Pseudowords: Generatucing, Evaluadating, and their Impactfluence

Thesis submitted in partial fulfillment
of the requirements for the degree of

Master of Science
in
Computational Linguistics
by Research

by

Mukund Choudhary
2018114015
mukund.choudhary@research.iiit.ac.in

International Institute of Information Technology
Hyderabad - 500 032, INDIA
June, 2023

International Institute of Information Technology
Hyderabad, India

CERTIFICATE

It is certified that the work contained in this thesis, titled "Pseudowords: Generatucing, Evaluadating, and their Impactfluence" by Mukund Choudhary, has been carried out under my supervision and is not submitted elsewhere for a degree.

Friday 30th June, 2023 — Advisor: Bapi Raju Surampudi

Friday 30th June, 2023 — Co-Advisor: Dipti Misra Sharma

“curiouser & curiouser” - Alice (in the Wonderland)

This thesis is dedicated to the language that exists outside a dictionary.

Acknowledgments

I am immensely grateful to several people (and cats) equally for their constructive feedback, fire-in-the-hole ideas, expressions of disappointment, heartfelt wishes, hours of annotations, exhilarating deliberations, warming up my food as well as my emotions, and all the other kinds of very kind & familial support I received while working towards and enduring through my work on this thesis.

My advisors, Bapi Raju Surampudi & Dipti Misra Sharma have been amazing guides in this cross-domain exploration of computational linguistics and cognitive science. I have always wanted to innovate in the field of linguistics, as someone who knows my interests in & around the field (through hours of not sipping tea but office-hour conversations), Prof. Dipti's recommendation for me to work on something around Aphasia in Indian Languages was the most apt & exciting idea. On the other hand, since Prof. Bapi's discussion with me on why I feel interested in joining the Cognitive Science Lab on this project, I have been constantly learning everything related to cognition, behavioral experimentation, analysis, and more. Prof. Bapi has also been very keen and indulging in view things from the linguistics side of things as much as Prof. Dipti was formative for me to be able to grow in the field and in my understanding of computational linguistics. Together, they have been motivating, patient, positively critical, and encouraging of the ideas I got to try out. Without their synergy, this project would have never taken shape or possibly be poised as of help to Indian language research in this currently understudied field.

I am also thankful to Prof. Priyanka Srivastava and Prof. Aditi Mukherjee for their incessant trust in me, tons of shared chuckles & chortles, giving me a lot of opportunities to learn, and as people who encouraged me to do more like publishing a course project to a leading international conference as a very under confident junior or to be confident in mentoring freshers about to start their journey into linguistics. All of these experiences contributed in major but subtle ways to the forming of the thesis.

As I am running out of adjectives to thank with, there cannot be enough for my collaborators whose work exist in this thesis or out of it. Aditya, Ishan, Tathagata, Abhinav, and Gaurja have been stalwart champions of what I do and reliable homes to crash at while working through this thesis and all works leading to it. They have been rubber ducks, code monkeys, and shoulders-to-lean-on at all times this thesis would prove to be larger than me. Outside of the thesis they have kept me fresh & working on more experiments & experiences than I have had in my life before college. They have also enabled me safely in my bads, while cheering me wholly for my goods. The heart of this thesis is them. (contd.)

Next is the list of outstanding-at-their-work people I would like to thank for mentoring me in various ways & letting me steal their time, space, company, opinions, and quips throughout my years at IIIT Hyderabad includes Prof. Manish Shrivastava, Prof. Radhika Mamidi, Prof. Manish Gupta, Alok, Vandan, Pruthwik, Animesh, and Saujas. I am always in awe of our conversations here and there (actually mostly walking around LTRC & in classes), while impossibly aiming to be like them as I keep moving on my academic journey.

Outside of IIIT, I am indebted to these extremely experienced, humane, kind, and parent-figures who impacted me immensely and guided me to this point. Monojit Choudhury has been a constant inspiration in aiming to do more in linguistics and innovating creatively. From just before college till today, Priyanka & Ashutosh Tripathi have always pushed me to do more, lead more, be confident more, and live more fully with the same caring and wholesome attitude each time.

I also thank my other collaborators and friends who have helped me in so many ways that I would need a longer space to list them out comprehensively! I am very grateful to Sridhar, Rishav, Sumanth, E. Nikhil, Naimeesh, Mihir, Priyanshi, Dipayan, Yash, Utkarsh, Suvadeep, Madhukar, Vamshi, Pratyaksh, Shashwat, and a lot of others for all the support & good moments so far that kept me going!

Finally people who will never accept the infinite gratitude I have for them, my family. No way could I reach the end of this thesis or begin it without my sister, mom, and dad giving me all of their happiness, energy, faith, love, and peace-of-their-minds round-the-clock, all days of the year. This thesis is a mish-mash of all of that and them bearing all of my hiccups, pain-points, brick-walls I encountered while working on it and making it all vanish away with warm hugs.

Abstract

Pseudowords are a part of language that are not translatable to another, as they have no meaning attached to them while also having the constraint of sounding like a phonologically valid sequence under the desired language's native phonotactics. This thesis thus explores automated language-agnostic pseudoword generation, evaluation of them, and use of them outside psycholinguistics research and clinical use.

As the thesis progresses, we highlight current research, draw inspiration from close topics of study, build a pipeline to generate pseudowords and generate Hindi and English pseudoword candidates for further experiemntation. We make this reusable pipeline available on a public repository, as one of the deliverables of this work. Then we show how the current evaluation work in this field is very scarce and sew an evaluation framework with reproducible details on how to design and analyse a human-in-the-loop experiment for something as tricky as pseudoword judgement, conducted for a layman native speaker. After showing various ways to prod a pseudoword set for quality, we compare notes against past sets in English and present observations summarising how comparable they are. However as there is no Hindi pseudoword dataset yet, we add in psycholinguistic features on top of results of evaluation metrics per Hindi pseudoword and release "Soodkosh" another fully public and usable for research resource.

Finally, we conduct two separate studies involving pseudowords to show the application, impact, and importance of them across fields. The first study uses pseudowords to establish gradient between high-frequency words, low-frequency words, and non-sensical sequences of alphanumerics used as passwords. The aim of this study is to find correlation and its strength between the perceived security and memorability of a password/phrase. The other part of this chapter is an exploration into language models' performance on Aphasia classification and if replacing pseudowords can help them. This is as pseudowords like neologisms, mis-pronunciations, and other novel forms generated by Aphasic speakers are largely out-of-vocabulary to a standard languge model that functions off of a pile of mostly well-formed and coherent data. As these are not directly helpful to the field of Aphasia, this work replaces one possible hurdle to see if it is a feasible solution. However the results show that pseudowords are passively used as features and cannot be replaced directly.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Scope of the Thesis

From all the phonologically possible wordforms in a language, the meaningful ones are termed as *words*, the ones with no meaning attached to them as *pseudowords*, and the phonologically impossible fraction as *nonwords* [2] in this thesis. This is a compilation of my work on and around Pseudowords that aims to be a useful, informative, and reproducible work on that could be used as a first step to solve Pseudoword data scarcity in Indian languages, and a variety of tasks like visual word recognition [3] in lexical cognition studies, as a part of lexical-decision tasks in crucial places like Aphasia type distinction [4], and in NLP as used by [5] and [6], to probe the semantic abilities of a language model.

Hindi Pseudoword	Word likeness	WT_{mean}	FR_{mean}	FT_{mean}	comp	poly	n poly	1-C	half	C+	V+	CV+C	C+V+C+	C+V+C+#	Dist IPA	Dist Dev	Orthographic Length	Matras	Aksharas	Syllables	Phonemes
स्घोट	7.00	1,766.29	5.60	3,503.17	0	0	0	1	0	1	0	0	0	0	3	2	5	1	2.5	1	4
तजरील	6.83	1,769.99	5.67	5,125.85	0	0	0	0	0	0	0	0	1	2	2	2	5	1	4	2	6
पनिवेश	6.50	1,164.23	3.17	3,939.47	0	0	0	0	0	0	0	0	0	0	2	2	6	2	4	3	7
बामल्लें	6.50	1,228.56	4.83	4,170.71	0	0	1	0	0	0	0	1	1	1	4	3	8	3	3.5	3	7
अक्सित	6.33	2,139.27	3.17	4,870.91	0	1	0	1	0	0	0	0	0	0	2	1	6	1	3.5	2	5
हरखिनित्व	6.33	1,463.89	6.00	3,466.55	0	0	1	0	0	0	0	0	1	3	5	4	9	2	5.5	4	9
तामबूलियों	6.33	1,714.09	4.17	4,312.48	0	0	1	0	0	0	0	0	0	0	1	3	10	5	5	4	9
तॉल	6.20	766.31	5.80	3,105.96	0	0	0	1	0	0	0	0	0	0	1	1	3	1	2	1	3
तमघ	6.20	2,345.44	3.00	4,431.32	0	0	0	0	0	0	0	0	1	1	2	2	3	0	3	2	5
अनाकषत	6.17	1,779.15	3.50	3,770.44	0	0	1	0	0	0	0	0	0	0	3	2	6	1	5	3	7
बेशिध्र	6.17	1,471.50	5.00	4,163.03	0	0	1	0	0	0	0	0	1	1	4	2	7	2	3.5	3	6
परिमयता	6.17	1,016.85	4.83	3,916.52	0	0	1	0	0	0	0	0	1	1	4	2	7	2	5	4	9
कव	6.00	1,511.02	3.67	3,948.90	0	0	0	1	0	0	0	0	0	0	1	1	2	0	2	1	3
सिफूत	6.00	1,507.47	3.83	3,309.64	0	0	0	0	0	0	0	1	1	1	2	2	5	2	3	2	5

Figure 1.1: A sample from Soodkosh, a dataset of 90 Hindi Pseudowords

Pseudoword generation or datasets of such stimuli is thus a necessity. There is work for the same in major languages like Dutch, German, Spanish [7], Polish [8], French [9] & [10], and English ([11] and others). There is also some scarce work in other languages like Vietnamese, Basque, [7], Arabic [12],

and Korean [13]. However, for conducting tasks like Aphasia tests across languages, we need these systems/datasets to cater to a wide range of languages which are different typologically (e.g. Indo-Aryan) or otherwise (e.g. shallow orthography-phonology). Thus the first chapter proposes a solution to this problem by setting up a generation pipeline that is not tied to a specific language (as it models on just IPA-like phoneme sequences of any language's lexicon it is run on). We show how a run of the pipeline on English compares to other methods using some metrics established by [1] and then the results on Hindi which are evaluated by humans and existing methods alike. These pseudowords are then compiled with their evaluation statistics and other pseudo-lexical statistics as features to them, into a publicly available dataset named *Soodkosh*[1].

The second chapter then proposes an extensible and a human-in-the-loop evaluation framework for pseudoword quality. There is work in evaluating phonological understanding of language models and there is work in exploring generation of pseudowords. However, these works do not evaluate pseudowords for quality. Additionally, neural models of phonology or neural pseudoword generation models (like the one proposed in the first chapter) could be language-agnostic and thus be very useful in working with pseudowords in many languages. Although as they are neural models, they abstract away explainability [14]. Thus in terms of what kind of pseudowords a model is able to generate or if the model is sensitive to particular languages, quality evaluation strategies can help users understand these neural modelling blackboxes in more linguistic detail.

On the topic of why researching on pseudowords is important, the third chapter showcases how pseudowords can be useful in an entirely different research area i.e. understanding people's metacognition in terms of their perception of passphrases' memorability versus security. Here they are crucial to show the transition in perceived memorabilty as a factor of wordlikeness from non-sensical sequences to meaningful lexical items being used as passwords. The chapter then also shows how identifying pseudowords (which make text incoherent for popular NLP models) and replacing them in an informed manner could change the results of NLP tasks, as demonstrated in the field of Aphasia, where a lot of the naturally uttered sentences by an Aphasic person could contain neologisms, slips of tongue etc. all being inferred as phonology abiding but out of vocabulary items by popular NLP pipelines.

1.2 Motivation

This thesis is a result of a series of exciting explorations that started from me being fortunately guided into the Cognitive Sciences Lab as a cross-disciplinary opportunity by our Computational Linguistics degree. Clinical Neurologists in India had been looking to collaborate with the above mentioned lab and work on developing resources for Aphasia in Indian Languages. There is a huge list of such resources that need to be created (and not translated) to make an authentic and usable Indian-language Aphasia Battery of tests to understand the type and extent of Aphasia.

[1] *sood* (with a ḍ) sounds like *pseud* in English and is also a pseudoword in Hindi. While *kosh* means a collection in Hindi.

This thesis concentrates on one of the requirements, which is to develop Pseudowords as a part of the Lexical Decision Task that is helpful in judging an Aphasic person's lexical understanding of the language, vocabulary still retained etc. It was intriguing to me that pseudowords are unique items in the Aphasia Batteries that cannot be translated as they do not have a meaning attached to them! Add the fact that there were no existing methods for generating pseudowords in languages like Hindi and mix a bit of my bias for Indian languages in general, and you would arrive at the motivation behind this thesis!

1.3 Thesis Layout

C1 This is the introductory chapter, which discusses the scope of the work carried out in this thesis in the context of pseduowords, addresses the problems we are attempting to pose solutions to, and adds some motivation for the methods that we will develop in the following chapters.

C2 This chapter details the proposed pseudowords generation pipeline, runs basic checks for validity, and introduces the problems to be tackled with, when working with pseudowords.

C3 The thesis then explains the evaluation methodology proposed with new metrics, existing ones, and how to expand on the same.

C4 This penultimate chapter shows how pseudowords are important outside of Lexical Decision Tasks taking two separate studies with pseudowords as an integral part of them.

C5 Finally the thesis concludes with a summary of methods and results discussed in this thesis, the practical use-cases of these methods, and the scope of extension of this work in the future.

1.4 Applications of our work to science

The work in this thesis which is made available for research freely, aims to help the fields of psycholinguistics directly and majorly. It could also be another stepping stone in generating interest and more work in this field which clinical use-cases can benefit from. The applications then branch out to places where one needs a place-holder for experiments, a part of the stimuli or otherwise, or could be used to prod language models just like humans to find out more about its lexical cognition. On the same lines, it can also be used to understand a model's learnings from the phonology of a language and if that is also how humans understand phonology the same way.

Chapter 2

Pseudowords & How to Generatuce some

a

2.1 Introduction

The introductory chapter showed how pseudowords are crucial for diverse psycholinguistic tasks, clinical tests, and in NLP, e.g. language learning, predicting the kind & severity of aphasia, word sense disambiguation, etc.

Pseudowords were discussed in a psycholingusitic light by Greenberg in 1964 [15] and this was cited for a similar exploration in Hindi by Ohala in 1983 [16], however they have been majorly a part of a study relating to phonology of modelling of phonology since then. Even in the studies focused around pseudowords, there is very little research on the automated generation outside languages like English and French. This chapter highlights the state of the field, introduces a novel way to generate phonologically valid pseudowords, and the first attempt at creating a Hindi pseudoword dataset: Soodkosh.

As a summary the chapter highlights the current state of research around pseudowords and their generation. Then proposes a language-modeling-inspired yet language-agnostic technique to produce a dataset of 90 pseudowords in Hindi (as a sample language in need of this resource). We make the generation pipeline, and the dataset publicly available for academic research and use in clinical settings here[1].

2.2 Overview of research around Pseudowords

The research in pseudowords have majorly been about ways to generate it, from rule-based manners to using Markov chains (summarised in the next section). A lot of the generation research has been on resource rich languages, where the generator could exploit and be controlled for various psycholinguistic features recorded in public lexicons of these languages. Finally, some work is also around

[1] https://github.com/Abhinav271828/soodkosh-acl2023/

cross-lingual generation using the same paradigm. Note that unlike other linguistic resources used in the clinical settings, NLP, ASR systems etc. pseudowords cannot be borrowed directly/translated from other languages. This makes it all the more important to study and find ways which are cross-lingual / langauge-agnostic (here we mean that we propose a language agnostic pipeline to make language specific pseudowords. The pipeline should be usable by all languages to generate pseudowords according to their respective phonotactics.).

Another but smaller line of research involves mimicking evaluation of phonology modelling studies (researched extensively) to evaluate pseudoword quality as well. These evaluation strategies involve pseudowords in a Lexical Decision Task like setting but the aim has not been geared towards quality evaluation of how 'good' a pseudoword is. Greenberg et al. [15] carried out one of the earliest of these studies, followed by Ohala [16] conducted for Hindi. In these studies, they would generate a few pseudowords (deemed candidature by researcher heuristics) and place the outputs with words (and not nonwords) to setup a Lexical Decision Task for a small set of humans. Note that these are binary/ternary word-nonword decisions which give us only the information on the possibility of a token being a pseudoword and nothing else about its quality, of how wordlike it is etc. For the same a more gradient based judgement could be helpful (which is what we explore in the proposed evaluation strategy).

