation between Ω score and best model in PSUD. The format is the same as in Figure 11.

6 Discussion

First, we focus on the two hypotheses object of study, namely that syntactic dependency distances are distributed following two exponential regimes (H_1), and that the break-point shows low variation across languages (H_2). Our results provide strong evidence for both hypotheses in a large group of languages, mainly Indo-European, consistently across annotation styles. Second, we reflect on the parameters yielding the best fit and pay attention to the greater steepness of the first regime with respect to the second one, and the homogeneity of the estimated slopes across languages. Finally, we discuss the relation between the best estimated model and the closeness of syntactic dependencies as captured by the optimality score Ω (Ferrer-i-Cancho et al., 2022), and summarize the effect of annotation style.

6.1 The reality of two regimes

6.1.1 The shape of the distribution

As it is often the case, the path to the truth seems to lie in the middle. We could neither generalize to all languages hypothesis H_1 (supported by 13/20 languages in PUD and by 11/20 in PSUD), first advanced by Ferrer-i-Cancho (2004), nor fully corroborate the finding that dependency distances are power-law distributed as reported for Chinese (Liu, 2007). However, we provided evidence for a possible explanation integrating both: a two-regime model in which the first regime is either exponential or power-law distributed, and the second one follows an exponential decay. A two-regime model is found in all languages when mixing sentences of different lengths (Table 7), while two regimes are robustly found for the majority of languages when specific sentence lengths are considered (Figure 6). However, while the picture is clear and consistent in the first case, discussion on sentences of specific lengths requires further elaboration. The shape of the distribution depends on the length of the sequence (Figure 6), which is expected by the relation between DDm and sentence length. Processing short distances implies lower cognitive effort and robust statistical evidence suggests that DDm might irrelevant or be canceled out by other word order principles in short sequences (Ferrer-i-Cancho, 2024; Ferrer-i-Cancho and Gómez-Rodríguez, 2021; Ferrer-i-Cancho et al., 2022). Then, the varying intensity of the pressure for minimization yields different distributions in different areas of the sentence length domain, which we characterized with the following (potentially overlapping) regions (Figure 6, Figure 11 and Figure 12).

- *Random linear arrangement.* In short sentences (approximately $n \leq 6$) DDm might be neglectable or weak enough to be surpassed by other word order principles (Ferrer-i-Cancho, 2024; Ferrer-i-Cancho and Gómez-Rodríguez, 2021; Ferrer-i-Cancho et al., 2022), resulting in Model 0 (green tiles in Figure 6, Figure 11 and Figure 12) sometimes being the best one to describe the distribution. Where it is not Model 0, a model with a single regime is the best one.

- *Single chunk.* Up to roughly 13 words the best model is mainly one of 1, 2, or 5 (yellow and pink tiles in Figure 6, Figure 11 and Figure 12) in most languages. This possibly indicates that the sentence can be processed as a single chunk when the number of words is small enough, and dependencies must be highly local to allow for this.

- *Two regimes.* The bulk of the sentence length domain is characterized by the presence of two-regime models (red and blue tiles in Figure 6, Figure 11 and Figure 12). In these longer sentences the burden on STM becomes heavier, and two regimes might emerge from the breaking down of the sentence into chunks. After a very steep decrease in probability, a long dependency becomes more likely in order to link a chunk to the previous one.

- *No consistent pattern.* For long (and rare) sentences no clear pattern appears, as the scarcity of examples for large sentence lengths introduces variability in the estimation of the best model.

6.1.2 On power laws

When mixing sentences of distinct length, the best model is always a two regime model (Table 7). Across sentence lengths, the majority of languages have a model with two regimes as the most frequent best one, and a few languages in both collections show a power-law behavior (Table 8). Nevertheless, setting a rather high representativeness threshold dramatically reduces evidence for single-regime power-law, especially in PUD (Figure 7). This is for instance the case with Chinese in both collections. In spite of this, for Arabic, Indonesian, and Thai the most frequent best model is robustly Model 5 when the SUD annotation style is used.

Although Chinese has been argued to follow a single-regime power law (Liu, 2007), our findings indicate that Chinese is better fitted by a two-regime model with an initial power-law regime (Model 6 or 7) when mixing sentences of any length (Table 7). However, if the representativeness threshold is set to a low value (Figure 7), a single-regime power law (Model 5) can be retrieved, but such a low threshold casts doubts on the statistical strength of the best model when mixing sentences of distinct length. In contrast, the claim of a power law for Chinese is supported clearly for sentences of fixed length, where Model 5 is the most frequent best model across sentence lengths (Table 8).

Overall, two exponential regimes are the most common distribution for both mixed and fixed sentence lengths. However, what our analysis also proposes is that power laws can well describe the distribution in the first regime for some languages (mainly non Indo-European) when sentence lengths are mixed, as well as the distribution for specific sentence lengths for a small subset of them. Importantly, power-laws can also arise from undersampling, as highlighted by our representativeness analysis (Figure 7). In previous research it has been argued that power-laws could emerge from mixing sentence lengths in which

distances are distributed following an exponential curve (Ferrer-i-Cancho and Liu, 2014; Stumpf and Porter, 2012). Our research invalidates this argument (at least in the scope of our sample of languages), and identifies instances of another sort of mixing: for Arabic, Indonesian, and Thai in PSUD, mixing sentence lengths that are individually power-law distributed results in a distribution with two regimes with a power-law in the 1st regime, suggesting that further investigation is required in this direction.

6.1.3 Tail variability

Plots of the best model against the real data allows one to visually assess the quality of its fit to the data (Figure 5 for PUD and Figure 15 for PSUD). The best models are able to very well capture the shape of the bulk of the distribution and the initial bending in all languages. However, they are not always able to fully capture the variability along the tail of the distribution. To begin with, noise naturally emerges for longer distances, which belong to rare long sentences. As we explained above, there are lengths for which only one sentence is observed. Taking this into account, the deviation from the best model could suggest the possible presence of an unveiled pattern for some languages. We hypothesize the existence of more than one break-point, implying incremental executions of a "chunk-and-pass" mechanism (Christiansen and Chater, 2016).

However, introducing more regimes would greatly increase both the complexity of estimation (maximum likelihood estimation already requires putting particular care in the estimation of 3/4 parameters, see section 4), and the risk of overfitting the data. Thus, a thorough and rigorous methodology would need to be employed for such modelling, which should be the subject of future research.

6.2 The homogeneity of the break-point

The break-point values we estimated are largely homogeneous across languages, and average values of 5 (PUD) and 4 (PSUD) words, with small variation. These values are consistent with the literature on limitations of short term memory: in no language d^* exceeds the "magical number" 7 (Miller, 1956), and the bulk of the values is centered at 4 ± 1, which is generally recognized to be the working memory limitation on a wide range of tasks (Cowan, 2001).

Nevertheless, some variability can still be observed, especially among the break-points of sentences of different lengths within a language. In fact, an implicit assumption of H_2 is that the value estimated globally for a given language is a reliable approximation of the constraint acting at the sentence level, and this can be verified by looking at the break-points estimated for each given length. We find that for languages in which H_1 holds (two exponential regimes), the distribution of d^* across sentence lengths is very narrow, and centered around the global value of d^*. The break-points estimated in Models 6-7 are more variable, but they still vary in a rather small range compared to the range of variation of the actual

sentences (Table 3 and Table 4).

The average length in words of simple declarative sentences is 3.7 (from 2.6 in Turkish up to 5.4 in Mandarin) (Fenk-Oczlon and Pilz, 2021).[1]. We believe that this variability in the size clauses is captured by our breakpoint (Figure 8) but this issue should be the subject of future research with a linguistic or cognitive focus.

6.3 Patterns in probability decay across regimes

Given the large applicability of the two-regime models, we take closer look to the speed of probability decay. The slopes observed across languages are quite narrowly distributed around the same values (Figure 9). It is interesting to notice that while the first slope is significantly larger in PSUD, q_2 shows little variation in the two collections. This suggests that, depending on annotation style, the distribution of the dependencies within word chunks will change, but beyond word chunks, the chunking mechanism follows a similar structure. Another interesting pattern concerns the steepness of the first regime with respect to the second one. When mixing sentences of different lengths the first regime is always steeper than the second one (Figure 9) and this is virtually always the case even when considering specific sentence lengths, with a very few exceptions in short sentences (Figure 10). This provides additional support for the "chunk-and-pass" paradigm (Christiansen and Chater, 2016). An explanation for that pattern could be that, when memory limits are approached in long enough sentences, the current chunk needs to be closed, and a new longer dependency becomes more likely in order to link the forthcoming chunk (thus reducing the speed of probability decay). The two regimes (and in particular Model 4) may be found even if the real distribution is Model 0, given their similar BIC scores (Figure 13). However, Model 4 could only mimic a linear curve (Model 0) if the second regime was steeper than the first one.

6.4 The best model versus the optimality of syntactic dependency distances

In section 6.1, we have described how the shape of the distribution varies depending on sentence length. Here, we aim to understand the interplay with different degrees of pressure for DDm for long versus short sentences. Previous research has pointed out at how $\langle\Omega\rangle$ is smaller in short sentences, likely due to DDm being neglected or canceled out by other word order principles (Ferrer-i-Cancho, 2024; Ferrer-i-Cancho and Gómez-Rodríguez, 2021; Ferrer-i-Cancho et al., 2022). We provide additional evidence for this phenomenon by unravelling a direct correspondence between sentences where $\langle\Omega\rangle$ is close to 0, and those in which the best model is Model 0 (Table 12). Moreover, we observe a relation between the intensity of DDm and the best model for the distribution. Namely, as pressure for minimization increases with sentence length, the best model changes (Figure 11 and Figure 12). While correlation does not

[1] The data can be found in the Supplementary Material (Sheet 1)

imply causation, it is crucial to understand that both the pressure for DDm and the best model for the distribution of syntactic dependency distances are not homogeneous through sentence length. Thus, distances belonging to sentences of different length are subject to different pressures, and this should be taken into account when trying to model the distribution. In particular, these different levels of pressure could yield different mechanisms. Indeed, the more complex distributions – those with two regimes – tend to emerge for long enough sentence length, when the pressure for DDm is stronger, likely calling for a structured processing mechanism.

6.5 The effect of annotation style

So far we have observed commonalities and differences between PUD and PSUD. Overall, the main qualitative results are robust to annotation style, supporting the soundness of the observed patterns, but some differences emerge. The discussion on the origins of such differences is open, and is connected to the fundamental question of whether an annotation style is a more accurate representation of our brain's functioning or the linguistic processing than the other, or whether different styles simply mirror different aspects of this functioning or processing. While providing a rather descriptive account of such differences, we partly attempt to address this question.

6.5.1 The shape of the distribution

The first main point concerns the very high consistency in the best estimated models (Figure 8). However, there are a few exceptions, which we classified in two types: differences in right truncation, and in the distribution in the first regime. The latter is clearly of greater interest and it concerns two languages, Japanese and Indonesian, both having Models 3-4 as the best model in PUD, and Model 7 in PSUD, but showing a very different behaviour. For Japanese, the best models estimated on specific sentence lengths and by mixing all sentence lengths are highly consistent within each collection, and in both cases the break-point value is $d^* = 6$. This suggests a real difference in probability decay within a chunk depending on the chosen annotation guidelines, but also conveys the concreteness of the quantified limit on memory for such language. On the other hand, for Indonesian we find mixed evidence, both in terms of estimated break-point, which goes from $d^* = 3$ in PUD to $d^* = 7$ in PSUD, and in terms of best model for fixed sentence lengths (which is consistently a one-regime power-law in PSUD). In fact, this takes us to one of the main differences between annotation styles (Figure 7): while in PUD the only language showing some evidence for a single power-law regime for fixed sentence lengths is Arabic, in PSUD we have three languages strongly supporting the reality of such distribution. For Arabic, Indonesian, and Thai, the two regimes observed for mixed sentence lengths contradict what is found when sentence lengths are analysed in isolation. This seems to reflect Simpson's paradox, a phenomenon according to which a statistical trend disappears when single groups are considered, and suggests that there is some

variability left to explain.

6.5.2 The break-point

We have seen in Figure 8 how the break-points estimated in both collections cover the same portion of domain, ranging from 3 to 7. However, while in PUD there is no settling around a particular value, in PSUD d^* is nearly uniform at $d^* = 4$, especially within Models 3-4. This raises the following questions: is this regularity given by chance? Or does it mirror a better ability of SUD to capture syntactic relations as formed by our minds? Given that – besides individual differences – the overall structure of the brain is assumed to be the same for all humans, the constraint on memory is expected to be uniform across languages (hence the motivation for H_2). Thus, one could speculate that SUD annotation style is actually more capable of unveiling this uniformity, that is assumed to exist.

6.5.3 Dependency distance minimization

SUD guidelines have been found to lead to shorter dependency distances (Ferrer-i-Cancho et al., 2022; Osborne and Gerdes, 2019; Yan and Liu, 2021). When dependency distances are conveniently normalized with respect to the gap between the random baseline and the minimum baseline, SUD reflects distances that are closer to optimality (Ferrer-i-Cancho et al., 2022). Such ability of SUD to reflect dependency distance minimization of effects is confirmed by our findings. In fact, despite predicting a power-law decay in the first regime for two more languages compared to PUD, q_1 is significantly higher in PSUD (Figure 9). This entails a faster decay in probability within the chunk, related to the predominance of short local dependencies in PSUD. Moreover, the values of Ω computed in the PSUD collection are generally larger (tiles in Figure 12 (b) are darker than in Figure 11 (b)), confirming a stronger degree of optimization of dependency distances in the SUD framework (Ferrer-i-Cancho et al., 2022).

7 Conclusion

Two decades after the first observations on the peculiar shape of the distribution of syntactic dependency distances (Ferrer-i-Cancho, 2004), some new light has been shed. A crucial finding is that the probability of observing a dependency – independently of the length of the sentence it belongs to – is best described by a double-regime model. Furthermore, the finding also holds at a finer-grained level, distinctively considering each sentence length. In this setting, for the great majority of languages a double-regime model is the most frequent one, while the few remaining languages show a power-law decay as the most frequent, partly in accordance with what has been found concerning a Chinese treebank, where however sentences of mixed lengths were analysed (Liu, 2007). Furthermore, the break-point between the two regimes estimated globally for each language varies in a small range ($3 \leq d^* \leq 7$), which becomes even

narrower when only languages in which H_1 holds are considered. In fact, H_2 seems to be related to the probability distribution observed in the first regime, leading to the identification of a group of languages where probability follows a two-regime exponential decay (H_1), and within which the break-point is very similar (H_2). This group is mainly populated by Indo-European languages. However, languages from this family are over-represented in our sample, and other interesting patterns could emerge if a larger group of languages from other families where analysed. These considerations hold independently of annotation style, but it has not escaped our attention that in PSUD values of d^* for such group are almost uniform at 4, a widely accepted quantification of the constraint on short term memory (Cowan, 2001). This could, in our opinion, reflect a higher sensitivity of SUD annotation style to the way in which our minds create and process language, bringing to light a "universal" constraint which is not language dependent. Another general pattern emerged is the relation between the speeds of the decays, whereas probability in the first regime is always faster than in the second one. As already pointed out, this result may look paradoxical: if cognitive pressure induces a decay in probability as syntactic dependency distance increases, why does such a decay slows down beyond the breakpoint? (Ferrer-i-Cancho, 2017)? In the framework of language processing, these findings provide strong support for the "chunk-and-pass" mechanism (Christiansen and Chater, 2016). In fact, the presence of these two different regimes could actually mirror the two different speeds at which probability decays within a chunk and beyond it. In physical terms, the true units of measurement of distance may change: within the word chunk the unit of distance are words whereas, beyond the word chunk, the actual distance may be chunks in the hidden space of incremental processing of the sentence. The breakpoint and the slow down after the breakpoint may arise because we have imposed the use of words as unit of measurement independently of the stage of syntactic parsing. In our view, this appears to be the most reasonable and pertinent explanation for the observed systematic decrease in the strength of DDm, but we do not exclude that other explanations could as well be plausible. Future work could further investigate the distribution in the second regime, exploring different combinations of exponential and power-law decay. Then, the possible presence of more than one break-point could be explored. Importantly, to understand the extent to which the observed phenomena can be considered universal, the same analysis shall be performed on a wider set of languages.

Acknowledgments

We are grateful to Jan Andres for helpful comments. We have benefited from discussions with G. Fenk-Oczlon and the contents of the talk that she gave at the 16th International Cognitive Linguistics Conference (August 2023), "Working memory constraints: Implications for efficient coding of messages". They helped us in terms of presenting STM for a general audience and to find a linguistic interpretation to the breakpoint. SP is funded by the grant "Thesis abroad 2021/2022" from the University of Milan. RFC

is supported by a recognition 2021SGR-Cat (01266 LQMC) from AGAUR (Generalitat de Catalunya). SP and RFC are supported by the grants AGRUPS-2022, AGRUPS-2023 and AGRUPS-2024 from Universitat Politècnica de Catalunya.

References

Alemany-Puig, L., Esteban, J., Ferrer-i-Cancho, R. (2021). The Linear Arrangement Library. A new tool for research on syntactic dependency structures. *Proceedings of the Second Workshop on Quantitative Syntax (Quasy, SyntaxFest 2021)*, 1–16. https://aclanthology.org/2021.quasy-1.1

Anderson, D. R., Burnham, K. P. (2004). Model selection and multi-model inference. *Second. NY: Springer-Verlag*, *63*(2020), 10.

Bolker, B. (2007). bbmle: tools for general maximum likelihood estimation. https://doi.org/10.32614/cran.package.bbmle

Campos, D., Bartumeus, F., Méndez, V., Andrade, J. S., Espadaler, X. (2016). Variability in individual activity bursts improves ant foraging success. *Journal of The Royal Society Interface*, *13*(125), 20160856. https://doi.org/10.1098/rsif.2016.0856

Christiansen, M. H., Chater, N. (2016). The now-or-never bottleneck: A fundamental constraint on language. *Behavioral and Brain Sciences*, *39*, e62. https://doi.org/10.1017/S0140525X1500031X

Cowan, N. (2001). The magical number 4 in short-term memory: A reconsideration of mental storage capacity. *Behavioral and Brain Sciences*, *24*(1), 87–114.

Cowan, N. (2017). The many faces of working memory and short-term storage. *Psychonomic Bulletin & Review*, *24*(4), 1158–1170. https://doi.org/10.3758/s13423-016-1191-6

Dagpunar, J. (1988). *Principles of random variate generation*. Oxford University Press, USA.

Devroye, L. (1986). *Non-uniform random variate generation(originally published with*. Springer-Verlag. http://cg.scs.carleton.ca/~luc/rnbookindex.html

Ercsey-Ravasz, M., Markov, N. T., Lamy, C., Essen, D. C. V., Knoblauch, K., Toroczkai, Z., Kennedy, H. (2013). A predictive network model of cerebral cortical connectivity based on a distance rule. *Neuron*, *80*(1), 184–197. https://doi.org/http://dx.doi.org/10.1016/j.neuron.2013.07.036

Esteban, J., Ferrer-i-Cancho, R. (2017). A correction on Shiloach's algorithm for minimum linear arrangements of trees. *Society for Industrial and Applied Mathematics*, *46*(3), 1146–1151. https://doi.org/https://doi.org/10.1137/15M1046289

Fenk-Oczlon, G., Pilz, J. (2021). Linguistic complexity: Relationships between phoneme inventory size, syllable complexity, word and clause length, and population size. *Frontiers in Communication*, *6*, 66. https://doi.org/10.3389/fcomm.2021.626032

Ferrer-i-Cancho, R. (2017). A commentary on "The now-or-never bottleneck: A fundamental constraint on language", by Christiansen and Chater (2016). *Glottometrics*, *38*, 107–111. http://hdl.handle.net/2117/107857

Ferrer-i-Cancho, R. (2004). Euclidean distance between syntactically linked words. *Physical Review E*, *70*, 056135.