More recently, work in evaluation of pseudowords from existing popular datasets of pseudowords have also been analysed for the importance of shallow morphology in context of native speaker acceptability [2]. The next chapter on evaluation highlights such evaluation methods over the years and the ideas borrowed from each to set up a language-agnostic behavioral experiment for us to record human understanding of pseudoword quality.

2.3 Research in Pseudoword Generation

2.3.1 Generation methods

Pseudowords have been generated in multiple ways over the years (as detailed by [1]). Note that most methods need a validation step where they remove the generated forms if they appear in a dictionary of the language. This section thus provides a summary of the methods tried out so far and paves way for the pipeline the thesis proposes. The methods summarised below have been exclusively used to generate pseudowords (and not just a part of a study relating to phonotactic modelling):

- **Manipulation**: This is done by changing words to produce forms that don't exist in the dictionary. Done on a character level, manipulations involve adding, removing, editing, or transposing some characters in a word. [3] has one of the largest databases of this kind, where they don't describe the process of generating pseudowords (in their paper "nonwords", that are not *clearly* "nonwords") in not more than a line which is a paraphrase of the above. About 40K pseudowords were generated by this method in their work and some examples of manipulating a word (`bottle`) to obtain

pseudowords in English (`bottleb`, `botle`, `obttle`) are as follows: *Insertion* (`b`): `bottleb`, *Deletion* (`t`): `botle`, and *Transposition* (`b, o`): `obttle`.

- **Combination**: Popular tools in English like WordGen [11] and Wuggy [7], datasets like ARC [17] and The English Project [3], and similar resources in languages like French (The French Lexicon Project [10] and Lexique [9]) work by combining high-frequency n-grams. Generated by WordGen ([11]) and shown by [1], a pseudoword in English `reroin` can be generated by chaining `re`, `er`, `ro`, `oi`, and `in`. Respective frequencies of these phoneme bigrams are 4760, 7279, 2840, 468, and 7156 as calculated from [11]. This kind of high frequency is only one of the 7 ways to confirm a random string's possibility of being a pseudoword by [11]. For a string to be a possible pseudoword as per [11], it should not already be present in the lexicon and the following 7 constraints must be met (only for the few European language family's languages that it can operate in): number of letters, neighbourhood size, frequency, bigram frequency (initial, final, minimum and summated) and orthographic relatedness.

  Some of the other combination based methods include constraints on generation like minimum or aggregated frequency of the sub-syllabic units like existing onsets, nuclei, and codas. An example for this is: `scuf` (a pseudoword in English) formed from = `sc` (onset in ***sc****ar*) + `u` (nucleus in *b**u**n*) + `f` (coda in *ree**f***). Wuggy ([7]) is a method that uses this way of pseudoword generation, where all syllables are broken down into valid parts and are all listed out into a tree of onsets, nuclei, and codas, which is then traversed back to join different sub-syllabic elements in a valid way to generate new sequences in the end.

- **Prediction**: There is one method [1] which generates pseudowords (CGCA) by chaining character n-grams based on the probability of appearing within the language. This is done by a script first going through a lexicon of the language and building a dictionary of n-grams and their three respective scores (word-initial frequency, word-medial frequency, and word-final frequency). Then a frequent word-initial n-gram is picked from this dictionary and another n-gram with an overlapping n-1 gram is picked to continue forming the pseudoword until a word-final n-gram is reached. An example figure for the latter is shown below (Figure 2.1 from [1]) König et al. conduct the study majorly on English data and showed results for some other popular European languages as well (German, Spanish, Italian). They conclude with the caveat that the system still needs further testing to claim language-agnostic-ness of the system. This is as the orthography of all the tested languages were majorly the same, as an example, we don't know if the system works well for abugidas like Hindi too.

2.3.2 Other work where Pseudowords are generated

Dautriche et al. [18] use n-gram phonological modelling in combination with Probabilistic Context Free Grammar over broader units like syllables and words to generate phonotactically valid sequences (note that these are not neural network based approaches). However this approach has been used by

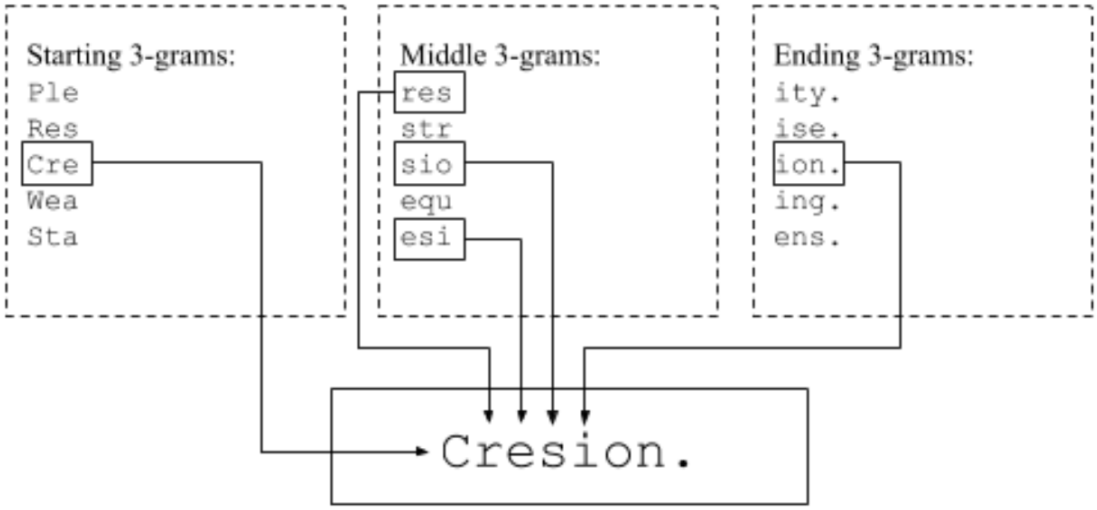


Figure 2.1: An example of a pseudoword being formed by the CGCA algorithm adapted from [1]

the same work, Trott et al. ([19] & [20]), and Caplan et al. [21] as a part of bigger experiments in liguistic research like homophony, or cognitive science research like the concept of Miller's monkey in phonology. For example in this study of Miller's monkey in phonology the researchers set out to prove that a natural lexicon made of randomly generated sequence of phonemes considering some constraints in phonology and semantics, is comparable in communicative efficiency of a lexicon that is made from actual words in context of a language. Thus this study used a generation methodology to (a) generate phonologically valid sequences (that could be anything) & (b) not for the aim of generating quality pseudowords to be of use in psycholinguistics, but for a larger aim of talking about Miller's monkey in a lexicography, linguistics, and cognitive science aspect.

2.3.3 Some problems with the current approaches and the way ahead

Manipulations and combinations generally require expert knowledge of the language to filter out the results which are not allowed by the sequential phonological constraints in a language [16] (apart from the automated constraints). Additionally, [14] find that n-gram models are dependent on fixed/specified context windows, which can cause the modelling to lose out on numerous long-distance dependencies. Finally (as discussed in the introduction), we also need generation methods which are multi-lingual or language-agnostic in nature. Following these requirements, we reviewed literature on neural models of a language's phonotactics (and how they compare to n-gram based modelling), as such a model could approximate the expert knowledge and if abstract enough, it could be used to generate phonotactically valid forms across languages. The review resulted in two parallel lines of relevant work in recent times.

One line of work led us to Trott et al. ([19]), who in their more recent work, noted that RNN based modelling could lead to comparable results, citing Pimentel et al.'s research on Homophony and Rényi

Entropy [22]. Here they use a similar architecture as ours (all works derived from the Pimentel work, including Trott's, use an LSTM based modelling of a IPA like lexicon to study different aspects of phonetics and phonology), to yet again model phonotactics of a language and not to generate pseudowords exclsuively. In the latter study, Pimentel et al. directly compare their results to n-gram based models and show how n-gram based modelling outputs misleading results. Finally, Futrell et al. [23] explore how the latent space of phonotactics is better represented by a model which can uncover phonological features and the hierarchy of them. Their modelling approach shows that n-gram based approaches definitely loses out on these underlying feature representations.

Another line of work built on the recent research by [24], [14], [25] and others who have used neural language models (LMs) like LSTMs and RNNs to probe such models' capability of understanding phonology in various ways like if distinctive features are obligatory for phonotactic learning, if character-level embeddings encode sound patterns etc. [14] also report that for Finnish, a language with vowel backness harmony in roots and in affixing, n-gram models may not be able to detect disharmonious but spread out vowel subsequences as well as neural models can. They find that RNN-based LMs correlate with human judgements on scores of attested (and marginal) forms.

To summarize, on one end, non-LSTM-based methods are preferred for their better interpretability. Additionally, their variables can be better controlled for generating pseudowords. On the other hand, LSTM-based methods have been shown to be better capable at generating substantially more pseudowords due to the underlying continuous sequence representations. They can also generate a variety of pseudowords creatively (as seen in the following sections).

Following these trends and findings in computational phonology studies, we see that statistical methods lose out on long range dependencies ([14]), while RNN-based ones correlate better with human judgement ([14] & [19]) while also understanding phonological features ([25]) which are important ([23] and are missed out on by a statistical n-gram based approach). Apart from this, we also believe that it would be easier to expand a pseudoword generation pipeline cross-lingually if the way to generate is more RNN-like than n-gram-like, as shown by other domains and tasks in NLP. Thus we design an LSTM-based pipeline to generate pseudowords in a language (like PhonRNN [24]) as described in the next section.

2.4 The PseudRNN Generation Pipeline

We essentially treat this task as language modelling (or more generally, next-token prediction) on a character level. The pipeline (PseudRNN) for the same, details the training data, the model, and the post-processing required. The pipeline requires an IPA-like lexicon as an input, with each lexicon item transcribed using a uniform phonetic transcription system with space separated phonemic tokens. This ensures that the model can learn a language's phonology from a phonologically valid set of words. The output of this model is intended to be a randonmly initialised, generated set of phoneme sequences. We finally subtract words from this set to obtain candidate pseudowords (as the model has only trained on

phonologically valid sequences, the output is expected to contain only phonologically valid sequences as well i.e. words & pseudowords.).

2.4.1 Data

Earlier works utilize the orthographic representation of a language's lexicon for generating pseudowords. For Hindi specifically, [26] enlists various issues with the orthographical representation like schwa-deletion not being fully reflected. Moreover, a cross-lingual analysis of character-level language models by [25] shows that textual character representations correlate strongly with sound representations for alphabetic orthographies. In this case therefore, it could be more inclusive to use sound representations rather than adapting to different orthographies, more so as the pipeline is intended for use across multiple languages. Finally, we ran this pipeline for unconverted orthographical, space-separated version of the Hindi input lexicon (Section 2.5.2.1) as well. The results showed us systematic errors that this modelling resulted in, which phonologically converted input did not show.

Thus like [27], [24], and [25], we intend to map orthography to a phonologically universal but distinctive feature space. A good approximation of which is WikiPron's (licensed under Apache 2.0) [28] grapheme-to-phoneme aligned, pronunciation datasets. Note that our work aligns with the wishes of WikiPron's authors i.e. the software be used for building and evaluating speech technologies for less-resourced languages. We scrape 12,608 words from Hindi Wiktionary using WikiPron and split it in an 80-10-10 fashion for the train, test, and validation sets. The data (in line with the above requirement) is composed of Wiktionary (licensed under CC-BY-SA) [29] transcriptions of Hindi words with syllable boundaries marked by a period symbol. While for English we used about 70K entries in the same format.

2.4.2 Model

We adapt our model pipeline using the PyTorch WordRNN implementation [30]. We begin by encoding all tokens in the vocabulary (phonemes, syllable bounds, etc) as unique one-hot-vectors. Additionally, during training, input sequences are appended with a special end-of-sequence token (<eos>). The model architecture starts with a fully-connected encoder layer that is used to learn corresponding embeddings for each input token. A subsequent decoder consisting of two layers of unidirectional LSTMs [31] follows. Finally, a Log-Softmax operation is applied to the output embeddings from the decoder to obtain the probabilities for the next most-likely token, which is compared against the expected output using a negative log-likelihood (NLL) loss. Hence, the loss function is used to evaluate and update the model upon its prediction of the next phoneme in the phone sequence given the past context fetched from the transcripts of valid words. A dropout of 0.2 was applied to all learnable layers in the model.

To generate pseudowords, we initiate an input sequence with the learned representation of a randomly selected token from the vocabulary. The model is then allowed to iteratively extend the sequence with

the most probable next token up to a maximum sequence length. The next-token generation pipeline is summarized in Figure 2.2.

All experiments were trained using an SGD optimizer with an initial learning rate of 20 for 40 epochs.

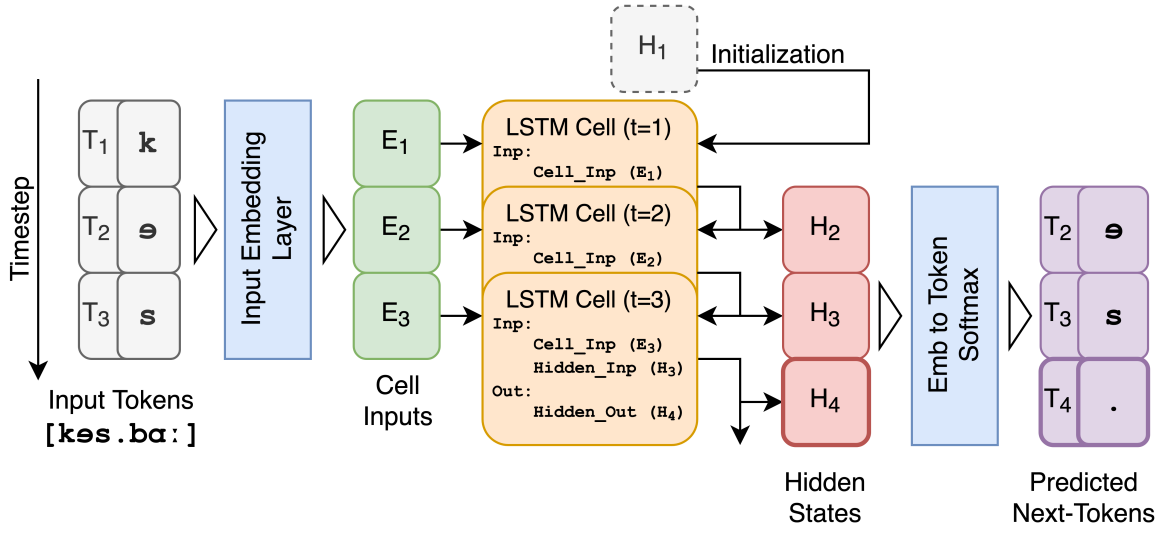


Figure 2.2: PseudRNN: Internal LSTM Architecture. At each timestep, the input embedding of the input token and the hidden state from the previous timestep is used by the LSTM cell to generate the output state of the current timestep. This is then used to predict the most likely next-token in the sequence.

2.4.3 Hyperparameter Search

Since there is no current automatic evaluation to find a neural model's ability to generate high-quality pseudowords, we observed the number of valid words generated by each model for hyperparameter tuning.

The most crucial hyperparameters we found to affect the model outputs are as follows. Note that all other model hyperparameters were not changed from their origin WordRNN source (as they did not hamper the results and were mostly consistent with PhonRNN too [24]) and were set to default values and have been listed in Table A.1 at Appendix A:

- **Input Embedding Size:** The input embedding size of the LSTM refers to the size of the input tensor representations assigned to each token in the input vocabulary. A larger embedding size is attributed to a greater bandwidth of input features encoded for each token, however, this size must be limited according to an appropriate quantity of training data. Too large an embedding size can lead to sparsity, with poorly trained dimensions leading to lower performance.
- **Layer Count:** Stacking layers of LSTMs (i.e. passing the cell & hidden outputs of an LSTM to another LSTM as inputs) helps encode higher-order features of the input sequences. How-

ever, similaar to input embedding size, too many layers with insufficient training data can lead to sparsity and lower performance.

- **Input Sequence Length:** The input sequence length defines the number of past tokens visible to the model for next-word prediction as context. A larger input sequence length allows for a better access (not utilization) to the past context, which can help encode long-term dependencies better. However, the optimal utilization of this context depends on the other corresponding properties of the LSTM model (i.e. input and hidden dimensionality, layer count, activation function etc).

Our findings are as follows:

- The number of real words generated increased when the embedding dimension was increased from 17 to 34. Thereafter, it plateaued, finally dropping again at 68.
- Increasing the number of layers beyond 2 resulted in poor syllabification (e.g. empty syllables were generated), except in the case where there were 68 input features. However, since this decreased the total words generated, this combination was not chosen.
- Chunking the input data into sequences of 10 also led to the highest number of words being generated. Decreasing this to 5 or increasing it (up to 40) resulted in relatively poor performance by this metric.

We finally opted for an encoder with an input embedding size of 34 and a decoder with two layers of LSTMs, with an input sequence length of 10.

2.4.4 Post-Processing

Using the above model, we obtained about 200 candidates for pseudowords. We then converted it back to orthographical form in Devanagari as this form is easier to look up in lexicons/search engines to find out if it is already a word or not and it is also easier to show as a stimuli to gather Wordlikeness ratings for (next chapter), and it is finally the form that would be required by an Aphasia battery or an NLP task etc.