Ferrer-i-Cancho, R. (2019). The sum of edge lengths in random linear arrangements. *Journal of Statistical Mechanics: Theory and Experiment*, *2019*, 053401. https://doi.org/10.1088/1742-5468/ab11e2

Ferrer-i-Cancho, R. (2024). The optimal placement of the head in the noun phrase. The case of demonstrative, numeral, adjective and noun. *Journal of Quantitative Linguistics*, in press. https://www.arxiv.org/abs/2402.10311

Ferrer-i-Cancho, R., Gómez-Rodríguez, C. (2021). Anti dependency distance minimization in short sequences. a graph theoretic approach. *Journal of Quantitative Linguistics*, *28*(1), 50–76.

Ferrer-i-Cancho, R., Gómez-Rodríguez, C., Esteban, J. L., Alemany-Puig, L. (2022). Optimality of syntactic dependency distances. *Physical Review E*, *105*(1), 014308.

Ferrer-i-Cancho, R., Liu, H. (2014). The risks of mixing dependency lengths from sequences of different length. *Glottotheory*, *5*(2), 143–155.

Futrell, R., Mahowald, K., Gibson, E. (2015). Large-scale evidence of dependency length minimization in 37 languages. *Proceedings of the National Academy of Sciences*, *112*, 10336–10341.

Gerdes, K., Guillaume, B., Kahane, S., Perrier, G. (2018). SUD or surface-syntactic Universal Dependencies: An annotation scheme near-isomorphic to UD. *Proceedings of the Second Workshop on Universal Dependencies (UDW 2018)*, 66–74. https://doi.org/10.18653/v1/W18-6008

Jiang, J., Liu, H. (2015). The effects of sentence length on dependency distance, dependency direction and the implications–based on a parallel English–Chinese dependency treebank. *Language Sciences*, *50*, 93–104.

Lewis, R. L., Vasishth, S. (2005). An activation-based model of sentence processing as skilled memory retrieval. *Cognitive Science*, *29*(3), 375–419. https://doi.org/10.1207/s15516709cog0000_25

Liu, H. (2007). Probability distribution of dependency distance. *Glottometrics*, *15*, 1–12.

Liu, H. (2008). Dependency distance as a metric of language comprehension difficulty. *Journal of Cognitive Science*, *9*, 159–191.

Liu, H. (2009). Probability distribution of dependencies based on a Chinese dependency treebank. *Journal of Quantitative Linguistics*, *16*(3), 256–273. https://doi.org/10.1080/09296170902975742

Liu, H., Xu, C., Liang, J. (2017). Dependency distance: A new perspective on syntactic patterns in natural languages. *Physics of life reviews*, *21*, 171–193.

Lu, Q., Liu, H. (2016). Does dependency distance distribute regularly. *Journal of Zhejiang University (Humanities and Social Science)*, *2*(4), 63–76.

Mathy, F., Feldman, J. (2012). What's magic about magic numbers? chunking and data compression in short-term memory. *Cognition*, *122*(3), 346–362. https://doi.org/https://doi.org/10.1016/j.cognition.2011.11.003

Miller, G. A. (1956). The magical number seven, plus or minus two: Some limits on our capacity for processing information. *The Psychological Review*, *63*(2), 81–97. http://www.musanim.com/miller1956/

Muller, M. E. (1958). An inverse method for the generation of random normal deviates on large-scale computers. *Mathematics of Computation*, *12*(63), 167–174.

Myung, I. J. (2003). Tutorial on maximum likelihood estimation. *Journal of Mathematical Psychology*, *47*(1), 90–100.

Newman, M. E. (2005). Power laws, Pareto distributions and Zipf's law. *Contemporary Physics*, *46*(5), 323–351. https://doi.org/10.1080/00107510500052444

Nivre, J., Zeman, D., Ginter, F., Tyers, F. (2017). Universal Dependencies. *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Tutorial Abstracts*. https://aclanthology.org/E17-5001

Osborne, T., Gerdes, K. (2019). The status of function words in dependency grammar: A critique of Universal Dependencies (UD). *Glossa: a journal of general linguistics*, *4*(1). https://doi.org/10.5334/gjgl.537

Shiloach, Y. (1979). A minimum linear arrangement algorithm for undirected trees. *SIAM Journal on Computing*, *8*(1), 15–32.

Stumpf, M. P. H., Porter, M. A. (2012). Critical truths about power laws. *Science*, *335*(6069), 665–666. https://doi.org/10.1126/science.1216142

Wagenmakers, E.-J., Farrell, S. (2004). AIC model selection using Akaike weights. *Psychonomic bulletin & review*, *11*(1), 192–196.

Wimmer, G., Altmann, G. (1999). *Thesaurus of univariate discrete probability distributions*. STAMM Verlag.

Yan, J., Liu, H. (2021). Semantic roles or syntactic functions: The effects of annotation scheme on the results of dependency measures. *Studia Linguistica*, *76*(2), 406–428. https://doi.org/10.1111/stul.12177

Appendices

A Model derivation

Here we detail the mathematical derivation of the non-standard models in Section 2.

Model 0.1 We consider a general model for sentences of varying length, defined as

$$p(d) = \sum_{n=min(n)}^{max(n)} p(d|n)\, p(n),$$

where $p(d|n)$ is the conditional probability of d given that the sentence length has n words, $p(n)$ is the proportion of sentences having length n, and $\min(n)$ and $\max(n)$ are the minimum and maximum observed values of n in the sample. By definition, $p(d|n)$ satisfies two conditions, i.e. $p(d|n) = 0$ when $d \notin [1, n)$ and

$$\sum_{d=1}^{n-1} p(d|n) = 1.$$

Thanks to these two conditions, it is easy to see that $p(d)$ is properly normalized, that is

$$\begin{aligned} \sum_{d=1}^{max(n)-1} p(d) &= \sum_{d=1}^{max(n)-1} \sum_{n=min(n)}^{max(n)} p(d|n)\, p(n) \\ &= \sum_{n=min(n)}^{max(n)} p(n) \sum_{d=1}^{n-1} p(d|n) \\ &= 1 \end{aligned}$$

By setting $p(d|n)$ according to the null hypothesis of a random shuffling of the words of a sentence of n words (1), which satisfies the two conditions above, we obtain

$$p(d) = \sum_{n=min(n)}^{max(n)} \frac{n-d}{\binom{n}{2}}\, p(n).$$

Model 2 We define the cumulative distribution of Model 1 as

$$P_1(d) = \sum_{d'=1}^{d} p_1(d').$$

where $p_1(d)$ is defined as in 2. Model 2 is derived via renormalization of Model 1 after right-truncation, that is

$$p_2(d) = \frac{p_1(d)}{P_1(d_{max})},$$

where

$$\begin{aligned} P_1(d_{max}) &= \sum_{d=1}^{d_{max}} q(1-q)^{d-1} \\ &= 1 - (1-q)^{d_{max}}. \end{aligned}$$

Hence

$$p_2(d) = \frac{q(1-q)^{d-1}}{1-(1-q)^{d_{max}}}.$$

Double-regime models Now we use $p_1(d)$ to refer to the definition of $p(d)$ for $d \leq d^*$ and $p_2(d)$ to refer to the definition of $p(d)$ for $d \geq d^*$. The definition of Models 3, 4, 6, 7 follows the template

$$p(d) = \begin{cases} p_1(d) = c_1 f_1(d) & \text{if } d \leq d^* \\ p_2(d) = c_2 f_2(d) & \text{if } d^* \leq d \leq d_{max}, \end{cases}$$

For models 3 and 6, one simply sets d_{max} to ∞. Thus, the assumption $p_1(d) = p_2(d)$ yields

$$c_2 = \tau c_1$$

with

$$\tau = \frac{f_1(d)}{f_2(d)}.$$

Recalling the definitions of the models (Table 1), it is easy to see that, for models 3 and 4,

$$\tau = \frac{(1-q_1)^{d^*-1}}{(1-q_2)^{d^*-1}}.$$

whereas for models 6 and 7,

$$\tau = \frac{d^{*^{-\gamma}}}{(1-q)^{d^*-1}}.$$

Let us derive the normalization factor c_1 for Models 3, 4, 6, 7 with the help of

$$\begin{aligned} S_1 &= \sum_{d=1}^{d^*} f_1(d) \\ S_2 &= \sum_{d=d^*}^{d_{max}} f_2(d). \end{aligned}$$

The normalization condition

$$\sum_{d=1}^{d_{max}} p(d) = 1$$

yields

$$c_1 = \frac{1}{S_1 + \tau S_2}. \tag{11}$$

For Models 3 and 4, S_1 is

$$S_1 = \sum_{d'=0}^{d^*-1} (1-q_1)^{d'} = \frac{1-(1-q_1)^{d^*}}{q_1}.$$

S_2 depends on the truncation point. For Model 3, the assumption $q > 0$ (thus $\lim_{d_{max}\to\infty}(1-q)^{d_{max}} = 0$) produces

$$\begin{aligned} S_2 &= \sum_{d'=d^*}^{\infty} (1-q_2)^{d'} \\ (1-q_2)S_2 &= S_2 - (1-q_2)^{d^*} + (1-q_2)^{\infty} \\ S_2 &= \frac{(1-q_2)^{d^*}}{q_2} \end{aligned} \tag{12}$$

By substituting S_1, S_2 and τ in 11, c_1 for Model 3 becomes

$$c_1 = \frac{q_1 q_2}{q_2 + (1-q_1)^{d^*-1}(q_1 - q_2)}$$

after some algebra.

In Model 4, probabilities are restricted up to d_{max}, thus

$$S_2 = \sum_{d'=d^*}^{d_{max}-1} (1-q_2)^{d'} = \frac{(1-q_2)^{d^*} - (1-q_2)^{d_{max}}}{q_2}. \tag{13}$$

Again, plugging S_1, S_2, and τ into 11 produces c_1 for Model 4, that is

$$c_1 = \frac{q_1 q_2}{q_2 + (1-q_1)^{d^*-1}(q_1 - q_2 - q_1(1-q_2)^{d_{max}-d^*+1})}$$

after some algebra.