We did this by asking two separate Hindi native speakers who were comfortable with IPA as well, to convert IPA to Devanagari. We then calculated inter-annotater agreement by using sequence overlap (intersection over union) style metrics, where similarity/agreement of the two annotators are a function of their Edit Distance, Jaro-Wrinkler Distance etc. Table 2.1 shows the agreement derived from various such metrics as defined in these footnotes[2], [3].

[2]the similarity for each of these metrics was calculated by inverting the distance metric and factoring in the length if the metric already did not take that into account.

[3]Manual: we used `0.5` as the distance in case of each "matraa" change and `1` otherwise; Levenshtein: distance between two words is the minimum number of single-character edits (insertions, deletions or substitutions) required to change one word into the other; Jaro-Wrinkler: designed and best suited for short strings such as person names, and to detect typos; `n`Gram: as defined by [32]; SD: similar to Jaccard index, the similarity is computed as $2 \cdot |V_1 \cap V_2| \,/\, |V_1 + V_2|$; OC: computed as $2 \cdot |V_1 \cap V_2| \,/ \min(V_1, V_2)$, and TER: adapted directly from the implementation of this work: [33]

Agreement Type	Mean	SD
Manual	0.97	0.04
Levenshtein	0.94	0.07
Jaro-Winkler	0.97	0.04
BiGram	0.99	0.02
TriGram	0.99	0.02
Sorensen-Dice (SD)	0.73	0.32
Overlap Coefficient (OC)	0.75	0.30
Translation Error Rate (TER)	0.99	0.02

Table 2.1: Summary of inter-annotator agreement across various metrics to highlight that there was reliable agreement on phonology to orthography conversion.

We see that all of these metrics show more than 73% agreement between the annotators, showing high agreement between them. This is because for Hindi phoneme-letter mapping is highly one-to-one. The places of difference were majorly the same across the annotations, where one annotator did not strictly follow the syllable boundaries as indicated by the PseudRNN outputs and another did (e.g. for the IPA transcription ˈəz.zəmˈ , we had two annotations, one which considered the first z as a consonant on its own, while the second annotation attached it to the second z).

We then removed any duplicates and then filtered away any sequence that was found in the Wiktionary lexicon, reported as an exact word match, or returned more than 5 results on an internet search. This resulted in a final set of 90 candidate pseudowords to be evaluated for quality.

2.5 Results & Additional Experiments

2.5.1 Main Experiment

Below is a preview of the main experiment pipeline (Figure 2.3) and the list of 90 pseudoword candidates that were finally generated (Figure 2.4):

2.5.2 Other Experiments

We also ran the pipeline separately on Hindi unconverted orthography lexicon to check if conversion to a phonologically oriented space was a required step, and an English IPA transcribed lexicon to compare our model against the ones in the past using existing metrics.

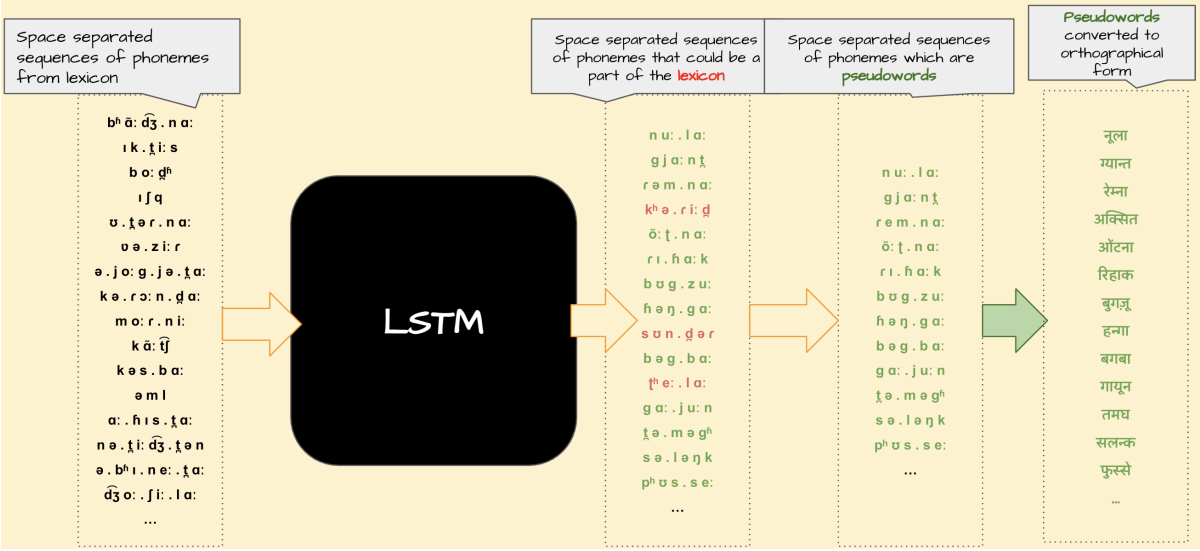


Figure 2.3: PseudRNN Pipeline

स्घोट, तजरील, पनिवैश, बामल्लें, अक्सित, हरखिनित्व, तामबूलियों, तॉल, तमघ, अनाकषत, बेशिध्र, परिमयता, कव, सिफूत, काराईट, अज़्ज़म, ननदान्स, रुदनीवाला, कोरुपाशवादी, इलाहना, ज़सूची, यर्धबाद, शीस्करण, केलिफोन, मुनमूल्य, बिशुभावल, नसविष्का, सफेरी, बुगज़ू, फुस्से, उटकाना, इन्स्टारा, असनस्पेतक, उत्गारेषियों, हन्गा, ताजपुन, युतको, पताग्रा, धुरन्वै, पदन्शानिया, मुर्तववादी, हफ़ल, सलन्क, अनातक, उन्जुका, नूला, फैखी, जफजा, औरावस्था, जांख, भीर्ड, वाविमा, शद्रगानी, मायामामन्द, रेम्ना, देहमूनिका, श्रिहागनतन, क्राब्लोनेविक, परीथाम, ग्यान्त, विका, अन्टा, आल्गा, लालीस, किरनोवक, वीशकाला, राइकरषित्क्ता, ना:, ओंटना, अपरचन, गुठ्ठानेबाद, आत्मखाई, दुस्सवपार्शिक, हनड़, जुल्क, लिठाई, धोगिमन, कुय, नायू, रिहाक, अबख़दा, दुमार्डा, प्रात्रिषक्षत, साथ्यदारी, गायून, किन्हूता, फ़िल्र, बेन्सीना, पेन्श, and मसरह .

Figure 2.4: The results of the generation pipeline

2.5.2.1 List of Pseudoword candidates generated from unconverted orthographical modelling

Since orthographical embedding space etc. could be different from the phonologically converted ones, we tried tuning the hyperparameters (Section 2.4.3) and found that the common pattern based mistakes were similar across the runs, thus we report the results from the model with the same settings as the main experiment above in Figure 2.5.

2.5.2.2 List of Pseudoword candidates generated in English

We also ran English data through the same hyperparameter settings as the main experiment done on the Hindi data. This was done to make the methods and outputs comparable. (Figure 2.6)

ा, खअज, ्सकी, बबब्, िसमफा, झऑज, ढ़बी, ड़ा, ध्योक, ाहली, ड्धो, ्कल्य, ड्जव, पल्इग, ीग, िक, ़मोम, ्ह, ग्ढाम

(a) Nonwords generated by unconverted Orthographic Generation

अलखलित, तमकी, नजुशुओ, चनवाह, करदाब, अथद, हंडवर, मटणो, जूफ, गिघा, कलज, अपता, रक़ब, माख, बक़बा, शमल, गौढ़ी, सबाह, वैजा, अंगीह, ग्यार, अटालि, सलकना, मौगे, इदक, खषैली, सुघर, मुक़, मूतान, उहबाद, भ्वशरिक, तगम, बन्बाहा, ज़ारप, गोक, संट, उओंगी, दुत्सम, अल्गन, पसाफ़, सौँज, थोख, पूटो, खकुद, मुनमु, ईगर, चोदनी, भकटी, डूग, क़राना, हिलताक, निंच, निव, भितर्बा, पौंचान, भोसटा, निष्थि, तलब्ध, नन, मूताल, द्राट, सस्नु, कूम, गुर्जन, छरग, बिकुक, खुखन्दू, नूब, रयुज, पुलू, सतेप, सरबलरा, खद्वान, दक्षा, सेहेसरी, किनित, इप्यधान, मृना, इयाबनी, तुँडनी, इनावचाज़, उसचात, इपसन, अपम, थूर, कृशल, अवस, नसम्ज्ञन, इतक, अहशवा, अमँभा, युघी, चुद्हक, बुहेल, होक, इम्मी, धूर्श, भूज, होंव, शेघा, शावाइक, कस्पलन, ईपा, पिंद्र, नहती, खखू, चइदर, पुतकवाच, प्रेवाज, विधक़र, and भेतई

(b) Pseudowords generated by unconverted Orthographic Generation

समा, नंगी, ला, वेप, लैस, सहस, असंख, बेड़ी, टिन, गिन, विश, खती, पायर, कोर, कँडा, गुँगा, टना, वहस, मगी, बहनी

(c) Words generated by unconverted Orthographic Generation

Figure 2.5: Manually divided outputs from a run of PseudRNN on a lexicon of Hindi words written in Devanagari

alagoma, alanbalation, alchuinitively, angapathermony, abilega, bodthlow, balute, beeicole, broocco, dalvation, demban, demiona, decliriate, dilivva, disloces, dunous, joquila, jalasala, gelveralogist, akima, edecency, explacating, embikken, aesterway, feracekers, phitodiscloying, finiver, fratical, fraimnation, ganenbia, gogilative, gaistproof, glockstone, greenwriter, hibertons, hernic, isha, ifball, intertenenities, kalelingking, cajaver, legaword, lethsightedly, ludle, macelousness, magnilint, meracrodema, midfrowned, nacrolosis, neeve, overbondery, pacidic, pimafactive, pimafactive, ranigal, reoginable, reoculated, rebird, reformentness, sombatomic, cytolization, sidewail, sigment, simbluiks, symphorously, scapeswerk, stadient, stafezing, switchbeck, taguling, trilogenase, chelidiness, vanentative, vegulytic, visures, unumbying, uncodicepous, and wasatlav

Figure 2.6: The results of the generation pipeline on English lexicon

2.6 Discussion & Making way for Evaluation

We see that PseudRNN generates about 45-50% candidate Hindi pseudowords out of all the generated strings in one run (Fig. 2.4). This is as the total strings output were around 200, out of which about 90 were not found in a dictionary or on Google. For Hindi, there were about 10 sequences generated were nonwords, as these had multiple syllabifiers placed adjacently, or were breaking a very clear Hindi phonological rule. For a comparison, the English data generated about 80% pseudowords. Out of the rest, 15% were found to be in the input lexicon and 5% were found as rare English words on Google. This shows us that differing data source sizes across languages can affect the pseudoword production efficiency in the pipeline.

Detailed statistics on phonological length, orthographical length, closest words etc. can be found at the end of the next chapter on evaluation, where we establish these metrics and report them. Although, we can already note here that the outputs from the phoneme sequence modelling contained only phonologically possible forms, out of which about 50% were actual words. This is consistent with English,

where the outputs are all phonologically legal, but the pseudoword-word ratio is higher as we got about 80% pseudowords and the rest as actual words in a dictionary. Thus, pre-evaluation, we can gauge that the modelling on phoneme sequences by PseudRNN (with these hyperparameters) is at least generating legal sequences as well.

On the other hand modelling without phonological conversion (Fig. 2.5) have about 20% (about 18-25% across different hyperparameter tests) strings consistently breaking Hindi word formation rules, like:

1. Matraa Positioning: Hindi vowel markers (matraas) cannot occur at the very beginning of a word-form. However the model does output multiple strings that have a standalone matraa in the beginning. We hypothesis that the cause of this is how some matraas are stylised such that when breaking down a word, they would appear at the start. Similarly, vowels themselves cannot occur in the middle of a word without converting into a marker unless on a syllable boundary. However the figure shows multiple places where it is not possible to pronounce a string because of a vowel unattached to a consonant, occurring in the middle.

2. Standalone Matraa: Sometimes the model has also output strings where the vowel marker was unattached to a consonant and has a placeholder instead. These are also not pronounceable.

3. Other Phonotactics: In Hindi, some consonants only occur in vowel environments, i.e. a vowel(-like) sound before and after are needed for these consonants to occur. However the sample non-word outputs show that this is a recurring mistake the model makes. Another observation is that the "nuqta" sign (generally indicates sounds not present in the script originally, but has come through via borrowing etc.) that is used with consonants for which we don't have such a variation for. However PseudRNN modelling orthography directly, showed that it liked to use nuqta with a lot of other consonants too. This made novel sound sequences that were not pseudowords strictly, as we did not know how to pronounce them.

For English sequences, it was trickier to convert IPA sequences to orthography (although it was straight-forward in the cases where we could find existing roots on comparison with the lexicon). This is because (at least in American English) as a study shows, English is not phonologically transparent [34] where as a part of one of the experiments, they try "back-translating" English spelling from IPA and could not do so as unambiguously as in a language like Finnish. As an example, this can also be seen in the case of schwa, where it can be translated to multiple vowels in the same environment in English. For the scope of this thesis, we find the closest IPA sequence in the lexicon to ensure maximal usage of existing mappings and then use bilingual speakers' intuition to find rest of the orthographical representation of pseudowords that we generated to be able to compare with existing metrics that only take such forms into account. The resulting Limitations are discussed below.

2.7 Limitations & Future Work

A major limitation of the pipeline is the availability of a native speaker well-versed with IPA (or IPA-like transcription system) and the target language's degree of orthography-phonology correspondence. For the latter, there is no absolutely unambiguous language, but as seen above, Hindi, Finnish, and other languages come close. Thus we could use the existing lexicon's closest matches in IPA sequences as a helping hand for languages like English, until a better alternative can be found.

Secondly, the generation pipeline could be improved by experimenting with hyperparameters to make a more language-agnostic model that needs little tuning to adapt to another language or testing out models like Bi-LSTM ([25] show right-to-left information improves phonotactic understanding). We could tinker with affixes, roots, or suprasegmental features like tones, allophones ([35] & [36]) etc. along with the word input to these models for better linguistic cohesiveness. Finally, the dataset can be made more usable by adding more psycholinguistic features like concreteness, imageability etc. Or by adding new linguistic information like possible part-of-speech judgements.

Finally, upon discussing with reviewers, we realised that pseudowords which visually look like the form of an existing word are also used by Lexical Decision Tasks and could be an interesting area to explore and model for, however that is presently out of scope of this pipeline.

Chapter 3

Evaluadating Pseudowords

3.1 Introduction

In this chapter, as a next step, we propose an inspired behavioural experiment design to evaluate pseudowords for their "wordlikeness" against native speakers' linguistic intuitions. The results from this showed that native speakers confirm the generation model's ability to capture the language's phonology and the pipeline's ability to generate and evaluate a variety of wordlike pseudowords. This design helps us record, analyse, and propose Wordlikeness Rating as a metric to evaluate pseudowords apart from 3 other materics based on Familiarity and response times that this chapter will detail. We talk about these 4 related metrics as *Pseudoword Acceptability Metrics* in this chapter.

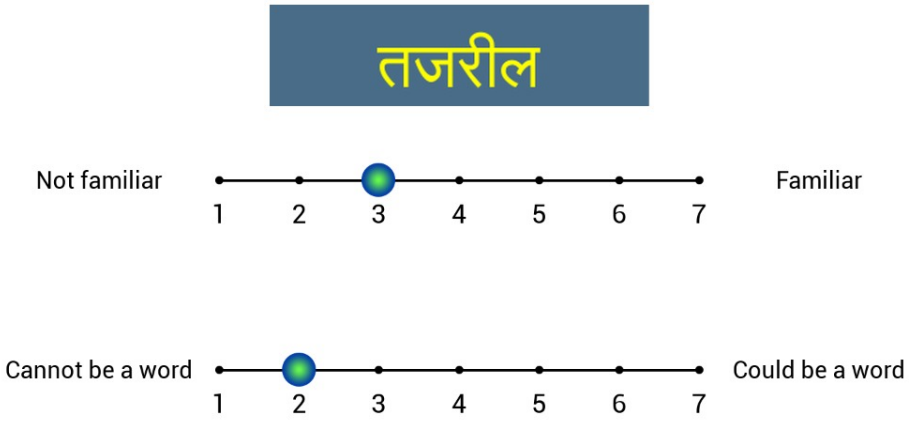


Figure 3.1: Preview of the experiment developed to evaluate a generated pseudoword in Hindi

Secondly, we present Soodkosh, a dataset of 90 Hindi pseudowords complete with various metrics of pseudoword quality and psycholinguistic features applicable to such strings. We explain these features and how they could be useful in various use cases as well.

We thus make the evaluation experiment and the dataset publicly available for academic research and use in clinical settings here[1].

3.1.1 Related work

There is scarce work in pseudoword evaluation, as it is to be checked for quality against a language's phonology, which is not cut and dry. Although pseudowords have been evaluated for their phonological validity in a few language-specific cases, carried out by comparing them to existing word-forms in various ways. Research by König et al. [1] extends this by proposing two metrics to check for the generated pseudowords' orthographic legality and task suitability in English. This is the most that has been done, outside of studies aiming to understanding other phonological aspects which involve pseudowords as a part of the study, as described below.

Like phonology modelling studies for generation, evaluation of phonology models have been researched extensively (these do not evaluate pseudowords). These evaluation strategies involve pseudowords in a Lexical Decision Task like setting but the aim is not geared towards quality evaluation of them. In terms of evaluating pseudoword quality, [15] carried out one of the earliest of these studies, [16] conducted this for Hindi, and more recently, [2] where they analysed the importance of shallow morphology for native speaker's acceptability of a wordform.

To specifically establish an evaluation strategy for pseudowords, we follow the idea of Gradient Acceptability laid out by [37]. We use a similar paradigm to the evaluation strategies by [37], [15], and [16] and design a behavioral experiment to gather native speaker intuition of pseudowords' wordlikeness measured primarily by their wordlikeness rating, but accompanied by their familiarity with the pseudoword and their response time. We aslo incorporate [1]'s metrics among others to propose more ways to evaluate artificially generated pseudowords in a language.

Similarly, there has been no work on evaluation, or dataset creation of Hindi Pseudowords. Thus we use our pipeline to generate English Pseudowords and compare them against existing metrics to position our pipeline among current methods by using the existing evaluation for English, apart from showing how the metrics we propose can be used in evaluating Hindi pseudowords and establish a dataset.

Finally, the evaluation pipeline we propose is a concoction of parts of monolingual pseudoword evaluation methods in the past, thus we mention the rest of the related work in the Design Choices subsection (3.3.4).

metrics by [1] ->	**Legality**				**Suitability**				
pseudowords' source	**C+**	**V+**	**CV+C**	**C+V+C+**	**1-C**	**comp**	**poly**	**npoly**	**half-real**
CGCA [1]	0.01	0	0.01	N/A	0.24	0.04	0.45	0.27	N/A
ARC [17]	0.03	0.01	0.14	N/A	0.34	0.01	0.01	0.58	N/A
ELP [3]	0.04	0.02	0.06	N/A	0.43	0.06	0.03	0.58	N/A
WordGen [11]	0.08	0.05	0.18	N/A	0.48	0.01	0.08	0.36	N/A
Wuggy [7]	0.03	0.01	0.19	N/A	0.47	0.03	0.07	0.37	N/A
Meara [38]	0	0	0.06	N/A	0.14	0.09	0.1	0.26	N/A
PseudRNN (Hindi)	0.04	0.01	0.14	0.68	0.27	0.02	0.06	0.32	0.12
PseudRNN (English)	0.77	0.33	0.99	0.99	0.31	0.12	0.17	0.46	0.21

Table 3.1: Comparing *legality* & *suitability* across different methods of generation (*legality*: error percents, *suitability*: count percents. Rounded off to 2 decimal places). Note that CGCA values are the mean of 8 reported models. The N/A marks the fact that the features `C+V+C+` and `half-real` are introduced by this work and have not been tested out by the authors of any previous research in the table.

3.2 Existing Metrics

In this section, we compare the generated pseudowords against metrics proposed by [1] for their *legality* and *suitability*. As this is the only existing work on evaluating generated pseudowords (not on quality, but on specific features i.e. legality & suitability), we explore these and understand what else can be done.

Starting off with *legality*, according to the paper a pseudoword can be considered "suspect" if it has one of the following error types (first 3):

1. `C+`: sequences of consecutive consonants from a pseudoword that don't exist in the lexicon.
2. `V+`: sequences of consecutive vowels from a pseudoword that don't exist in the lexicon.
3. `CV+C`: sequences of consecutive vowels including one leading and one trailing consonant from a pseudoword that don't exist in the lexicon.
4. `C+V+C+`: sequences of consecutive vowels including consecutive (one/more) leading consonants and consecutive (one/more) trailing consonants from a pseudoword that don't exist in the lexicon.

[1] `https://github.com/Abhinav271828/soodkosh-acl2023/`

Note that these are binary metrics i.e. if a sequence has more than 1 C+ errors, it would be counted as 1. For more details on the first three legality metrics, please refer to [1]. We have added the check for C+V+C+ sequences as well to encourage more research into legality as the first three are not exhaustive and one needs to account for more valid sequential productions that are legal across languages. As an example, the C+V+C+ sequence occurs in **str**ing (English) and **str**ee (Hindi). On calculating the errors for Soodkosh, we get a 0.68 error ratio. While for English the errors were higher (0.99) where almost all the words broke this new proposed rule.

Table 3.1 (adapted from reported scores by [1]) compares Soodkosh (the 90 Hindi pseudowords we generate) and PseudRNN generated English pseudowords (section 2.5.2.2) against various English pseudoword sources. Note that PseudRNN generated (Soodkosh/English) pseudowords' legality was rated by an automated script (available on the repository) by us, following the definitions in [1].

To assess the *suitability*, [1] defines (first 4):

1. 1-C: *"One-character dissimilarity"*, count of pseudowords that are one character away from a word in the lexicon.

2. npoly: *"n-polymorphic"*, number of pseudowords formed from a non-existent root but a real affix.

3. poly: *"polymorphic"*, number of pseudowords formed from a real root & a real affix.

4. comp: *"compound"*, number of pseudowords formed by two words in the lexicon.

5. half-real: *"half-real"*, number of pseudowords formed from a real and a non-existent root/affix.

For more details on the first four suitability metrics, please refer to [1]. Similar to legality, the comparison for suitability can be found at Table 3.1)[2]

Similar to our C+V+C+ proposal, we propose half-real as we found that Soodkosh has 12% of these while PseudRNN generated English pseudowords had about 21%. We encourage future work to come up with more lexical metrics similar to these (section 3.4 follows for more motivation from the observations from generated pseudowords).

In conclusion, we observe that these metrics can be expanded upon and are not a judge of a pseudoword's quality but some features. WE see that one can employ experts to determine on the basis of morphology and semantics, how suitable a pseudoword is and how PseudRNN generated English pseudowords are more suitable in certain tasks and the Hindi productions were comparable to other English methods. We also find that the English productions don't seem legal, while they are generated from the same pipeline as the one that generated the comparabale Hindi pseudowords. These English

[2] 2 independent Hindi native speakers rated, following [1]. Inter-rater agreement was above 90% (Cohen's kappa >0.8) in most cases, except for comp. In case of dispute, a third expert was asked to choose. For English PseudRNN results, most agreements like between Cohen's kappa 60% and 75% except for 1-Character dissimilarity which is 29%

pseudowords were also found suitable in various cases by annotators which is only possible if they were legal (A more detailed discussion can be found at Section 3.4.1.2). Thus we propose a set of metrics to directly evaluate a pseudoword's quality in terms of how wordlike it is and describe how to gather & interpret them (Section 3.3).

3.3 Pseudoword Acceptability Metrics

3.3.1 Introduction

Acceptability judgments have been a way to evaluate novel word forms (pseudowords here) based on well-formedness constraints or the implicit phonotactic grammar (unlike school-taught syntactic grammar) of a native speaker, in formal linguistic research [39]. Additionally, [35] reports that in the past fifty years, numerous studies suggest that phonotactic knowledge is gradient and that the effects of the factors from this grammar cannot be reduced to lexical statistics. We thus propose to find out *Pseudoword Acceptability* (on the basis of gradient acceptability) as a metric to check how wordlike a pseudoword presented to a native speaker is and as a result a way to evaluate pseudoword generation methods as well.

However, as [37] mentions, an important check of phonotactic modelling is human judgement, although not the only one as we still need finer ways [40] . Thus, after proposing the design and evaluation strategies on this metric, we expand it to other proposed metrics like suitability, legality (as seen above), and against basic psycholinguistic features. Detailed below is the experiment design to collect gradient acceptability judgements for pseudowords (in Hindi).

In a nutshell, we present a behavioral experiment design that gathers the Wordlikeness ratings and related dependent variables (Familiarity ratings, and response times for both ratings) forming a set of 4 metrics composing the Pseudoword Acceptability Metrics that we propose, in a language agnostic fashion (with Hindi as a low-resource example to begin with). A major change from the work before (like [37] or [16]) is the fact that we include non-words as a part of the experiment to establish a baseline, and re-purpose the aim of the experiment to judge how close a generated candidate is to a word, simultaneously how far it is from a non-word rating, and how to compare & gauge these in a standard manner so that other evaluation studies can replicate this and poke into the generation quality better.

3.3.2 Experiment Setup

The experiment was designed in Neuropsydia [41] and requires participants to rate stimuli for their familiarity and wordlikeness on a 7-point scale. After obtaining detailed consent (Section B.1 at Appendix B) and demographic details like age, gender, and language proficiency in Hindi, the participants were instructed on how to navigate the experiment and rate stimuli, by trying out a warm-up experiment. The main experiment followed next. These experiments were conducted on a CRT monitor

(1024x768 resolution), at a refresh rate of 100Hz, and the participants sat 60cm from the monitor in a dimly lit experiment room. Note that we had conducted a pilot experiment with lesser pseudowords, participants, smaller time constraints per stimuli, familiarity scale placed after wordlikeness scale, and different wordings for scales and instructions like "*Is a Word?* and *Is not a Word*". This informed us on a lot of decisions that follow in the below section.

3.3.3 Participant Demographics

The experiment was attempted by 44 native Hindi-speaking undergraduate students. (We required a minimum of 30 participants to obtain at least 4 ratings per pseudoword, but recorded more as they turned out to be accessible.) The participants included 14 female and 30 male participants, in the age range of 18 - 24 years (M = 20.205, SD = 1.440). Participants' self-rated proficiency in various tasks with Hindi was obtained on a 5-point Likert scale and was found to be as follows: for reading (M = 4.091, SD = 0.984), for writing (M = 3.727, SD = 1.065), for speaking (M = 4.386, SD = 0.689), and for understanding (M = 4.500, SD = 0.699).

Participants' self-rated language use in formal and informal settings was also obtained. The questions for this were adapted from the Language Use Questionnaire by [42]. The results for some of these were found to be as follows: for retelling a sequence of events or reciting a story (M = 4.705, SD = 0.594), while talking to friends and neighbours (M = 4.727, SD = 0.499), and while watching reels, series, or movies (M = 4.659, SD = 0.645). The questionnaire (Section B.3) and the detailed statistics for all the questions (Section B.2) can be viewed at Appendix B.

3.3.4 Design Choices

Before we detail the warm-up and the main experiment, we explain why and how we use certain elements of the experiment for easy replicability and change if required:

3.3.4.1 Rating Scales

We use a 7-point scale to record the metric as used by ([37], [16], [15] and many others) as they found that native speakers' judgements are not binary for pseudowords.

3.3.4.2 Response Times (RT)

We record RTs as we found that [43] recorded a strong effect of it in a phonotactic understanding experiment, while recently [44] studied Italian (a shallow orthography language) which (in contrast to English & French) showed that Pseudoword Superiority Effect was significant for RTs.

3.3.4.3 Familiarity measure

While [45] argues for the importance of recording this, a pilot also showed that its clearer to participants that the Wordlikeness scale does not require them to report their Familiarity with a stimulus exclusively when presented as a separate scale first.

3.3.4.4 Wording & Instructions

From our pilot, participants remarked that wording the Wordlikeness scale from *"Is not a word"* to *"Is a word"* is harsh. Thus for this experiment, we used *"Cannot be a word"* and *"Could be a word"*, for the extremes (1 & 7, respectively). While instructing, we followed [37] and [46] by telling participants that they need to think *'How good would . . . be as a Hindi word?'* and rating **1** would mean that the stimulus is *'so strange that it is unlikely for . . . to be used as a new Hindi word'* while **7** would be *'so Hindi-like that it would be easy to imagine . . . as a neologism.'*. Additionally, our pilot revealed that since Hindi (like English) has a collection of modern loanwords, it was advised to the participants to not judge the stimuli's Hindi wordlikeness negatively based on their presence in another language.

3.3.4.5 Stimuli Presentation

We present the stimuli (including pseudowords) to the participants in their orthographical form (and not audio samples) following [16]. They explain that this is because pseudowords could be misheard (consistent with [15]) and because alphabetic scripts like Devanagari can be used phonetically as well.

3.3.5 Warmup Experiment

A pre-cursor to the main experiment was a warm-up test consisting of 5 randomised word stimuli. Our pilot showed us that participant ratings for the first few stimuli would take more time and sometimes the experiment had to be restarted in case the participant was confused. Thus this warmup experiment is meant as a single solution to help participants be acquainted with the experiment setting and for the experimenter to explain some aspects of the experiment too.

The 5 stimuli consisted of random code-mixed, and/or low frequency words of the language. This was done to make sure that before rating the actual stimuli, participants could better understand that unfamiliar words (low-frequency stimuli) can be marked as low on the Familiarity scale and not necessarily on the Wordlikeness scale as it was revealed to them that the warm-up stimuli were all real Hindi words, setting the ideal wordlikeness score to 7. Similarly, possible loanwords/codemixed (or loanword-like) stimuli could be judged for their target-language-likeness independent of the source language.

3.3.6 Main Experiment

To evaluate the gradient acceptability of the pseudowords generated & the quality of a pseudoword generator, we can test the Alternative hypothesis that from a given set of words, generated pseudowords, and nonwords, a generation model produces good pseudowords if the native speakers judge the pseudowords as a category which is distinct from nonwords and closer to the words in terms of acceptability of their wordlikeness. (The Null Hypothesis to reject would be that from a given set of words, generated pseudowords, and nonwords, a generation model produces bad pseudowords if the native speakers judge the pseudowords as a category which is not distinct from nonwords and is either closer to the nonwords in terms of acceptability of their wordlikeness or the rated the same as them.)

Thus in a pseudo-randomised manner and balanced for length for all 3 categories, it presents the participant with a set of 12 words from the Shabd corpus [47] in the case of Hindi (additionally balanced for frequency), 12 handmade nonwords (by ensuring that they break a Hindi phonotactic constraint), and 12 pseudowords generated from the PseudRNN model above.

It took an average of 5 mins for the 44 participants to complete the main experiment, 10 mins overall:

1. consent & demographics section - 3 mins

2. warmup section + instructions - 2 mins

3. main experiment - 5 mins

Each stimulus was presented with the Familiarity scale and the Wordlikeness scale (previewed at Figure 3.1) closely following it for 5 secs each. In the case that a participant couldn't rate the stimuli in the given time period, we stored a default **-1**. Any two stimuli presented were distanced by a 1-sec gap.

We thus collected 44 ratings for all word and nonword stimuli for both the scales and at least 4 (in some cases 5) ratings for the 90 pseudoword candidates (as only 12 pseudowords were rated by each participant.)[3] We record and study 4 variables per stimuli, the Wordlikeness (or Wordlikeness Rating - `WR`) to gauge the gradient acceptability, Familiriaty Rating (`FR`) to make Wordlikeness Rating distant to the idea of Familiarity and to additionally understand how much a participant could confuse a given pseudoword with another familiar word, Time taken to gauge wordlikeness (`WT`) which is used as support to show that Pseudowords should be distinct from Non-words (as they are probably more easily identified as a non wordlike form because they break rules) and not necessarily follow Words (as they are known word forms)[4], and Time taken to gauge Familiarity (`FT`) as another support to the hypothesis & if clinicians need that information to conduct quick/difficult tests.

[3]Note: For each run a handmade script (available at the repository) picks the 12 pseudowords which have been rated the least times so far in the experiment.