For the second pair of double-regime models (Models 6 and 7), combining a zeta and a geometric distribution, S_1 is

$$S_1 = \sum_{d=1}^{d^*} d^{-\gamma} = H(d^*, \gamma),$$

while the second regime is shared with Models 3-4, so that S_2 corresponds to 12 for Model 6 and to 13 for Model 7. Then, the normalization factors are obtained again through 11, so that for Model 6

$$c_1 = \frac{q}{qH(d^*, \gamma) + d^{*-\gamma}(1-q)},$$

while for Model 7

$$c_1 = \frac{q}{qH(d^*, \gamma) + d^{*-\gamma}(1 - q - (1-q)^{d_{max}-d^*+1})}$$

after some algebra.

B Log-likelihood functions

In our setting, the log-likelihood of a model is

$$\mathcal{L} = \log \prod_{i=1}^{N} p(d_i) = \sum_{i=1}^{N} \log p(d_i) = \sum_{d=1}^{max(d)} f(d), \log p(d).$$

Next we derive the log-likelihood functions for each model with the help of Table 1.

For Model 0.0, where d_{max} is the only free parameter, we have

$$\begin{aligned}
\mathcal{L} &= \sum_{d=1}^{max(d)} f(d) \log \left(\frac{2(d_{max} + 1 - d)}{d_{max}(d_{max} + 1)} \right) \\
&= \sum_{d=1}^{max(d)} f(d) \left[\log \left(\frac{2}{d_{max}(d_{max} + 1)} \right) + \log(d_{max} + 1 - d) \right] \\
&= N \log \left(\frac{2}{d_{max}(d_{max} + 1)} \right) + W,
\end{aligned}$$

where

$$\begin{aligned}
N &= \sum_{d=1}^{max(d)} f(d) \\
W &= \sum_{d=1}^{max(d)} f(d) \log(n - d).
\end{aligned}$$

For Model 0.1, in which the observed sentence lengths are supplied and there is no free parameter, we have

$$\begin{aligned}
\mathcal{L} &= \sum_{n=min(n)}^{max(n)} \sum_{d=1}^{max(d)} f(d) \log \frac{2(n - d)}{n(n - 1)} \\
&= \sum_{n=min(n)}^{max(n)} \sum_{d=1}^{max(d)} f(d) \left[\log \frac{2}{n(n - 1)} + \log(n - d) \right] \\
&= \sum_{n=min(n)}^{max(n)} \left[N_n \log \frac{2}{n(n - 1)} + W_n \right]
\end{aligned}$$

where

$$\begin{aligned}
W_n &= \sum_{d=1}^{max(d)} f(d) \log(n - d) \\
N_n &= \sum_{d=1}^{max(d)} f(d)
\end{aligned}$$

in sentences of length n. For the *geometric models*, we start from the derivation of the right-truncated

version, namely Model 2

$$\begin{aligned}\mathcal{L} &= \sum_{d=1}^{max(d)} f(d) \log \frac{q(1-q)^{d-1}}{1-(1-q)^{d_{max}}} \\ &= \sum_{d=1}^{max(d)} f(d) \left[\log \frac{q}{1-(1-q)^{d_{max}}} + (d-1)\log(1-q)\right] \\ &= N \log \frac{q}{1-(1-q)^{d_{max}}} + (M-N)\log(1-q),\end{aligned}$$

where $M = \sum d = 1^{max(d)} f(d)\ d$. Then, the log-likelihood function of Model 1 as a particular case of that of Model 2 in which $d_{max} = \infty$, i.e.

$$\mathcal{L} = N \log q + (M-N)\log(1-q)$$

since $q > 0$ and thus $\lim_{d_{max}\to\infty}(1-q)^{d_{max}} = 0$. For the *two-regime geometric models*, we start from the log-likelihood of Model 4, i.e.

$$\begin{aligned}\mathcal{L} &= \sum_{d=1}^{d^*} f(d) \log \left[c_1(1-q_1)^{d-1}\right] + \sum_{d=d^*+1}^{max(d)} f(d) \log \left[c_2(1-q_2)^{d-1}\right] \\ &= \sum_{d=1}^{d^*} f(d)\ [\log c_1 + (d-1)\log(1-q_1)] + \sum_{d=d^*+1}^{max(d)} f(d)\ [\log c_2 + (d-1)\log(1-q_2)] \\ &= N^* \log c_1 + (M^*-N^*)\log(1-q_1) + (N-N^*)\log c_2 + \\ &\quad (M-M^*-N+N^*)\log(1-q_2) \\ &= N^* \log c_1 + (N-N^*)\log c_2 + (M^*-N^*)\log\frac{1-q_1}{1-q_2} + (M-N)\log(1-q_2)\end{aligned}$$

where

$$\begin{aligned}M^* &= \sum_{d=1}^{d^*} f(d)\ d \\ N^* &= \sum_{d=1}^{d^*} f(d),\end{aligned}$$

while c_1 and c_2 are defined as explained in Section 2 for Model 3 and 4. Thus, the log-likelihood functions of Model 3 and Model 4 only differ in the computation of c_1 and c_2. For the *right truncated power-law* distribution, namely Model 5,

$$\begin{aligned}\mathcal{L} &= \sum_{d=1}^{max(d)} f(d) \log \frac{d^{-\gamma}}{H(d_{max},\gamma)} \\ &= \sum_{d=1}^{max(d)} f(d)\ [-\gamma \log d - \log H(d_{max},\gamma)] \\ &= -\gamma M' - N \log H(d_{max},\gamma),\end{aligned}$$

where $M' = \sum_{d=1}^{max(d)} f(d)\log(d)$. Finally, for Models 6 and 7, we start from the derivation of Model 7,

$$\begin{aligned}\mathcal{L} &= \sum_{d=1}^{d^*} f(d)\log(c_1 d^{-\gamma}) + \sum_{d=d^*+1}^{max(d)} f(d)\log\left[c_2(1-q)^{d-1}\right] \\ &= \sum_{d=1}^{d^*} f(d)\left[\log c_1 - \gamma\log(d)\right] + \sum_{d=d^*+1}^{max(d)} f(d)\left[\log c_2 + (d-1)\log(1-q)\right] \\ &= N^*\log c_1 - \gamma M'^* + (N-N^*)\log c_2 + (M-M^*-N+N^*)\log(1-q),\end{aligned}$$

while c_1 and c_2 are defined as explained in Section 2 for Model 6 and 7.

C Model selection validation

C.1 Artificial data generation

In the following, let $p_x(d)$ be the probability of d according to Model x. The parameter values used to generate each model are reported in Table 14, while sample size is $N = 10^4$ for each model. For right-truncated models sentence length is set to $n = 20$, and the maximum distance is set to $d_{max} = 19$. Then Model 0.0 is equivalent to Model 0 with $n = 20$. We choose $\gamma = 1.6$ because it has been obtained from fitting a right-truncated Zeta distribution to a Chinese treebank (Liu, 2007).

Table 14: Parameter values used to generate artificial samples. Here Model 0 is the same as Model 0.0.

Model	d_{max}	q	q_1	q_2	d^*	γ
0	19	-	-	-	-	-
1	-	0.2	-	-	-	-
2	19	0.2	-	-	-	-
3	-	-	0.5	0.1	4	-
4	19	-	0.5	0.1	4	-
5	19	-	-	-	-	1.6
6	-	0.2	-	-	4	1.6
7	19	0.2	-	-	4	1.6

Models 1 and 2 For the geometric distribution and its right-truncated version, namely Model 1 and Model 2, we use Dagpunar's fast inversion method (Dagpunar, 1988). For Model 1, a random distance d is obtained by producing a random uniform deviate x and then calculating

$$d = 1 + \left\lfloor \frac{\log x}{\lambda} \right\rfloor,$$

where $\lambda = \log(1-q)$, and q is the parameter of the desired geometric distribution. For Model 2, a value of d is produced until $d \leq d_{max}$.

Model 5 For Model 5, we employed the algorithm proposed by Devroye to efficiently generate a random deviate from a zeta distribution (Devroye, 1986), adapting it to allow for right-truncation. The algorithm is called one or more times until a value of d such that $d \leq d_{max}$ is obtained.

Model 0 and two-regime models For the sake of simplicity, random samples of Model 0 and of the two-regime models, namely Models 3, 4, 6, and 7, are generated using a tabular inversion method (Devroye, 1986; Muller, 1958). This method generates artificial distances in a pre-specified range, namely $d \in [1, \delta]$. Thus, in order to simulate Models 3 and 6 – which do not have a right-truncation – we set $\delta = 10^6$ to ensure that $p(d) \approx 0$ for $d \geq \delta$, while for Models 0, 4 and 7 we have $\delta = d_{max} = 19$. For simplicity, the method is implemented through binary search. Hence, a random deviate is produced in time $O(\log \delta)$.

C.2 Results

For each model, the best model yields a good visual fit to each artificially generated sample Figure 14. Indeed, the real underlying distribution is identified for every artificial random sample (Table 15 and Figure 13). See Figure 13 for the magnitude of the difference in BIC score between a given model and the best model (the model that minimizes BIC). The BIC of the double-regime models is always close to the BIC of the best model. The reason resides in the greater flexibility allowed by the existence of the break-point, which is however compensated by the penalty imposed on the additional parameter by the BIC score 9. Another concern could rise from the fitting of the random sample of Model 0, in which the BIC score of Model 4 is not much larger than that of the best model. Indeed, two geometric regimes could mimic the linearity of Model 0, but only in the case in which the second regime decays faster than the second. The values of the parameters estimated by maximum likelihood for each artificially generated random sample are shown in Table 16. See Table 17 for a comparison of the estimated values against the real values used to generate the data for each of the artificial samples. The error between the real values and the optimal parameters is either 0 or very small. In particular, maximum likelihood seems prone to underestimate the real value rather than the opposite.

Table 15: BIC scores on artificial random samples. Each row corresponds to a random sample generated by a given model. In each row, we show first the name of the true model and then we show the AIC values of each candidate model. The true Model 0 is Model 0 that is equivalent to Model 0.1 here. The candidate Model 0 is Model 0.0.

True model	Model 0	Model 1	Model 2	Model 3	Model 4	Model 5	Model 6	Model 7
0	55570.65	57527.90	55881.07	55750.98	55615.79	56724.85	55963.35	55697.17
1	60974.42	50037.40	50040.08	50049.99	50056.13	53256.86	50054.30	50057.24
2	51569.46	48995.12	48739.88	48801.18	48755.09	50086.11	48993.61	48757.98
3	76657.57	54995.13	55004.33	51553.49	51561.32	52694.62	51681.76	51689.79
4	51638.78	47122.05	46967.51	45359.06	44595.90	44716.30	44818.89	44685.95
5	49460.37	39609.04	39602.60	37251.07	37076.27	36864.76	36937.93	36881.25
6	61658.20	39436.89	39446.10	37196.76	37204.83	37684.75	37133.38	37141.30
7	48909.08	38217.08	38217.08	36343.48	36242.39	36270.85	36239.71	36175.27

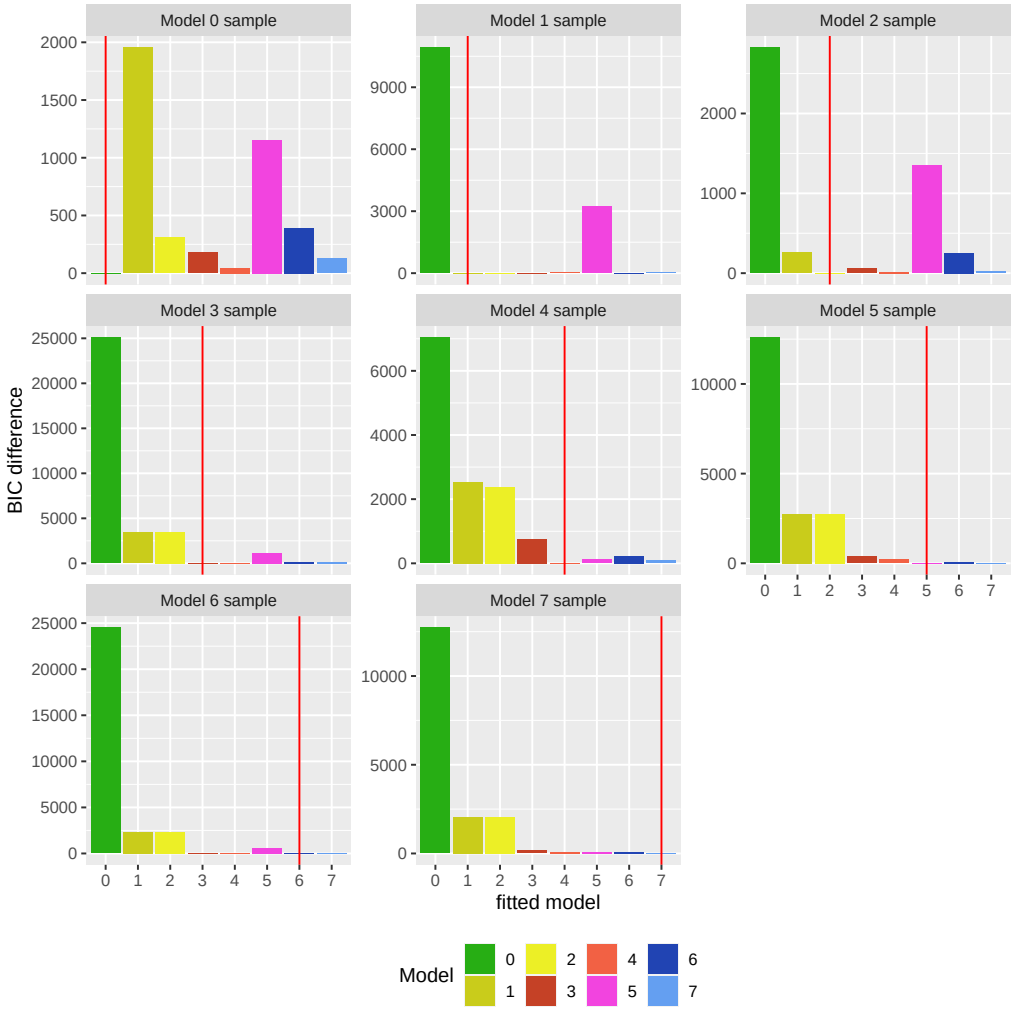


Figure 13: BIC differences in artificial random samples. The BIC difference is the difference between the BIC of the model and the BIC of the best model (the model that minimizes the BIC for the sample). The red vertical line indicates the best model according to BIC.

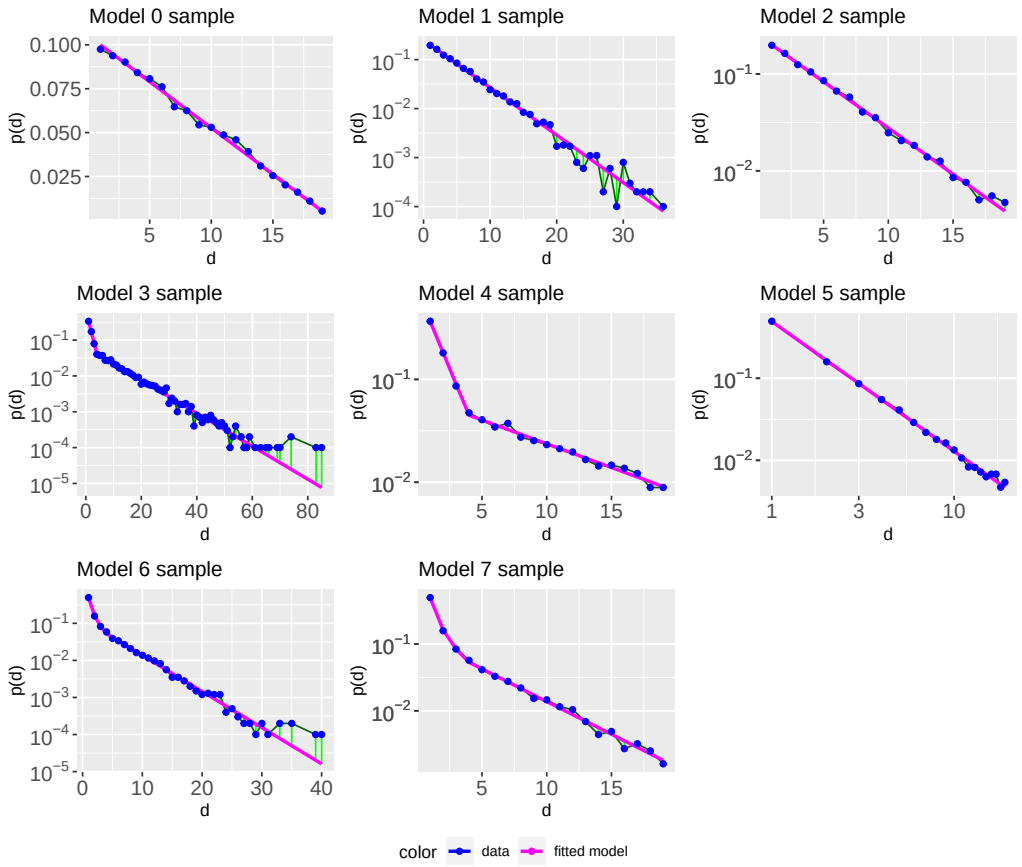


Figure 14: $p(d)$, the probability that a dependency link is formed between words at distance d according to the best model for artificially generated samples.

D Model selection results

We here report the results of model selection when sentences of any length are mixed for each language. See Table 18 and Table 20 for the AIC scores for PUD and PSUD, respectively; see Table 19 and Table 21 for the corresponding AIC differences. The AIC difference of a model is defined as the difference of its AIC and the AIC of the best model (the model that minimizes AIC) (Anderson and Burnham, 2004). The parameters estimated by maximum likelihood are shown in Table 22 for PUD and in Table 23 for PSUD. Finally, see Figure 15 for the best model fitted to the empirical distribution for languages in PSUD.

Table 18: AIC scores of each model in the PUD collection on sentences of mixed lengths. Here Model 0 refers to Model 0.1.

Language	Model 0	Model 1	Model 2	Model 3	Model 4	Model 5	Model 6	Model 7
Arabic	84577	55188	55190	52011	52012	52264	51866	51864
Chinese	86281	68121	68123	65826	65827	67025	65737	65738
Czech	68424	48729	48731	47872	47872	49499	48212	48214
English	85821	60122	60124	59402	59402	62649	60055	60056
Finnish	52893	38423	38425	37955	37956	38921	37927	37928
French	109197	71748	71750	69420	69418	72291	70944	70946
German	86626	68510	68512	66699	66700	68821	66955	66957
Hindi	107388	83075	83077	75832	75828	76676	75788	75782
Icelandic	73752	50242	50244	49411	49413	51252	49716	49718
Indonesian	76351	50676	50678	48875	48876	49596	48916	48917
Italian	104223	68313	68315	66370	66369	69289	67786	67788
Japanese	135512	93746	93748	85222	85221	87112	86524	86525
Korean	64173	50365	50367	45474	45472	45647	45337	45332
Polish	66255	45103	45105	43851	43852	44719	43956	43958
Portuguese	100042	67213	67215	65361	65361	68010	66557	66559
Russian	70474	47879	47881	46750	46751	48291	47201	47203
Spanish	101194	67353	67355	65377	65376	67934	66641	66643
Swedish	75639	53807	53809	53135	53136	55623	53633	53635
Thai	108081	69553	69555	65717	65718	66242	65519	65521
Turkish	62864	51439	51441	47362	47358	47697	47250	47245

Table 19: AIC differences of each model in the PUD collection on sentences of mixed lengths. Here Model 0 refers to Model 0.1.