[4]Results show interesting observations on how Pseudowords were more readily recognised as wordlike than low-frequency words in Hindi

3.3.7 Results required to evaluate model quality

To evaluate the hypothesis mentioned in section 3.3.6, we conduct ANOVAs on the Pseudoword Acceptability Metrics (4 ANOVAs, one for each metric) and pair-wise post-hoc tests to check the effect of the Broad Stimuli Type (Words, Pseudowords, and Nonwords) and Length (Short or Long, about 6 on transcribed length was the cutoff) on these metrics (Section 3.3.7.1). We then follow up on this analysis by doing separate ANOVAs on high-frequency, low-frequency, and pseudoword categories (Narrow Stimuli Type, here frequency is obtained from [47] and the extremes are used as high and low frequency stimuli) to see if there is a consistent trend outside of the more obvious non-words and if the quality of generated pseudowords is on an acceptable gradient (Section 3.3.7.2). Note that N=44 (no. of participants) for all the category-wise means used in the analyses, there is a tight significance level requirement of 0.001, and all the visualisations and tables are generated by JASP [48].[5]

len	**str_type**	**WR$_{\text{Mean (SD)}}$**	**WT$_{\text{Mean (SD)}}$**	**FR$_{\text{Mean (SD)}}$**	**FT$_{\text{Mean (SD)}}$**
short	word	6.288 (0.485)	1342.15 (506.241)	5.664 (0.573)	2853.16 (771.437)
	pseudoword	5.095 (0.965)	1803.51 (757.25)	3.019 (0.991)	4090.2 (1010.76)
	nonword	2.848 (1.09)	1784.2 (756.911)	1.949 (0.83)	3828.93 (1038.05)
long	word	6.424 (0.548)	1208.03 (486.716)	5.872 (0.721)	3079.74 (811.506)
	pseudoword	5.227 (0.933)	1857.16 (700.49)	3.113 (1.186)	4699.75 (912.43)
	nonword	3.433 (1.333)	2390.98 (911.8)	2.18 (1.006)	5267.6 (1323.15)

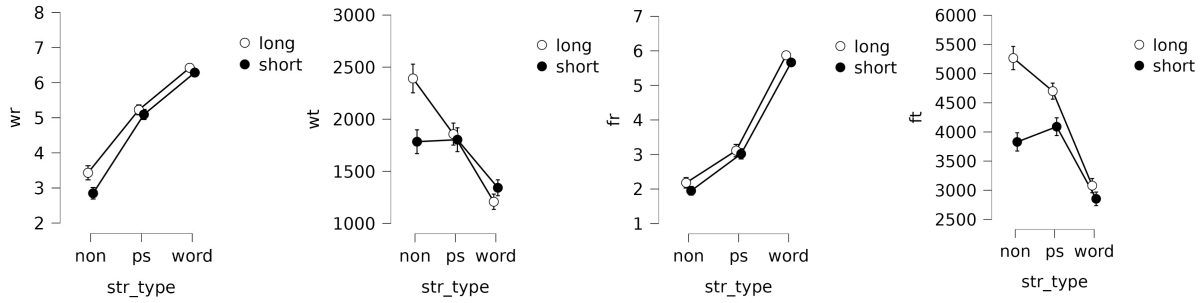
Table 3.2: Descriptive analysis/overview of results Means and Standard Deviations (SD) for Wordlikeness Rating (WR), Wordlikeness Response Time (WT), Familiarity Rating (FR), Familiarity Response Time (FT) across Length (len) (short or long) and Broad Stimuli Types (str_type: Words, Pseudowords, and Nonwords)

3.3.7.1 Effects of Stimuli type and Length on Pseudoword Acceptability Metrics

With a quick glance at the descriptive analysis (Table 3.2 & Figure 3.2) we can see that the Wordlikeness ratings (WR) for Pseudowords (pseudo) closely follow Words' (word), and are distanced from Nonwords' (non). Familiarity ratings (FR) follow the same trend. Finally, note that Wordlikeness Response Time (WT) & Famillarity Response Time (FT) maxes out on long types, especially nonwords (non), followed by pseudowords (pseudo), and then words (word).

A two-way ANOVA (Table 3.3) was performed to analyze the effect of Length (len: short or long) and Broad Stimuli Types (str_type: Word, Pseudoword, Nonword) on Wordlikeness Rating. Simple

[5]In this section, we present the results and they are also to be read as instructions on how to evaluate a pseudoword set generated and passed through the behvioral experiment as described above.



(a) WR (b) WT (c) FR (d) FT

Figure 3.2: Wordlikeness Rating WR, Wordlikeness Response time WT, Familiarity Rating FR, and Familiarity Response Time FT descriptive analysis, presented visually. (Note: nodes = Means, error bars = Standard Errors.

Cases	**Sum of Squares**	**df**	**Mean Square**	**F**	**p**	η^2
len	5.346	1	5.346	6.019	0.015	0.008
str_type	464.819	2	232.41	261.672	.001	0.663
len*str_type	2.975	2	1.488	1.684	0.188	0.004
Residuals	230.924	260	0.888			

Table 3.3: Two-way ANOVA results for Wordlikeness Rating WR as the Dependent Variable and Broad Stimuli Types (str_type: Words, Pseudowords, and Nonwords) & Length (len: short or long) as Independent Variables.

main effects analysis shows that length (len: short or long) did not have a statistically significant effect (p = 0.015), while Broad Stimuli Types (str_type: Words, Pseudowords, and Nonwords) did have a statistically significant effect on Wordlikeness Rating ($p < 0.001$).

Similarly, the two-way ANOVAs for Wordlikeness Response Time ($p_{len} = 0.048$, $p_{str\_type} < 0.001$) and Familiarity Rating ($p_{len} = 0.112$, $p_{str\_type} < 0.001$) follow Wordlikeness Rating, where only Broad Stimuli Types (str_type: Word, Pseudoword, Nonword) had a statistically significant effect. However, Familiarity Response Time (FT) ($p_{len} < 0.001$ and $p_{str\_type} < 0.001$) shows that both Length (len: short or long) and Broad Stimuli Types (str_type: Words, Pseudowords, and Nonwords) had a statistically significant effect.

Table 3.4 shows Post hoc comparison results of Broad Stimuli Types (str_type: Words, Pseudowords, and Nonwords) (averaged over levels of length (len: short or long) and the p-value adjusted to compare a family of 3) using Tukey HSD (and cross-checked with Bonferroni) test indicated that the Broad Stimuli Types (str_type: Words, Pseudowords, and Nonwords) are pair-wise signifi-

	Avg Diff	t	p	Cohen's d
pseudo-word	-1.2	-8.4	<0.001	-1.27***
non-pseudo	-2.0	-14.2	<0.001	-2.14***
non-word	-3.2	-22.6	<0.001	-3.41***

Table 3.4: Post Hoc of Broad Stimuli Types' (`str_type`: Words (`word`), Pseudowords (`pseudo`), and Nonwords (`non`)) effects on Wordlikeness Rating `WR`

	Avg Diff	t	p	Cohen's d
non-pseudo	-1.0	-7.3	<0.001	-1.11***
pseudo-word	-2.7	-19.8	<0.001	-2.99***
non-word	-3.7	-27.18	<0.001	-4.09***

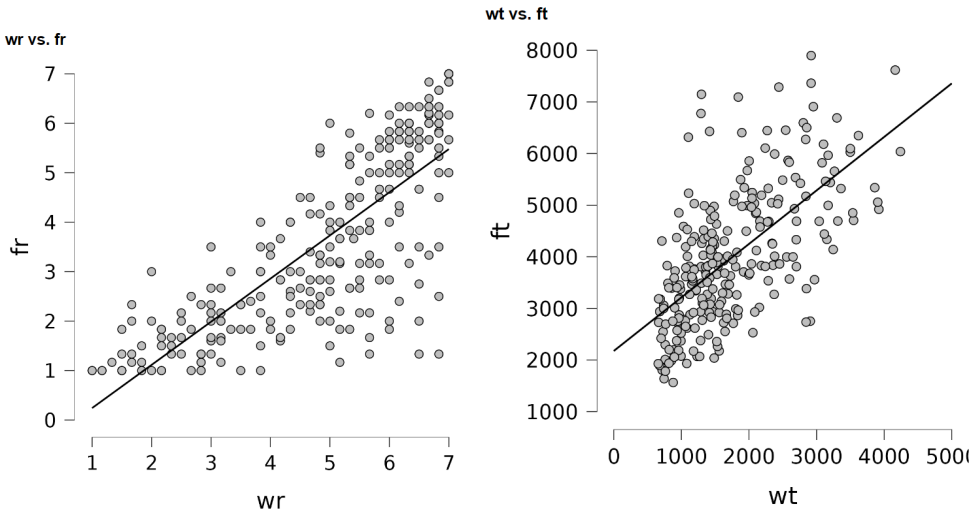
Table 3.5: Post Hoc of Broad Stimuli Types' (`str_type`: Words (`word`), Pseudowords (`pseudo`), and Nonwords (`non`)) effects on Familiarity Rating `FR`

cantly different from each other (in addition to the ANOVA across the 3 categories). The pseudowords (`pseudo`) - words (`word`) pair has a lower effect size (Cohen's d) than the nonwords (`non`) - pseudowords (`pseudo`) pair, indicating that the pseudowords (`pseudo`) is closer to words (`word`) than nonwords (`non`). The same is observed across Post hoc comparisons for Wordlikeness Response Time and Familiarity Rating (Table 3.5).

Post hoc comparisons of Length (`len`: short or long) (averaged over Broad Stimuli Types (`str_type`: Words, Pseudowords, and Nonwords)) indicated that the mean time taken for long strings for Familiarity Response Time was significantly different from the short ones ($p<0.001$). For Wordlikeness Rating and Wordlikeness Response Time, they were not very significantly different ($p=0.015$ and $p=0.048$ respectively) and it wasn't significant for Familiarity Rating at all ($p=0.196$).

	Pearson	Shapiro-Wilk
WR-WT	-0.243***	0.921***
WR-FR	0.787***	0.949***
WT-FT	0.634***	0.98***

Table 3.6: Pearson's correlations (for Broad Stimuli Types (`str_type`: Words, Pseudowords, and Nonwords)) for various pairs among Wordlikeness Rating (`WR`), Familiarity Rating (`FR`), Wordlikeness Response Time (`WT`), and Familiarity Response Time (`FT`).



(a) WR-FR (b) WT-FT

Figure 3.3: Correlation Plots between Ratings (Wordlikeness (WR) & Familiarity (FR)) and Response Times (Wordlikeness (WT) & Familiarity (FT))

Correlations: As shown in the Table 3.6 & Figure 3.3, a Pearson correlation coefficient was computed to assess the linear relationship between the ratings of Wordlikeness and Familiarity. There was a significantly positive correlation between the two variables. We can see this between the response times as well, and between Wordlikeness Rating & Reaction Times (although a negative correlation).

Cases	**Sum of Squares**	**df**	**Mean Square**	**F**	**p**	η^2
str_type_2	194.755	2	97.378	195.819	.001	0.752
Residuals	64.15	129	0.497			

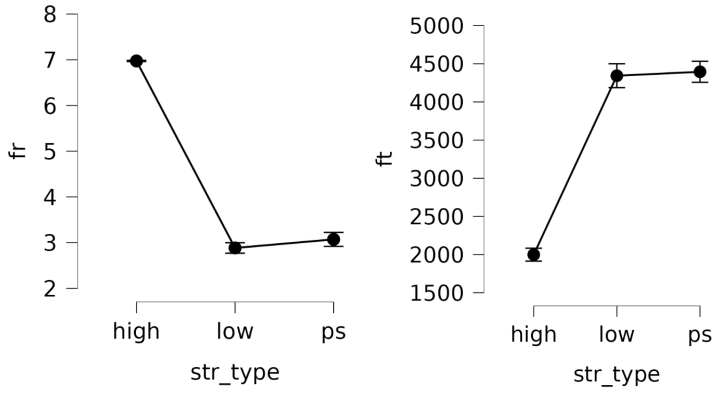
Table 3.7: One-way ANOVA results for Wordlikeness Rating WR as the Dependent Variable and Narrow Stimuli Types (str_type_2: High Frequency Words, Pseudowords, and Low Frequency Words) as the Independent Variable.

3.3.7.2 Impact of frequency on Pseudoword Acceptability Metrics

In addition to the results above, we also check if the quality of pseudowords (pseudo) compares against high-frequency (high) and low-frequency (low) words. A one-way ANOVA was performed to compare the effect of these categories (Narrow Stimuli Types (str_type_2: High Frequency Words, Pseudowords, and Low Frequency Words)) on Wordlikeness Rating (Table 3.7). It revealed that there was a statistically significant difference in mean Wordlikeness Rating between the 3 categories (F(2) =

	Avg Diff	t	p	Cohen's d
high-low	2.95	19.6	<0.001	4.2***
high-pseudo	1.8	12.0	<0.001	2.6***
low-pseudo	-1.15	-7.7	<0.001	-1.6***

Table 3.8: Post Hoc of Narrow Stimuli Types (`str_type_2`: High Frequency Words (`high`), Pseudowords (`pseudo`), and Low Frequency Words (`low`)) effects on Wordlikeness Rating `WR`



(a) `FR` (b) `FT`

Figure 3.4: Descriptive Plots for Narrow Stimuli Types (`str_type_2`: High Frequency Words (`high`), Pseudowords (`pseudo`), and Low Frequency Words (`low`)) against Familiarity Rating (`FR`) & Familiarity Response Time (`FT`). Note that the X-axis label on the graphs imply high_freq._word - low_freq._word - pseudoword comparison.

[195.8], p <0.001). Post hoc tests (which are significant pairwise) show that pseudowords (pseudo) is closer to high than low as shown by Cohen's d values (Table 3.8) which are bigger between high and low as compared to high and pseudowords (pseudo). Familiarity Rating (Figure 3.4a) and Wordlikeness Response Time follow the same. However, Familiarity Response Time (Figure 3.4b) shows that high is marginally closer to low than to pseudowords (pseudo).

To summarise, we looked at the various components of Pseudoword Acceptability Metrics, understanding the differences between the Broad Stimuli Types (& checking for the impact of length, if any), and then looking deeper into the same but for frequent words, rare words, and pseudowords.

3.4 Interpretation of results & Model quality evaluation

3.4.1 Interpreting quantitative metrics

3.4.1.1 Pseudoword Acceptability

ANOVA on Broad Stimuli Types (str_type: Words, Pseudowords, and NonWords) (Table 3.3) and the post hoc analyses (Table 3.4) show that Hindi pseudowords generated by PseudRNN have been rated closer to words than nonwords (on Wordlikeness Rating scale) even though the participants find nonwords & pseudowords less familiar than words (Table 3.5). Interestingly, participants also found pseudowords to be more acceptable as a word than low-frequency words (see analyses in section 3.3.7.2). Thus we see that pseudowords are on a *gradient* between both nonword-word & high_frequency - low_frequence extremes (Figure 3.2a & Figure 3.4a respectively). Thus the Pseudoword Accpetability Metrics tell us that the model generated pseudowords are thought to be wordlike and familiar by the participants, readily (in terms of response times). The Null hypothesis is also disproven by obsering that the pseudwords do not lie closer to non-words than the words.

To support the fact that the experiment design and the stimuli set were suitable for this task, the correlation between the Wordlikeness and Familiarity ratings (WR-FR at Figure 3.6) is high at the extremes, implying that participants were sure that words are extremely wordlike and familiar, which non-words are neither. Through these matching results on expected and known stimuli (words and non-words) we can rely on the experiment design while also expecting a similar distribution from any other runs of the experiment, regardless of language.

Finally, like [49], we also find that the time taken to decide the acceptability of stimuli as word (WT) was consistently the shortest for actual words and that the participants took more time deciding on pseudowords, while long nonwords took the most time (Familiarity Response Time confirmed the same trend). This is probably due to multiple phonotactic rules broken over the length. Interesting statistic supporting this result is the distance of the closest word in the lexicon and Familiarity Reaction Time which was a significant correlation (p <0.001).

3.4.1.2 Existing metrics

Comparing Soodkosh to existing metrics shows that Hindi pseudowords are also found to be fairly *legal*. Since there are very few polymorphic tokens (`poly`), the model might not be *suitable* for some tasks. On the other hand, these comparisons are crosslingual and analyses might not be well-founded. As an example: The average Wordlikeness Rating ($\texttt{WR}_{\text{mean}}$) for pseudowords (`pseudo`) with `C+` errors in Soodkosh is 5.6, for `V+` it is 5.7, and for `CV+C` it is 5.12. Showing that the participants accept the wordlikeness of suspect pseudowords too.

Since the work on *legality & suitability* is proposed as language agnostic, we did initial analyses only on Hindi. We then ran an English lexicon through the pipeline to generate comparable results. We found that (ref Table 3.1) PseudRNN generated English pseudoword candidates were considered not legal in comparison to other methods, they were much more suitable than most other methods in the categories of `1-C`, `comp`, `poly`, and `npoly`. This result is probably due to the fact that the algorithm to judge legality works orthographically. We know that phonemes of a langauge are organised in hierarchies and there are sisters which majorly work with the same sequence constraints (this fact also is exploited by a generative phonology modelling work [23]). Thus a modified version of the algorithm could be to find illegal sequences based on sister phonemes too e.g. if "rm" is legal and "lm" is not, then consider "kalma" as a legal pseudoword, as "r" & "l" are both liquids.

Thus with a modified version (and with appending more metrics of the same family like `half-real` and `C+V+C+`) of the existing metrics of the *legality & suitability* domains, one could gather more information on pseudowords & compare them against other methods possibly. These metrics are also eventually useful as features, as an example: clinicians might want to use pseudowords with more illegal sequences to make an easy set.

3.4.2 Interpreting qualitatively

After matching expectations from research & intuition of the Pseudoword Acceptability Metrics gathered from the experiment, and comparing our generated pseudowords against other methods, we now present some qualitative observations to understand how one could look at the results to find out more.