Language	Model 0	Model 1	Model 2	Model 3	Model 4	Model 5	Model 6	Model 7
Arabic	32712.96	3323.95	3325.95	146.57	147.28	399.60	1.44	0.00
Chinese	20544.43	2383.68	2385.68	88.77	90.07	1288.62	0.00	0.72
Czech	20552.34	857.52	859.51	0.00	0.70	1627.44	340.07	341.97
English	26419.25	720.65	722.64	0.00	0.63	3247.77	652.91	654.90
Finnish	14965.50	496.10	498.00	27.96	29.28	993.94	0.00	1.02
French	39779.35	2329.86	2331.86	1.94	0.00	2873.31	1526.42	1528.33
German	19927.31	1811.19	1813.19	0.00	1.49	2122.61	256.63	258.34
Hindi	31605.72	7292.04	7294.03	49.53	45.49	893.65	5.61	0.00
Icelandic	24340.78	831.13	833.13	0.00	1.94	1841.66	305.52	307.52
Indonesian	27476.04	1800.38	1802.38	0.00	1.08	721.10	41.17	41.86
Italian	37854.79	1944.06	1946.06	1.10	0.00	2920.54	1417.74	1419.67
Japanese	50290.71	8524.56	8526.56	0.58	0.00	1890.84	1302.96	1303.51
Korean	18840.12	5032.98	5034.98	141.29	139.91	314.13	4.31	0.00
Polish	22403.20	1251.25	1253.25	0.00	0.73	867.34	104.69	106.27
Portuguese	34681.44	1852.22	1854.22	0.00	0.34	2649.16	1196.46	1198.33
Russian	23723.74	1129.28	1131.28	0.00	1.37	1540.80	451.36	453.32
Spanish	35817.93	1976.75	1978.74	1.07	0.00	2557.41	1264.87	1266.59
Swedish	22503.61	671.54	673.54	0.00	0.88	2487.59	497.78	499.75
Thai	42561.20	4033.29	4035.29	197.05	198.82	722.50	0.00	1.23
Turkish	15618.79	4193.97	4195.96	117.05	112.74	452.16	5.51	0.00

Table 20: AIC scores of each model in the PSUD collection on sentences of mixed lengths. Here Model 0 refers to Model 0.1.

Language	Model 0	Model 1	Model 2	Model 3	Model 4	Model 5	Model 6	Model 7
Arabic	83964	49718	49720	45461	45461	45444	45249	45248
Chinese	86004	66658	66660	63187	63188	63852	63043	63043
Czech	67743	43660	43662	42048	42049	42711	42164	42166
English	84875	51868	51870	49860	49860	50895	50293	50295
Finnish	52389	35247	35249	34450	34451	35057	34434	34436
French	108313	62309	62311	57458	57458	58294	58037	58036
German	85580	64289	64291	62110	62111	63642	62229	62230
Hindi	106846	79001	79003	68777	68760	69540	68495	68483
Icelandic	72807	42153	42155	39927	39929	40396	40000	40002
Indonesian	75765	45212	45214	41927	41928	42005	41809	41808
Italian	103370	59445	59447	55354	55354	56373	56039	56040
Japanese	135293	88560	88562	72667	72667	72316	71831	71831
Korean	64065	49797	49799	44509	44508	44683	44366	44364
Polish	65708	40765	40767	38674	38676	39020	38697	38698
Portuguese	99155	58440	58442	54381	54381	55150	54846	54846
Russian	69999	43597	43599	41455	41457	41963	41541	41543
Spanish	100350	58907	58909	54617	54615	55352	55105	55103
Swedish	74683	46288	46290	44584	44586	45399	44815	44817
Thai	107549	63835	63837	57879	57881	57879	57564	57565
Turkish	62784	50771	50773	46448	46442	46728	46325	46318

Table 21: AIC differences of each model in the PSUD collection on sentences of mixed lengths. Here Model 0 refers to Model 0.1.

Language	Model 0	Model 1	Model 2	Model 3	Model 4	Model 5	Model 6	Model 7
Arabic	38715.46	4469.92	4471.92	212.61	213.36	196.35	0.49	0.00
Chinese	22961.93	3615.17	3617.16	144.40	145.21	809.83	0.67	0.00
Czech	25695.55	1612.08	1614.08	0.00	1.23	663.54	116.10	117.76
English	35015.25	2008.17	2010.17	0.09	0.00	1034.96	433.57	435.32
Finnish	17954.96	812.85	814.84	15.62	16.93	622.93	0.00	1.41
French	50855.16	4851.44	4853.44	0.05	0.00	836.35	579.48	578.26
German	23469.70	2179.07	2181.07	0.00	1.12	1532.51	118.75	120.29
Hindi	38363.24	10518.04	10520.03	294.32	277.30	1056.95	11.70	0.00
Icelandic	32879.81	2225.42	2227.42	0.00	1.79	468.28	72.30	74.23
Indonesian	33956.92	3404.16	3406.16	119.39	119.95	196.85	1.41	0.00
Italian	48016.12	4091.05	4093.05	0.00	0.83	1019.71	685.68	686.27
Japanese	63462.28	16729.23	16731.23	836.09	836.29	485.06	0.28	0.00
Korean	19701.40	5432.90	5434.90	145.71	144.65	319.57	1.91	0.00
Polish	27034.25	2090.57	2092.57	0.00	1.52	346.04	23.00	23.88
Portuguese	44774.69	4059.49	4061.49	0.00	0.29	769.23	465.71	465.70
Russian	28543.92	2141.67	2143.67	0.00	1.47	508.13	86.14	87.86
Spanish	45735.15	4292.48	4294.48	2.28	0.00	736.75	490.45	487.74
Swedish	30099.05	1703.55	1705.55	0.00	1.64	815.39	231.31	233.16
Thai	49985.45	6271.66	6273.66	315.48	317.16	315.43	0.00	1.29
Turkish	16466.09	4453.49	4455.47	129.96	124.00	410.41	7.45	0.00

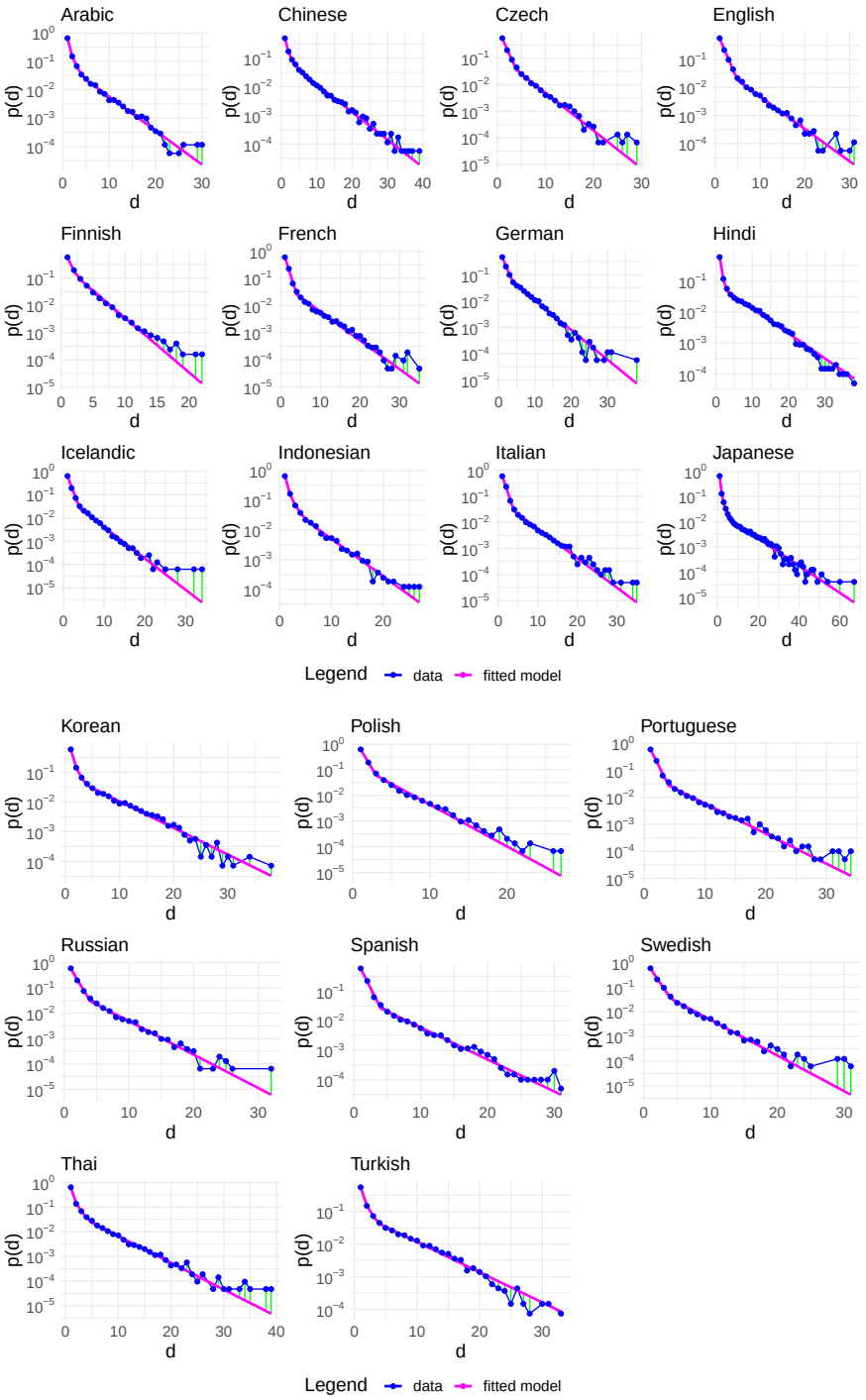
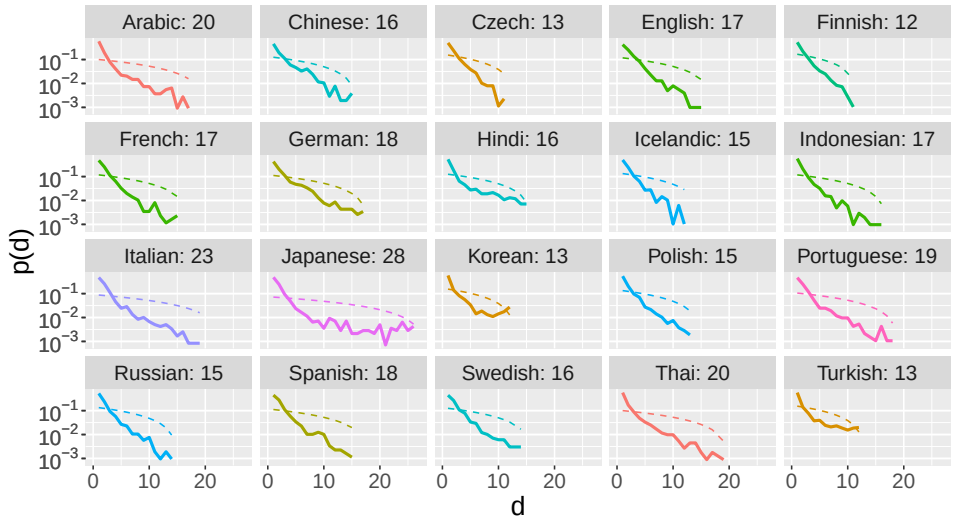


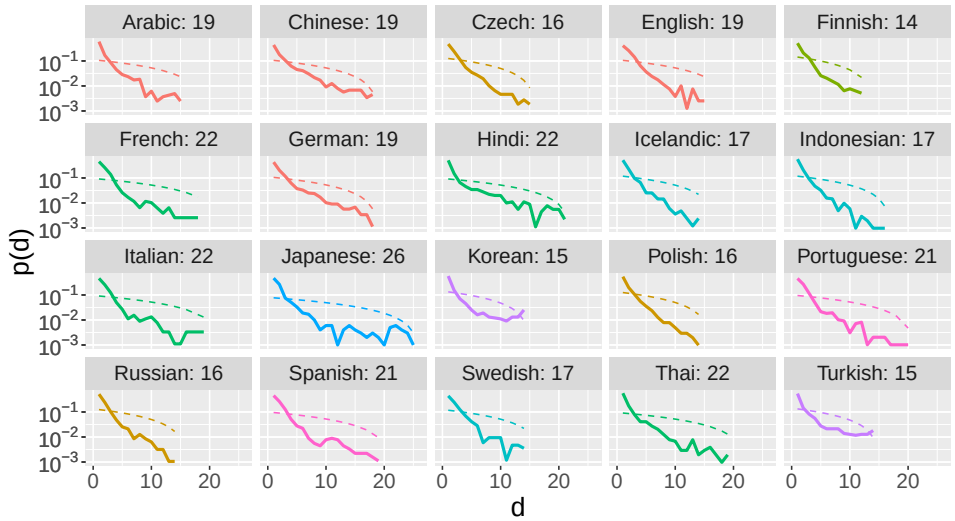
Figure 15: $p(d)$, the probability that a dependency link is formed between words at distance d according to the data and the best model for every language in PSUD.

E The distribution of dependency distances for characteristic sentence lengths.

See Figure 16 (a-b) for the distributions in PUD, for modal and mean sentence length respectively; see Figure 17 (a-b) for PSUD. As mean sentence length, we use the results of rounding the actual mean sentence length to the nearest integer.

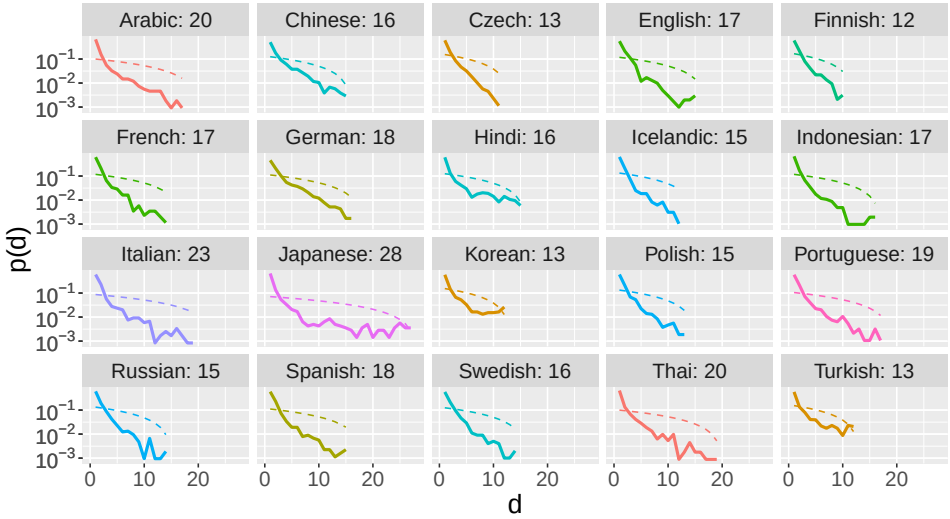


(a) Modal sentence length.

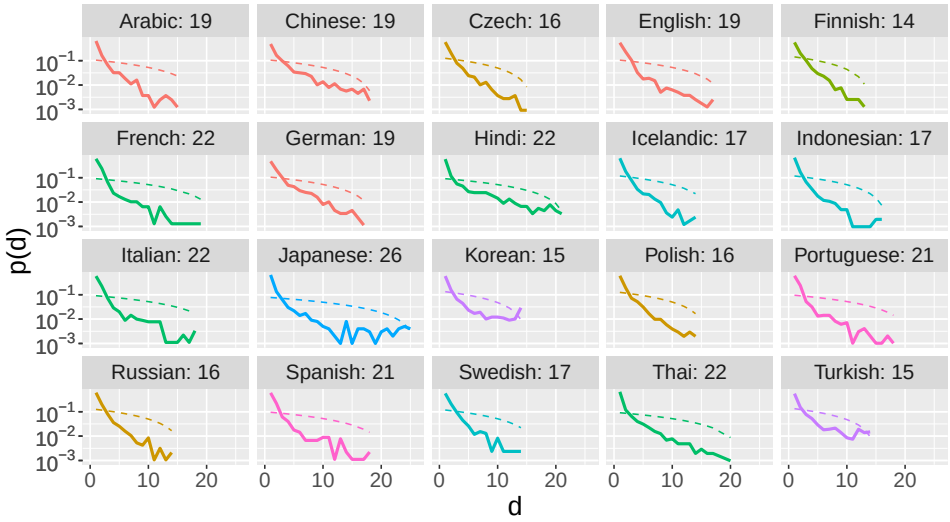


(b) Mean sentence length.

Figure 16: $p(d)$, the probability that linked words are at distance d in sentences of modal (a) and mean (b) length for each language in PUD. Mode and mean are shown next to the respective language label. The dashed line shows the probability according to Model 0 (1). Points where $p(d) = 0$ are not shown.



(a) Modal sentence length.



(b) Mean sentence length.

Figure 17: $p(d)$, the probability that linked words are at distance d in sentences of modal (a) and mean (b) length for each language in PSUD. The format is the same as in Figure 16.

Table 13: The log-likelihood $\mathcal{L}$ for each of the probability mass functions. K is the number of free parameters, N is the sample size, M is the sum of distances weighted by frequency, i.e. $M = \sum_{i=1}^{max(d)} f(d_i)d_i$, M^* is the same sum up to d^*, i.e. $M^* = \sum_{i=1}^{d^*} f(d_i)d_i$, M' is the sum of log distances weighted by frequency, i.e. $M' = \sum_{i=1}^{max(d)} f(d_i)log(d_i)$, and W is such that $W = \sum_{i=1}^{max(d)} f(d_i)log(d_{max} + 1 - d_i)$. N^* is the sum of distance frequencies up to d^*, i.e. $N^* = \sum_{i=1}^{d^*} f(d_i)$, and M'^* is M' up to d^*, i.e. $M'^* = \sum_{i=1}^{d^*} f(d_i)log(d_i)$. For model 3, c_1 id defined in 3; for model 4, c_1 is defined in 5; for models 3 and 4, $c_2 = \tau c_1$ with τ defined as in 4. For model 6, c_1 id defined in 6; for model 7, c_1 is defined in 8; for models 6 and 7, $c_2 = \tau c_1$ with τ defined as in 7. Finally, $W_n = \sum_{i=1}^{max(d)} f(d_i)\log(n - d_i))$ and $N_n = \sum_{i=1}^{max(d)} f(d_i)$ in sentences of length n.

Model	**Function**	K	$\mathcal{L}$
0.0	Null model	1	$N\log(\frac{2}{d_{max}(d_{max}+1)}) + W$
0.1	Extended Null model	0	$\sum_{n=min(n)}^{max(n)}[N_n \log\left(\frac{2}{n(n-1)}\right) + W_n]$
1	Geometric	1	$N\log q + (M-N)\log(1-q)$
2	Right-truncated geometric	2	$N\log\left(\frac{q}{1-(1-q)^{d_{max}}}\right) + (M-N)\log(1-q)$
3	Two-regime geometric	3	$N^*\log c_1 + (N-N^*)\log c_2 + (M^*-N^*)\log\left(\frac{1-q_1}{1-q_2}\right) + (M-N)\log(1-q_2)$
4	Two-regime right-truncated geometric	4	$N^*\log c_1 + (N-N^*)\log c_2 + (M^*-N^*)\log\left(\frac{1-q_1}{1-q_2}\right) + (M-N)\log(1-q_2)$
5	Right-truncated zeta distribution	2	$-\gamma M' - N\log(H(d_{max},\gamma))$
6	Two-regime zeta-geometric	3	$N^*log(c_1) - \gamma M'^* + (N-N^*)log(c_2) + (M - M^* - N + N^*)log(1-q)$
7	Two-regime right-truncated zeta-geometric	4	$N^*log(c_1) - \gamma M'^* + (N-N^*)log(c_2) + (M - M^* - N + N^*)log(1-q)$

Table 16: Best parameters estimated in artificial random samples by maximum likelihood. The 1st column indicates the true model while the header row indicates the candidate model. Here Model 0 refers to Model 0.0.