Starting with Soodkosh (Hindi) we found that our ratings were distributed all the way from a perfect 7 (word) to 3 (just above nonword). To understand the perception of participants, we looked at pseudowords rated highly, mediocre, and low separately. Some common observations (as displayed in Figure 3.5) were that:

1. Highly rated Hindi Pseudowords: apart from some Pseudowords which were orthographically & morphologically too complex to breakdown, we observed that a lot of the pseudowords looked like a misspelling/possibly alternate spelling of actual words (as an example, the first two words in the figure).

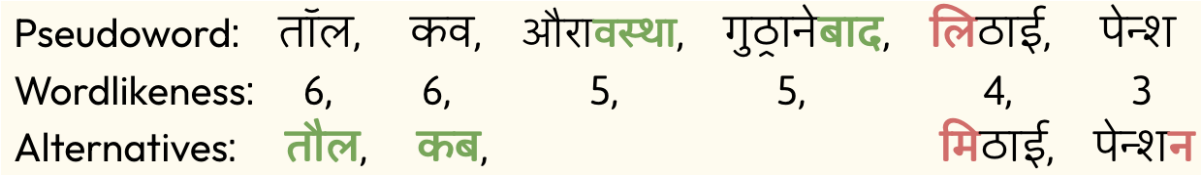


Figure 3.5: Some examples from Soodkosh for qualitative analysis

2. Mediocre rated Hindi Pseudowords: a lot of the pseudowords around this range (3rd & 4th strings in the figure) contained an existing & semantically meaningful (not only grammatical) suffixes.

3. Low rated Hindi Pseudowords: A lot of these were visibly an incorrect/implausible variation of a popular word. Unlike high rated pseudowords, these cannot be a spelling mistake.

Thus the pseudowords generated were interpreted and rated possibly on a variety of ways, including orthographical typos (high), morphological possibility because of meaningful suffixes (mid), and implausible forms of popular words (low).

On the other hand, annotators for English pseudowords generated by PseudRNN were unsure on how to adapt to the suitability metric in case of pseudoword candidates like: *macelousness* & *reformentness* where there were multiple fake and real affixes with a fake root. This is as the suitability metric restricts the analysis to only one of each (root or affix) possibly because of how the paper proposing it only went upto 8 length n-grams. Secondly, as instructed by the paper, we included only *a*, *e*, *i*, *o*, and *u* as vowels but there were situations were semivowel consonants functioned as vowels and were not counted to calculate the errors/were the cause of more errors as they were treated as improbable Consonant clusters.

3.4.3 Conclusion

Through these analyses we find PseudRNN's generation of Hindi pseudowords is acceptable and that the pipeline & evaluation strategy of collecting gradient acceptability of wordlikeness, judging legality, and suitability can be used to build models/datasets for high-quality pseudowords. We also compared the model to other methods by processing an English lexicon through the same pipeline, presented observations, and explained them.

Between the analysis of wordlikeness ratings & legality errors (section 3.4.1.2 and that of familiarity response times against orthographical distance of the closest word in lexicon (section 3.4.1.1), we see that the ratings by participants for Hindi pseudowords are not directly explained by the existing metrics but a more by a combinaiton of multiple features, justifying the need for more metrics that can bring out features that are included in the perception of a native speaker while judging these novel forms.

The study was done on Hindi but can be possibly expanded to other languages as it abstracts away language-specific requirements in both generation (LSTM trained on phonemic transcriptions) & evaluation (wordlikeness gradient acceptability test requiring an experiment setup with words, nonwords, and candidate pseudowords from the language).

3.5 Soodkosh

Finally, we present Soodkosh, a dataset of 90 Hindi pseudowords generated by PseudRNN with its features (Table 3.9). It incorporates:

1. the Means of each Pseudoword Acceptability Metric for use in lexical decision tasks like selecting a relevant pseudoword according to its wordlikeness a.k.a. difficulty in distinguishing it from a word.
2. features from [1] to gauge if a word is orthographically legal & morpho-lexically suitable for a task e.g. on an L2 language learning test set on affixes.
3. distance from the closest word in Hindi Wiktionary (both IPA-wise & Orthographically), as used by phonology modeling & probing studies.
4. count of Syllables & Phonemes.
5. features adapted from the Shabd corpus [47]: Matras (count of vowel ligatures), Aksharas (count of consonant ligatures), and Orthographic Length (aggregate of Matras & Aksharas) are specific to Hindi psycholinguistic studies

Table 3.10 also lists out the features for PseudRNN generated English pseudowords.

3.6 Future Work

As stated before, Pseudowords can be evaluated on more metrics like: their phonotactic probability, closest word's lexical frequency etc ([14] and [50]). Gorman [40] argues that wordlikeness judgements are not necessarily representative of wellformedness, thus it would be interesting to innovate language-agnostic measures that can also approximate a target language's phonology in terms of wellformedness, this is supported by the fact that the generated English pseudowords by PseudRNN were not legal according to existing metrics but this is probably because of the fact that the meager 3-4 metrics of legality are not enough to judge well-formedness.

The other improvement could be done on automating the entire process & removing the need of annotators to convert IPA sequences to Orthographical ones. That could be done by algorithms which look at the IPA sequences in a candidate pseudowords and find the highest probable orthographical sequence that could replace it. If the sequence in novel, the algorithm could also search by replacing some phonemes in the sequence with sister alternatives.

3.7 Limitations

Works like [49] use the phonotactic probability of constituents in a string to gauge wordlikeness. This has been a key feature but can be only reliably calculated for large lexicons with a good balance

Features	Mean	SD	Range
$\texttt{WR}_{mean}$	5.161	0.867	3.167 - 7.0
$\texttt{WT}_{mean}$	1835	492.9	66.3 - 3603
$\texttt{FR}_{mean}$	3.072	1.062	1.6 - 6.0
$\texttt{FT}_{mean}$	4299	781.5	2732 - 6129
comp	0.022	0.148	binary
poly	0.056	0.230	binary
npoly	0.322	0.470	binary
1-C	0.267	0.445	binary
half	0.122	0.329	binary
C+	0.044	0.207	binary
V+	0.011	0.105	binary
CV+C	0.144	0.354	binary
C+V+C+	0.678	0.470	binary
$\texttt{C+V+C+}_{\#}$	0.933	0.804	0 - 3
Dist-IPA	2.678	1.620	1 - 7
Dist-Dev	2.400	1.279	1 - 7
Orth-Len	6.667	2.380	2 - 13
Matras	1.922	1.114	0 - 5
Aksharas	3.822	1.255	2 - 7
Syllables	2.722	1.039	1 - 5
Phonemes	6.544	2.284	2 - 12

Table 3.9: Descriptive stats for Soodkosh – Hindi

Features	Mean	SD	Range
comp	0.115	0.322	binary
poly	0.167	0.375	binary
npoly	0.462	0.502	binary
1-C	0.308	0.465	binary
half	0.218	0.416	binary
C+	0.769	0.424	binary
V+	0.333	0.474	binary
CV+C	0.987	0.113	binary
C+V+C+	0.987	0.113	binary
$C+V+C+_{\#}$	2.500	1.041	0 - 4
Dist-IPA	2.218	1.147	0 - 6
Dist-Orth	2.400	1.179	1 - 6
Orth-Len	9.115	2.444	4 - 15
Syllables	3.346	1.115	1 - 6

Table 3.10: Descriptive stats for English pseudowords generated by PseudRNN

of quantities in varieties of such constituents. For low-resourced languages that fulfil their pseudoword requirements in an ad-hoc fashion, such a feature could prove to be very helpful. However, there is no research on utilizing low-resource lexicons in innovative ways to extrapolate such features, a limitation of this work thus is that we could not build a pipeline which was more frugal than neural networks as it was difficult to calculate these statistics reliably for languages like Hindi that we want to scale to.

The other limitation was the scale of experimentation and the quality of participants. We could only collect about 4-5 annotations per pseudoword (total of 90) in a 15 min experiment. For a more concrete rating that represents various dialects and styles of a language, we need a bigger scale of experimentation with a more linguistically diverse native speaker base. It follows that this small-scaled experiment could not gauge a person's proficiency in the language outside of the self-report on various measures. Thus a better way to vet a volunteer's language expertise & nativeness would help.

Finally a related limitation to above was the access to English natives who are also IPA experts in our study. Thus to gauge suitability, the pipeline would require such people to be accessible as a part of the study in another language.

Chapter 4

The Impactfluence of Pseudowords

4.1 Introduction

This chapter showcases the application and importance of pseudowords in fields outside of psycholinguistics. There are two explorations detailed:

1. Use of pseudowords in studying Metacognition of Passwords: This work checks for the correlation between the perceived memorability and the perceived security of a password. To be able to establish a correlation, a gradience in stimuli was needed. As a result, this study uses the analysis of Pseudowords based passwords as a crucial step after analysis of High frequency words & Low frequency words and before implausible & unpronounceable passwords. It is also used in important experiments in the work to support the main argument (as used at section 4.2.5).

2. Replacing pseudowords with "Average" words for a standard NLP task in Aphasia: This study explores Aphasic data classification by a standard language model. Since Aphasic data is incoherent largely due to the presence of pseudowords like: neologisms, words with switched phonemes etc., this study aims to understand the difference between classifying Aphasic data with and without pseudowords. This is done by making a new 'pseudoword' replaced dataset, where another language model predicts the masked out-of-vocabulary item that is a pseudoword. Note that this an ongoing study and the progress presented here is just a summary of preliminary research done so far.

From the first study we can see how pseudowords are crucial in an experiment in the field of cybersecurity and psychology. On the other hand the second work shows that naturally occurring pseudowords (in Aphasic speech) are irreplaceable and might also be informing a language modelling system to some extent. They thus need to be studied more carefully and exhaustively. We need to find more about their linguistic properties and how to process these wordlike tokens that do not have a semantic representation, while utilising their benefits of being phonologically well-formed.

4.2 Is convenient secure? Exploring the impact of metacognitive beliefs in password selection

4.2.1 Overview

Recently, there has been research on what factors influence a user's password setting practices, which include various types of emotions such as anger, risk-taking tendencies, etc. However, research has shown that factors such as memorability and perceived memorability have a greater influence on password choice. Some recent research has shown a negative correlation between the *perceived memorability* and the *perceived security* of passwords, particularly passphrases (that are technically more secure). However, it is unclear whether this effect can be extended to groups with good experiences with digital spaces (IT professionals, entrepreneurs, etc.). Furthermore, it has not been determined whether random, uncommonly-worded, or complex structure passphrases would also maintain the correlation, as opposed to relatively less secure, common/simple passphrases. This study examines this problem using a diverse demographic and different categories of passphrases.

4.2.2 Introduction

Password strength is critical for healthy digital interaction, especially in recent years, with growing trends in the use of applications, digital devices, and other highly secure digital interfaces. It is becoming increasingly important to understand user's views on password setting practices. Nordpass (2020) reported that last year 2,543,285 people set their passwords to `123456`. This is a vulnerable password to choose, even though previous studies have shown that users know about standard safe password setting practices like using different types of characters, not using personal information, dictionary words, common combinations, etc. (Woods & Siponen, 2019).

This is not just limited to passwords, there is also a particular interest in passphrases. Recent studies have specifically found that the best way to create strong and memorable passwords is to use four or more words [51], which implies that passphrases and their memorability is proving to be an increasingly important area.

There are numerous situational, theoretical, judgement-based factors found behind the unsafe password setting behaviour over the years. An interesting method is to link metacognitive theories to explain the same. Metacognition is thinking about thinking itself. The two major processes of "monitor" and "control" [52], were interpreted in this context as "Users **monitor** passwords and decide on their security". This in turn "**control**s their decision" on whether to use the password in a particular environment.

This interpretation was tested by [53], which builds on research showing that memorability and security have a negative correlation, and the study examines whether perceived memorability (PM) has a similar correlation with perceived security (PS), i.e. do users believe "an easy to remember password is not secure"?

The study surveyed 40 Portuguese university students and found that the more heterogeneous a password is, the more secure it is perceived (PS), and the less memorable (PM). For example passwords with just lowercase characters (like `jfhdnele`) are less heterogeneous than passwords with a mix of lower and uppercase characters, symbols, and numbers (like `hR5@io88`).

They also found that passphrases were not considered (like `no longer freshman` [53]) as the most secure type of passwords. The authors concluded that the PS values for passphrases were ranked lower than some other categories. This is because their PM values were higher than most categories perceived as secure such as the category with a mix of lowercase, uppercase and numbers.

They concluded that PM & PS have a negative correlation and these results were reinforced by analysing intention of use levels in a critical vs. a non-critical website scenario. However, upon closer examination, we found that this study and other similar studies, did not focus on what could be these **other factors** (that affect the PS of passphrases), the participants **were not diverse** in terms of experience, and the passphrases that were used were **limited** to meaningful/easy to remember sentences.

In this study, we address these issues and hypothesize that a diverse participant base will **show trends** in the behaviour but also that it would generally be consistent. This is because even if a user knows these factors, they would ultimately act on perceptions and **not** to **facts**. Second, since we study pass**phrases**; the order of the words, their commonality, etc. may also be responsible for how users perceive their security. Therefore, we would **not** observe a strong negative correlation between PM and PS across *well-structured and simple passphrases* and *complex/uncommonly worded passphrases*. Finally, in order to understand the population that uses mobile devices frequently, we have also included use cases such as non-critical **mobile applications**.

4.2.3 Method

This study was conducted in 2020 and therefore had to be conducted "completely online" (due to the coronavirus pandemic). We selected Psytoolkit [54, 55] to script and float it to a diverse demographic of participants for 20 days in November 2020. This section describes the demography, the resources used for the survey, and the conduct of the survey.

4.2.3.1 Participants

We collected a total of 118 complete responses and after looking at our response times for the pilot ($N = 12$), we found that the minimum time required to complete the survey was approximately *10 minutes*. The 7 responses that took lesser time to finish and one response that indicated that the participant was uncomfortable with English were not included in our analysis. Finally, we had a set of 110 responses to analyse (42 Females), from a broad age group (range: 14-72 years; M=29.74 years; SD=13.3 years), a wide range of educational backgrounds, (12th grade or below: 6 (5.45%); college degree (current/completed): 63 (57.27%) and Masters/PhD, etc: 41 (37.27%)) and a diverse professional background (student: 55 (50.0%); unemployed: 4 (3.63%); retired: 6 (5.45%); employed: 45 (40.91%)).

#	Item	Response
1	Approximately how many passwords do you use on a daily basis?	M: 5.45 ; SD: 3.71
2	Frequently used passwords are easier to remember.	96.40% agree
3	An easy to remember password is a safe password.	36.94% agree
5	Using characters of different types in a password is safer than characters of the same category.	81.98% agree
6	Have you been hacked before?	10.81% say yes
7	A shorter password is less secure than a longer one.	59.46% agree
8	A password based on personal information or dictionary entries is secure.	16.22% agree
10	A complicated/difficult to remember password is more secure.	83.78% agree

Table 4.1: Security Awareness section Responses

Since the survey material was exclusively in English, we asked the participants to report their knowledge of English (basic: 6 (5.45%); Good: 14 (12.72%); Professional: 48 (43.63%); Fully-Professional: 25 (22.72%) and Native-Speaker: 17 (15.45%)).

4.2.3.2 Material

Materials used for the survey, how they were collected, etc. are described below.

- **Passwords:** For the experiment 45 passwords were used, which were divided into 9 categories, with 5 passwords each. The 9 categories were: (***LF***): Low-Frequency Words (such as `meteoric`), (***HF***): High-Frequency Words (such as `children`), (***PD***): Pseudowords (such as `dwaughts`), (***LC***): Lowercase (such as `mjzxxvyt`), (***+U***): Lowercase + Uppercase (such as `ShpzczSo`), (***+N***): Lowercase + Uppercase + Numbers (such as `47Qn3nUD`), (***+S***): Lowercase + Uppercase + Numbers + Special Characters (such as `qy~c)Aw4`), (***CP***): Common Phrases (such as `the book is under the table`), and (***RP***): Random/Complex Phrases (such as `shake medicine read floor`).

  - **Categories 1 to 7** were all 8 characters long and the last two categories were 21-23 characters long with an average length of 22 characters. Methods of acquiring these passwords are fully reproducible and randomized where they could be. All the following passwords and passphrases were selected based on a normalised, averaged, and aggregated total of their security ratings by multiple websites as referenced [56, 57].
  - For **Categories 1 and 2**, we used the MRC Psycholinguistic Database (Wilson, 1988) with filters on Brown Frequency, Kucera-Francis Frequency, and Thorndike-Lorge Frequency

apart from length and then selected the results accordingly (for exact filtering methods, see the linked shared folder: `https://bit.ly/isconvenientsecure`). Similarly, for the 3rd category, we used the ARC Nonword Database [17].

  - For **categories 4 to 7**, we used KeePassX 2.0.3 to generate passwords filtered by length, entropy, etc. For **Category 8**, we used English learning websites such as EnglishSpeak (EnglishSpeak, n.d.), to find introductory sentences in English and filtered them by length.
  - Finally, for **Category 9**, we used passphrase generators (randomised) such as "Use a Passphrase" [58], etc. and filtered some according to their length and whether they contained known but rare words.

- **IMS Section:** Since the experiment was conducted **online**, the Immediate Mood Scaler (IMS) [59] helped us determine the mental state of the participants and analyze whether they responded in a stable mood. It was the standard 24-item inventory with a 1-7 scale for mood pairs such as "depressed" or "happy", etc. Some items were: distracted or focused, hopeless or hopeful, etc.

- **Security Awareness Section:** The main judgement tasks, were followed by a short questionnaire consisting of 10 objective questions. 8 of which were a basic security health and awareness assessment through questions such as "How often do you change your passwords?" and "Using characters of different types in a password is more secure than characters in the same category." (Yes/No). These were selected on the basis of previous studies and from inventories used in other password preference studies [60, 53, 61]. The other two were binary answer questions that helped us understand whether particpants' beliefs matched the judgements done in the previous sections. The questions were *"A complicated/difficult to remember password is more secure."* (like `Tr0ub4dor&3`) and *"An easy to remember password is a safe password."* (like `correct horse battery staple`).

4.2.3.3 Procedure

We shared the link to the Psytoolkit form with willing participants who were informed that it took about 30 minutes to complete. The form consisted of 6 parts, which were presented to them in the following order:

- **Consent & Demographics:** In this part, the anonymity of the data and its use were clearly explained. The participants were also informed that this should be done without interruption except between some sections. We then asked for basic demographic details such as age, gender, profession, fluency in English, etc.

- **IMS:** After filling in the demographic data, participants read a description of the IMS scale and had to scale their emotions to the 24 items, according to their current behaviour.

- **Memorability Judgement:** This section was the first of three sections which presented the 45 passwords for user judgement. We asked participants to rate each of these passwords on a scale from 0 to 100% (less memorable to more memorable), by reflecting on the following prompt for each password: *"How likely are you to remember this password 2 days from now?"*.

- **Security Judgement:** Security was the second judgement task. We asked the participants to rate each of the 45 passwords on a scale from 1 to 6 (not secure to very secure) while thinking about the prompt: "How secure is this password?". The passwords were in the same order as in the last section.

- **Usability Judgement:** The final judgement task was to select all possible use cases for the given password. We listed 5 use cases for each of the 45 passwords and asked users to select all possible cases in which the displayed password could be used. A sample prompt: *For the password "sample_password", select all scenarios you could use this for: (Please select all situations that apply for the particular password. Do not select situations that do not apply for this password.)*

  - *In an Important Online Service like banking online on SBI*
  - *In a Casual Online Service like reading an article on Medium or some e-newsletter*
  - *When registering is time-bound and you need to fill in a password quickly.*
  - *Using Personal/Private Accounts on apps like Instagram or Facebook.*
  - *Utility Apps/Gaming Apps like Calculator or Candy Crush, Temple Run, etc.*
  - *None of the above.*

- **Security Awareness:** As mentioned in the Materials section, participants were asked 10 objective questions about their opinions and awareness of secure password setting practices. After completing this part of the survey, the participants were thanked for their participation and forwarded to the Google homepage via Psytoolkit.

4.2.4 Results

For this section, Analyses of the variance (ANOVAs) were calculated across the seven measures (PS, PM, and usability in 5 environments) for each of the 9 password types, Pairwise Student's t-tests between the values of the 7 metrics for each password type (total 63 comparisons) are mentioned, indicating a trend. Pearson correlations are averages over the respective values retrieved from all 110 participants and the full record of the data as well as the statistics are available at: `https://bit.ly/isconvenientsecure`.

4.2.4.1 Perceived Memorability

An ANOVA (number of comparisons detailed above at section 4.2.4) showed significant differences, $F(8, 848) = 119.02$, $p < .001$ (Figure 4.1). In decreasing order of PM ratings, HF and CP are ranked first, followed by LF, RP, and PD. LC, +U, +N and +S ranked lowest. In our study, however, the **passphrases** were additionally branched into CP and RP, revealing a significant difference between them in terms of their PM ($t = 12.17, p < .001$).

4.2.4.2 Perceived Security

An ANOVA (number of comparisons detailed above at section 4.2.4) showed significant differences, $F(8, 848) = 96.89$, $p < .001$ (Figure 4.1). The general order of PS within password types showed the opposite trend compared to PM (except for CP and RP). In addition to previous studies, the highly negative Pearson's correlation $r = -0.92$ also supports this trend.

4.2.4.3 Usability in Specific Environments

- **Critical Services (CritWeb):** An ANOVA showed significant differences, $F(8, 848) = 190.08$, $p < .001$ (Figure 4.2). Of the 10 top-rated passwords for critical services, 5 were "+S" and 4 were "+N". All of these passwords are *character-level* and not dictionary entries. In Figure 4.2 as we move from top to bottom, we see a sharp rise in the *"Intention of use"* for Critical Services with the addition of more character classes (+U, +N), peaking at +S. These ratings closely follow the PS ratings, with a Pearson correlation of r = 0.93.

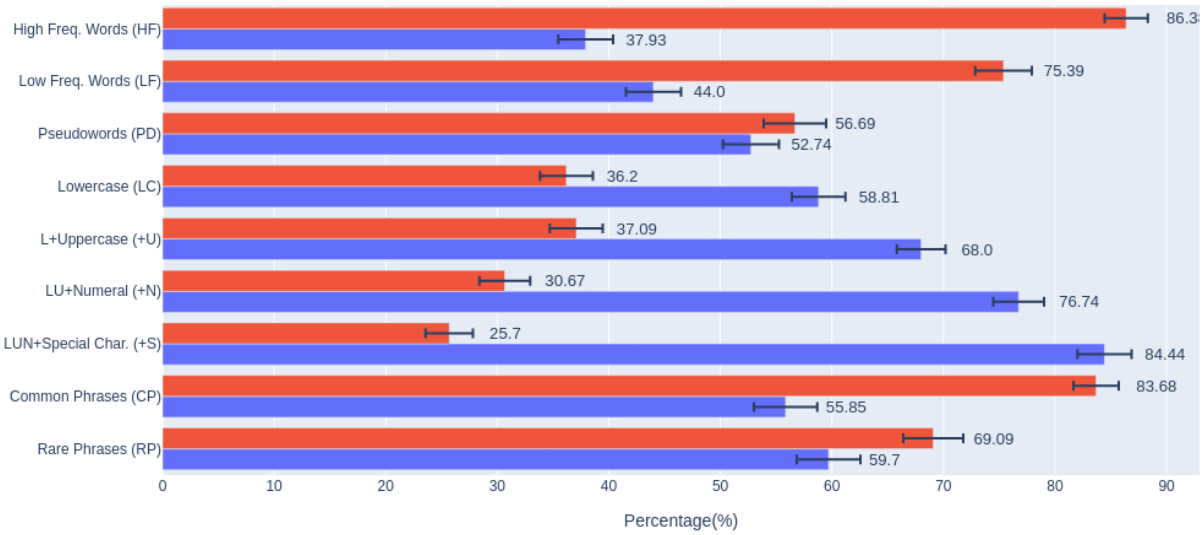


Figure 4.1: Mean ratings for PM and PS for each type of password. Key: (Blue: Perceived security, Red: Perceived memorability)

Table 4.2: question-4: "When do you normally change passwords?" Responses

Response	**% of participants**
On forgetting	28.82%
As the service reminds	20.72%
Depends on the service	20.72%
Regularly (every month)	7.21%
Rarely (annually)	17.12%
Never	5.41%

- **Non-Critical Services (NonCritWeb):** Unlike in previous studies, the ANOVA showed significant differences in usability share *even* for non-critical services, $F(8, 848) = 11.29$, $p < .001$ (Figure 4.2). Of the 10 top-rated passwords, 4 were PD and 2 of HF.
- **Time-Bound Services (Time):** An ANOVA showed significant differences, $F(8, 848) = 22.61$, $p < .001$ (Figure 4.2). Of the 10 top-rated passwords, 5 were LF and 4 HF. With a Pearson correlation of r = 0.82, the usability ratings for time-bound services resemble non-critical services.
- **Critical Apps (CritApp):** An ANOVA showed significant differences, $F(8, 848) = 15.35$, $p < .001$ (Figure 4.2).
- **Non-Critical Apps (NonCritApp):** An ANOVA showed significant differences, $F(8, 848) = 16.32$, $p < .001$ (Figure 4.2). With 4 PD and 3 HF among the 10 top-rated passwords, usability in non-critical applications shows a very similar behaviour to non-critical and time-bound services (Pearson's correlation of $r = 0.95$ and $r = 0.84$ respectively).

Finally, The 10 top-rated passwords *per usage environment* did not consist of Common or Random **Passphrases**.

4.2.4.4 Security Awareness

Tables 4.1, 4.2, and 4.3 (of the Security Awareness section) show that the majority of participants were aware of common safe password setting practices and had not yet been hacked. Question 7 confirms that not everyone knows that length is important for a technically more secure password.

79.28% of the participants stated that they use between 0 and 10 passwords daily (question 1), 47.7% rely exclusively on their memory to store the passwords (question 9), and 18.18% prefer to use the 'forgot password' option over memory. 50% of the participants who use 0-10 passwords daily change passwords only when reminded of it by the service, and a further 23.86% rarely change their passwords.

Table 4.3: question-9: “Check all options where you have passwords stored now:” Responses

Option	**% of participants**
Sticky notes on digital devices	13.51%
Noting offline	23.42%
Password Manager	15.31%
Nowhere (rely on memory)	48.65%
Rely on OTP/Forgot Password	18.02%
Some other place	21.62%

4.2.4.5 Demographics

We decided to do a correlational analysis between Demographics & passwords' Usability Environments, between Demographics & PM, and between Demographics & PS. We observed violation of normality assumptions by using Shapiro-Wilk test for each of the (above mentioned) series (p-values for all series were found to be less than 0.05). Thus we selected Spearman's correlation coefficient to study these correlations.

The participants were aged between 14 - 72, (M: 29.745 ; SD: 13.3). There were 42 women (38.18%), 66 men (60%), and 2 preferred not to say. 6 participants were currently enrolled in a school (5.45%), 63 in a college/university (57.27%), and 41 have graduated or are pursuing a higher degree (37.27%). There were a total of 55 students (50%), 4 were unemployed currently (3.64%), 45 were employed (40.91%), and 6 were retired (5.45%). Finally 6 participants reported *Basic* understanding of English (5.45%), 14 reported *Good* (12.73%), 48 reported *Professional* (43.64%), 25 reported *Fully Professional* (22.73%), and 17 reported themselves as *Native Speakers* (15.45%).

We also used some groupings of password categories for the results in this section (see Tables 4.4 & 4.4), they are as follows: *"words"* includes High & Low Frequency words, *"heterogeneous"* includes Lowercase, Lower+ Uppercase, Lower+ Upper+ Numerals, and Lower+ Upper+ Numerals+ Special Characters, *"common"* includes High frequency words and Common Phrases, *"rare"* includes Low frequency words and Rare Phrases, and finally *"passwords"* includes all categories except Common and Random Phrases.

Similarly, some usability environments were also grouped: *"crit"* includes Critical Apps and Services, *"webapp"* includes all (critical or not) Apps and Services, and *"app"* includes Critical and Non-Critical Apps.

The demographics (apart from *age*) were numericalised using the following mapping:

- *English Proficiency (eng)*- 1: No knowledge, 2: Basic, 3: Good, 4: Professional, 5: Fully Professional, and 6: Native Speaker.

- *Profession (occ)*- 1: Student, 2: Unemployed, 3: Employed, and 4: Retired.
- *Education (edu)*- 1: No Schooling, 2: 12th Grade or below, 3: College Degree, and 4: Masters/-Doctorate etc.

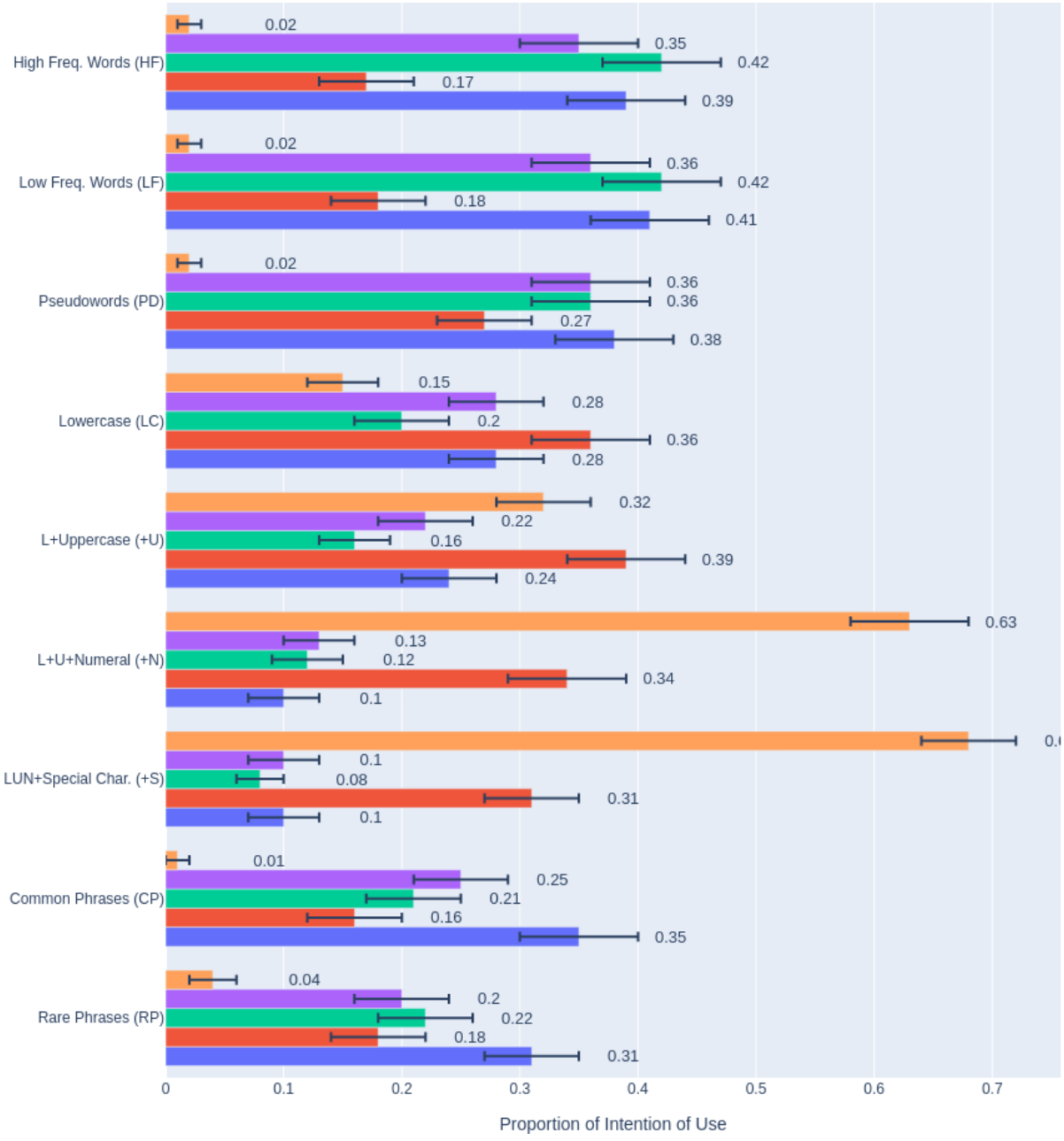


Figure 4.2: Mean ratings for Usability for each category. Key: (Red: Critical apps, Blue: Non-critical app, Orange: Critical services, Green: Time-bound services & Purple: Non-critical services)

Table 4.4: Spearman correlation between user-demographics and usability-(password-type) pairs

Demo-graphic	**Use Case(s)**	**Password Type(s)**	**Spearman Rho**; **p-value**
eng	webapp	heterogeneous	0.2449 ; **
	critapp	common	-0.3109 ; ***
	critapp	rare	-0.2412 ; *
	crit	HF	-0.3755 ; ***
	crit	LF	-0.369 ; ***
	app	HF	-0.2851 ; **
	crit	+S	0.2675 ; **
edu	time	passwords	-0.2663 ; **
	noncritapp	common	-0.2493 ; **
occ	critapp	heterogeneous	-0.2693 ; **
	webapp	+U	-0.2464 ; **
age	critapp	heterogeneous	-0.2747 ; **
	webapp	+U	-0.2807 ; **

Table 4.5: Spearman correlation between user-demographics and (PM-PS)-(password-type) pairs

Demo-graphic	**Rating Type**	**Password Type(s)**	**Spearman Rho; p-value**
eng	PM	all	0.2774 ; **
	PS	words	-0.2715 ; **
	PS	LF	-0.2601 ; **
occ	PM	LF	-0.2616 ; **
	PM	PD	-0.3173 ; **
age	PM	all	-0.2635 ; **
	PM	LF	-0.2771 ; **
	PM	PD	-0.3537 ; **

Table 4.6: t-test value experiment

Password-Type Pair	**t-value (PM)**	**t-value (PS)**
PD and +S	$10.1 \approx 10.0$	17.099
LC and +S	$9.955 \approx 10.0$	20.74
CP and RP	*12.17*	*4.638*
HF and RP	14.45	40.034

4.2.5 Discussion

In this section we explain the results obtained and present our inferences.