Model	$\max(d)$	d_{max}	q	q	d_{max}	q_1	q_2	d^*	q_1	q_2	d^*	d_{max}	d_{max}	γ	γ	q	d^*	γ	q	d^*	d_{max}
		0	1	2		3			4				5		6			7			
0	19	19	0.142	0.100	19	0.088	0.643	17	0.071	0.242	13	19	19	0.522	0.302	0.343	13	0.264	0.206	11	19
1	36	36	0.200	0.200	36	0.200	0.543	34	0.191	0.203	5	36	36	1.204	0.274	0.201	2	0.278	0.201	2	36
2	19	19	0.210	0.197	19	0.197	0.622	18	0.199	0.091	17	19	19	0.985	0.418	0.221	4	0.295	0.198	2	19
3	85	85	0.160	0.160	85	0.502	0.101	4	0.502	0.101	4	85	85	1.422	1.373	0.103	5	1.373	0.102	5	85
4	19	19	0.228	0.219	19	0.549	0.166	3	0.503	0.101	4	19	19	1.242	1.234	0.644	18	1.332	0.097	6	19
5	19	19	0.314	0.313	19	0.628	0.201	3	0.641	0.180	3	19	19	1.582	1.578	0.610	18	1.588	0.058	15	19
6	40	40	0.317	0.317	40	0.623	0.204	3	0.624	0.204	3	40	40	1.718	1.613	0.201	4	1.614	0.201	4	40
7	19	19	0.333	0.332	19	0.613	0.221	3	0.622	0.206	3	19	19	1.608	1.541	0.299	12	1.610	0.202	4	19

Table 17: Best estimated parameters, real parameters used to generate the artificial samples, and their difference. Here Model 0 refers to Model 0.0. The header row indicates the true model.

	0	1	2	3				4				5		6			7			
	d_{max}	q	q	d_{max}	q_1	q_2	d^*	q_1	q_2	d^*	d_{max}	d_{max}	γ	γ	q	d^*	γ	q	d^*	d_{max}
estimated	19	0.200	0.197	19	0.502	0.101	4	0.503	0.101	4	19	19	1.582	1.613	0.201	4	1.610	0.202	4	19
real	19	0.200	0.200	19	0.500	0.100	4	0.500	0.100	4	19	19	1.600	1.600	0.200	4	1.600	0.200	4	19
error	0	0.000	−0.003	0	0.002	0.001	0	0.003	0.001	0	0	0	−0.018	0.013	0.001	0	0.010	0.002	0	0

Table 22: Best parameters estimated by maximum likelihood on sentences of mixed lengths in the PUD collection.

			1	2		3			4					5		6				7	
Language	$\max(n)$	$\max(d)$	q	q	d_{max}	q_1	q_2	d^*	q_1	q_2	d^*	d_{max}	d_{max}	γ	γ	q	d^*	γ	q	d^*	d_{max}
Arabic	50	30	0.434	0.434	30	0.668	0.269	3	0.668	0.269	3	30	30	1.973	1.840	0.240	7	1.842	0.238	7	30
Czech	44	29	0.418	0.418	29	0.499	0.270	5	0.500	0.269	5	29	29	1.837	1.385	0.347	3	1.385	0.347	3	29
German	50	42	0.322	0.322	42	0.485	0.222	4	0.485	0.222	4	42	42	1.675	1.351	0.234	5	1.351	0.234	5	42
English	56	31	0.395	0.395	31	0.453	0.256	6	0.454	0.255	6	31	31	1.759	0.930	0.380	2	0.930	0.380	2	31
Finnish	39	21	0.446	0.446	21	0.639	0.388	2	0.562	0.360	3	21	21	1.855	1.440	0.374	3	1.440	0.374	3	21
French	54	36	0.396	0.396	36	0.491	0.197	6	0.492	0.196	6	36	36	1.826	1.462	0.296	4	1.462	0.296	4	36
Hindi	58	42	0.303	0.303	42	0.671	0.175	3	0.672	0.174	3	42	42	1.735	1.798	0.170	4	1.800	0.169	4	42
Indonesian	47	27	0.442	0.442	27	0.629	0.305	3	0.630	0.304	3	27	27	1.935	1.713	0.295	5	1.714	0.294	5	27
Icelandic	52	34	0.432	0.432	34	0.531	0.309	4	0.531	0.309	4	34	34	1.888	1.412	0.359	3	1.412	0.359	3	34
Italian	60	35	0.403	0.403	35	0.490	0.207	6	0.491	0.206	6	35	35	1.830	1.446	0.307	4	1.446	0.307	4	35
Japanese	70	65	0.337	0.337	65	0.521	0.119	6	0.522	0.118	6	65	65	1.849	1.754	0.130	13	1.755	0.130	13	65
Korean	43	37	0.364	0.364	37	0.700	0.197	3	0.701	0.197	3	37	37	1.886	1.886	0.180	5	1.888	0.179	5	37
Polish	39	27	0.449	0.449	27	0.569	0.288	4	0.569	0.287	4	27	27	1.936	1.653	0.324	4	1.653	0.324	4	27
Portuguese	58	34	0.396	0.396	34	0.504	0.234	5	0.504	0.233	5	34	34	1.812	1.436	0.301	4	1.436	0.301	4	34
Russian	47	32	0.440	0.440	32	0.529	0.261	5	0.529	0.260	5	32	32	1.916	1.564	0.332	4	1.564	0.332	4	32
Spanish	58	32	0.399	0.399	32	0.510	0.231	5	0.510	0.230	5	32	32	1.816	1.460	0.300	4	1.460	0.300	4	32
Swedish	49	31	0.404	0.404	31	0.462	0.257	6	0.462	0.257	6	31	31	1.791	1.214	0.358	3	1.214	0.358	3	31
Thai	63	38	0.409	0.409	38	0.653	0.258	3	0.653	0.258	3	38	38	1.933	1.770	0.230	7	1.770	0.230	7	38
Turkish	37	34	0.343	0.343	34	0.670	0.201	3	0.671	0.200	3	34	34	1.797	1.812	0.195	4	1.815	0.194	4	34
Chinese	49	39	0.323	0.323	39	0.569	0.233	3	0.569	0.232	3	39	39	1.694	1.439	0.219	6	1.440	0.219	6	39

Table 23: Best parameters estimated by maximum likelihood on sentences of mixed lengths in the PSUD collection.

			1	2		3			4			5		6			7				
Language	$\max(d)$	$\max(n)$	q	q	d_{max}	q_1	q_2	d^*	q_1	q_2	d^*	d_{max}	d_{max}	γ	γ	q	d^*	γ	q	d^*	d_{max}
Arabic	50	30	0.488	0.488	30	0.727	0.263	3	0.727	0.263	3	30	30	2.150	2.101	0.241	5	2.102	0.241	5	30
Czech	44	29	0.475	0.475	29	0.602	0.279	4	0.602	0.279	4	29	29	2.027	1.814	0.306	5	1.814	0.306	5	29
German	50	38	0.355	0.355	38	0.523	0.228	4	0.523	0.228	4	38	38	1.758	1.489	0.244	5	1.489	0.244	5	38
English	56	31	0.472	0.472	31	0.575	0.234	5	0.575	0.233	5	31	31	2.025	1.810	0.299	5	1.810	0.299	5	31
Finnish	39	22	0.490	0.490	22	0.625	0.362	3	0.625	0.362	3	22	22	2.007	1.715	0.369	4	1.715	0.369	4	22
French	54	35	0.470	0.470	35	0.652	0.216	4	0.652	0.216	4	35	35	2.090	2.007	0.206	9	2.008	0.204	9	35
Hindi	58	38	0.329	0.329	38	0.729	0.162	3	0.730	0.161	3	38	38	1.843	2.298	0.171	3	2.302	0.170	3	38
Indonesian	47	27	0.500	0.500	27	0.717	0.283	3	0.717	0.283	3	27	27	2.150	2.055	0.254	7	2.056	0.251	7	27
Icelandic	52	34	0.521	0.521	34	0.653	0.266	4	0.653	0.266	4	34	34	2.184	2.029	0.295	6	2.030	0.295	6	34
Italian	60	35	0.475	0.475	35	0.645	0.229	4	0.645	0.228	4	35	35	2.087	1.981	0.226	8	1.982	0.225	8	35
Japanese	70	67	0.367	0.367	67	0.699	0.121	4	0.699	0.121	4	67	67	2.060	2.218	0.117	6	2.218	0.117	6	67
Korean	43	38	0.370	0.370	38	0.713	0.193	3	0.713	0.193	3	38	38	1.915	2.016	0.187	4	2.018	0.186	4	38
Polish	39	27	0.501	0.501	27	0.688	0.313	3	0.688	0.313	3	27	27	2.112	1.970	0.287	6	1.971	0.286	6	27
Portuguese	58	34	0.469	0.469	34	0.642	0.226	4	0.642	0.225	4	34	34	2.074	1.971	0.224	8	1.972	0.223	8	34
Russian	47	32	0.489	0.489	32	0.629	0.262	4	0.629	0.262	4	32	32	2.092	1.930	0.282	6	1.930	0.282	6	32
Spanish	58	31	0.469	0.469	31	0.646	0.222	4	0.647	0.221	4	31	31	2.071	1.981	0.221	8	1.982	0.219	8	31
Swedish	49	31	0.482	0.482	31	0.607	0.283	4	0.607	0.283	4	31	31	2.050	1.832	0.309	5	1.832	0.309	5	31
Thai	63	39	0.454	0.454	39	0.723	0.240	3	0.723	0.240	3	39	39	2.100	2.053	0.221	5	2.053	0.221	5	39
Turkish	37	33	0.350	0.350	33	0.680	0.199	3	0.681	0.198	3	33	33	1.818	1.859	0.194	4	1.863	0.193	4	33
Chinese	49	39	0.335	0.335	39	0.619	0.219	3	0.619	0.219	3	39	39	1.755	1.574	0.200	7	1.574	0.199	7	39