4.2.5.1 Perceived Memorability & Security

The basic results for PM and PS are consistent with the previous studies. However, we saw that the results from additional branching in the PM section underscored the need to consider different types of passphrases based on their structure and vocabulary.

- **Passphrase Experiment:** In order to determine an expected variation in PS ratings, we found pairs whose t-values (for PM) are close to the CP-RP pair (Table 4.6). These pairs were ordered by PM. We observe that the PS are also in ascending order, except for the CP-RP pair. It shows a much lower t-value for PS ($t = 4.39$, $p < .001$) compared to the t-values of the closest pairs ($t = 20.74$, $p < .001$ and $t = 40.03$, $p < .001$ in order).

This suggests that PS is not influenced by PM only. *Other factors also play a role*, otherwise, we would have seen a much larger variation between the PS ratings for CP and RP.

4.2.5.2 Usability in Specific Environments

We see that the *Critical Services* results show that PS is the major control variable for the usability of a password in a "critical service" and that the type of distribution in *Non-Critical Services* suggests a shift towards the use of word-like passwords, suggesting that PM becomes the deciding factor as the relative severity of the usage environment declines.

However as compared to Critical Services (not mobile applications), the usability distribution of *Critical Apps* is much more distributed across the categories. Since the criticality of the environments is equivalent, the preferred password-types are the same (***means** of distribution* of the two distributions show negligible differences), but the difference in the environment (web services vs. mobile applications) influences the general agreement on the preferred password-types (variance for critical-applications is much higher).

We can also see that results from *Time-Bound Services* and *Non-Critical Apps* suggest that participants in these use cases give a higher preference to retrievability than PS i.e. password types that have significantly higher PM than those preferred for critical use.

Finally, *Passphrases* not being considered usable across different use cases can be explained by observing that Common and Random passphrases fail due to their low PS ratings in use cases dominated by high PS passwords.

4.2.5.3 Demographics

We discuss a few significant correlations (as obtained from Tables 4.4 & 4.5) between some demographics and use cases, PM or PS, below:

We can see that the participants who are more proficient in "English" (as self reported), also have lower preference for meaningful words and phrases in critical scenarios. This is reinforced by the similarly low preference of high frequency passwords in even non-critical applications and the preference to use highly heterogeneous passwords (that don't have a meaning), in critical environments. Education levels give slightly ambiguous results where the participants show a low preference for generic password types in a time constrained scenario, this might be because of how difficult it is to retrieve such a string on a short notice. Finally, from Table 4.4 we can also see that "profession" and "age" show similar results, an older participant shows higher correlation with "simpler" passwords in most scenarios.

Moving onto the correlations with PM and PS (Table 4.5), we see that reported-proficiency in "English" seems to correlate with higher memorability ratings, while security ratings follow the opposite trend. This result seems to be aligned with the gradient of password types, ranging from heterogeneous strings to meaningful words and phrases, which allows participants of different language proficiency level to gauge the passwords accordingly. The "profession" and "age" demographics indicate an opposite trend compared to English-proficiency. This may be supported by the fact that a younger and/or working (not retired) participant will be both exposed to many password types, and would use more passwords on a daily basis.

4.2.6 Conclusion

In short, our results show that the negative correlation between PS and PM in passwords is strong for a large and varied demographic. Combined with previous studies, this also shows its true for different languages, experiences, etc. This also correlates with password choice in a few different use cases, e.g. Passphrases are not a popular choice for any use case, but the password `@?kUGS8o` was almost unanimously the best choice for a critical website because of its heterogeneity, and that Pseudo**words**, Low-frequency **words**, and High-frequency **words** were the most popular choices for use in Non-critical websites, mobile applications, and in a time-bound scenario.

A majority of participants disagreed with question 3 *"An easy to remember password is a safe password."* and agreed with question 10 *"A difficult to remember password is a more secure password."*.

This is consistent for the shorter passwords but *not for the passphrase categories*, as participants also acknowledged that CP are more memorable than RP but ultimately rated CP and RP similarly in terms of Security. From this, we conclude that we need to go beyond PM as the *only* influencing factor and pay more attention to the factors that could make passphrases appear safer to users. As shown, it could be due to "randomness" in the way the phrase was formed. This randomness in turn can be due to the *syntax* (if the words strung together make grammatical sense) and *semantics* (if the words make sense when they are put together in any order) of the passphrase.

Finally, the our results regarding passphrases show the need for such metacognition based studies on th and informed that people regularly use passwords, forget them, and store them in places that are not secure, etc. even after being aware of safe password setting practices.

4.2.7 Future Work and Limitations

Continuing the above section, we plan to expand this study in a more (psycho-)linguistic direction. We see that passphrases are influenced by randomness in some domain, which is ***not*** *heterogeneity of characters* but is more related to how the units of the phrase function with each other. There have been a handful of studies linking passphrases to **semantics** like the one on semantic noise [62] or "guided word choices" [63] and **syntax** like the one on entropy vs. syntax [64]. Even fewer study about the cognitive aspects like the one done on augmented cognition and cognitive load [65]. However, there is no current study on the association between these linguistic aspects of passphrases with metacognition/perceived memorability/perceived security. We plan to improve the work in this study and find out if such associations exist and influence password choice. Furthermore, studies have been conducted to discover other factors for passphrase utility, such as *pronounceability* [66] and whether *multilingual* passphrases can be strong as well [67]. We intend to keep these options open to including in our next metacognition experiments as mentioned above.

We also aim to find solutions to possible limitations. One of them was that the survey was conducted with a majority of users who have learned English as a second language, however comfortable they might have been. As the majority of the population did not consider passphrases to be useful, this study was possible. However, a study that focuses exclusively on passphrases should be careful with this problem. There is also the concern that the phrases used in the experiment could have a bias for some participants as they might have heard it before/used frequently in some scenario, thus perceiving it as more memorable vs. some user perceiving a common phrase as less memorable because the variant of English they use in their regions and societies might use a synonym for the same. Thus future work can include more randomness and a pilot to be sure that the phrases themselves are not biased to a subset of participants and a check could be done to see if the participants have similar linguistic and sociolinguistic backgrounds.

Finally, this experiment was based on Judgement as a major task to determine the correlation. A Generation task can lead to different results. Taking into account strong concerns about privacy and

limiting the user through nudges [68] there is the possibility of creating a completely different experimental framework.

4.3 Can pseudoword replacement help an LM classify Aphasia?

4.3.1 Introduction

Aphasia is a neurological linguistic disorder, characterized by language impairments that affect generating & understanding, the structure or/and meaning of language [69]. Contemporary works in Aphasia related classification, specifically on the AphasiaBank [70] dataset (detailed in subsection 4.3.2.1), predominantly apply statistical methods on raw as well as extracted features from text/ transcriptions. The underlying task is often detecting aphasic text or its severity, using features like closed-class function words, revisions, and repetitions passed to statistical pipelines such as K-Means clustering followed by classification or regression using Random Forest [71], or derivative features from ASR or clinical models [72] employing multiple ML techniques. Research like [73, 74, 75] also aim to determine the fluency of the text (binary or on a gradient) using features such as the WAB-R fluency scale, utterance length, speech rate (words per minute), etc. Alternatively, some methods predict the severity of the text using features extracted from key NLP discourse tasks [76], sentence predictability & flow. [69] encodes sudden change in topics using BERT in addition to Bi-gram perplexity to better represent instances of lapse in comprehension. While research like [77] offers a Python library to extract key CL inputs such as phonological, lexicosemantic, morphosyntactic, discourse, and pragmatic features for further analyses and research.

Since Aphasia type classification is a tricky task for language models and even BERT employing work like [69] shy away from the direct unfiltered application of the dataset to extract features, we wanted to find out what we could do, with out-of-vocabulary tokens (a lot of them phonologically well-formed, as they are transcriptions of actual spoken forms) like pseudowords, to help this data be more processable by LMs. We thus check if replacing pseudowords with "average word" predictions using context by another LM could impact the results of classification.

4.3.2 Experimental setup

Following, we setup experiments where we perform classification on AphasiaBank data (preprocessed as described in the next section) and also on a new, masked version of the dataset, where out-of-vocabulary items/pseudowords (except interjections) were replaced by word predictions of another language model (detailed in subsection 4.3.2.2).

4.3.2.1 Dataset

The Aphasiabank data is a rich, manually encoded dataset of interactions with Aphasic patients, where their utterances are transcribed in a uniform manner and are encoded with various psycholinguistic features [70]. Since it is an every growing, well-maintained, and the most widely-used English Aphasia data for research, we use the same. The distribution of Aphasia types by speakers are as follows:

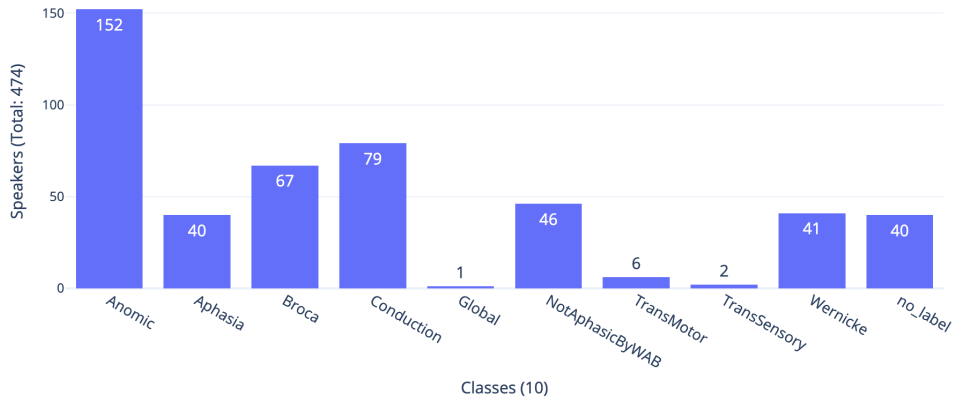


Figure 4.3: Distribution across classes by number of speakers

Since we want to run a simple classification, we make a simplifying assumption that each utterance by a speaker is classifiable into their respective aphasia types. However, this is not a necessity, and we also observe a huge amount of short well-formed replies like *yeah, ok* in the dataset, present across classes. We treat it as noise and reserve carefully removing well-formed and non-aphasic utterances for future work. Thus a sentence-wise distribution of the Aphasia types are as follows:

Figure 4.4: Distribution across classes by number of sentences

Finally, following [71], we only go with the 3 major and possibly easily differentiable classes of Anomic, Broca and Wernicke. Following are the key distinguishing features, speaker-wise, and sentence-wise distributions of the same:

Class	Features (consistent with AphasiaBank Transcripts)	Distribution	
		by Speakers	by Sentences
Anomic	Full of expressions of frustration like pauses, interjections etc.	195	20146
Broca	Speakers try to lexicalise thoughts shown by mispronouncing or partially pronouncing the full word.	100	10580
Wernicke	Majorly well-formed but is overall pretty difficult to follow in terms of meaning.	51	6988

Table 4.7: Final dataset distributions & features per class

4.3.2.2 Language Models used

The classification task on AphasiaBank is formulated as a multi-class classification with each class representing a type of aphasia evident from the input text. For this task, we fine-tune a DistilBERT model (`distilbert-base-uncased-finetuned-sst-2-english`) with a linear classifier head (with a 80-20 split of train & test data). This involves passing individual transcripts from AphasiaBank into the DistilBERT model, which generates a tensor representation (embedding) of the transcript. This embedding is then passed through a linear classification layer which produces an output tensor of size 3 i.e. the number of possible aphasia types for the scope of this classification task. This tensor is compared against the ground-truth binary label tensor with a cross entropy loss to tune the model. The prediction pipeline is summarized in Figure 4.5.

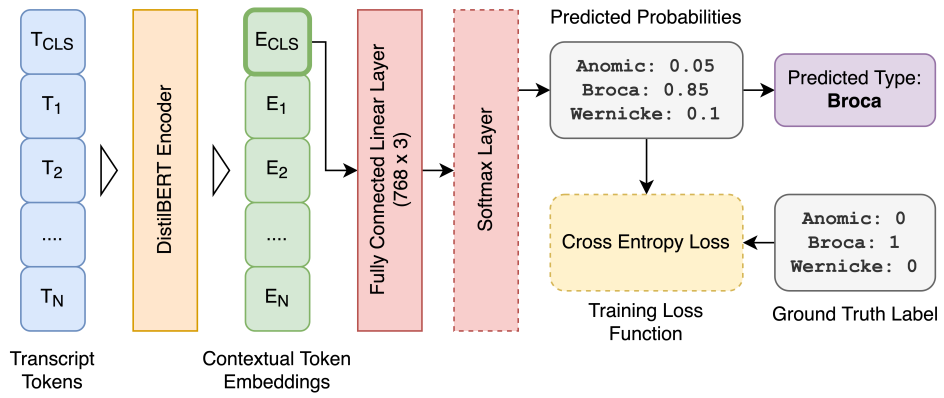


Figure 4.5: Aphasia type classification using DistilBERT. Specifically, the `[CLS]` token embedding from the encoder is used for classification by passing it though a linear layer to fetch class-specific normalized probabilities.

On the other hand, the masked-word prediction task was performed by running inference on a RoBERTa model (`roberta-base`). RoBERTa model was chosen for this task as it is pretrained solely towards the task of Masked Language Modeling (MLM) i.e. identical to the word prediction task we require. The output text is a modified variant of the original transcript, with a valid token predicted in place of each masked token. The MLM pipeline is summarized in Fugure 4.6. The extent

of mask-based replacement of pseudowords is varied across different runs, and the modified transcripts obtained are used as supervised datasets to fintune separate instances of DistilBERT for the aphasia-type classification as mentioned above.

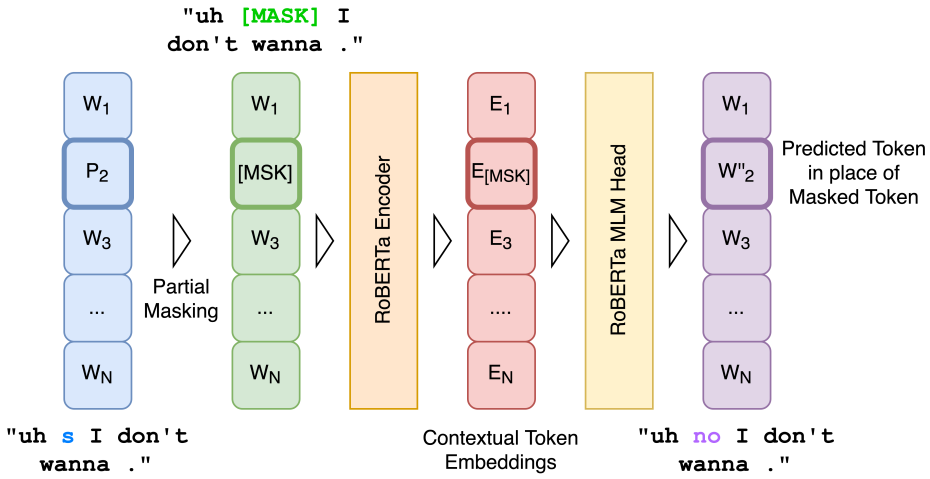


Figure 4.6: Masked Language Modeling using RoBERTa. The masked pseudowords are replaced by the most-likely words.

Both models were imported from and trained using the utilities provided by the HuggingFace suite [78] on a single NVIDIA T4 TensorCore GPU for 4–6 epochs.

4.3.3 Results

To understand the gradience in results, we ran classification on the unmasked dataset (max 0% masking per sentence), a dataset made of sentences with max 25% masking done, another with 50% max, and another with a max of 100%. Figure 4.7 shows examples from the data that were masked and how a gradation of masking was obtained. To make these different datasets, we picked a subset of the entire data which was masked to a maximum of a threshold percentage.

Unmasked sentence	# tokens	Masked sentence	# masks	Ratio
kimyi kini kingi dom kindom	5	[MASK] [MASK] [MASK] [MASK] [MASK]	5	1
um si sillin ned um absi sillin sindi sindull cinderella .	11	um [MASK] [MASK] [MASK] um [MASK] [MASK] [MASK] [MASK] cinderella .	7	0.6363636364
ramp lamp ra ra the jilk	6	ramp lamp [MASK] [MASK] the [MASK]	3	0.5
and cerendel cinderella would like wanted to go .	9	and [MASK] cinderella would like wanted to go .	1	0.1111111111
and she was very happy .	6	and she was very happy .	0	0

Figure 4.7: Different ratios of masking done according to the number of pseudowords found.

Type of Metric	Masking Ratio	Anomic			Broca			Wernicke		
		P	R	F1	P	R	F1	P	R	F1
Sentence	0	0.59	0.67	0.62	0.37	0.35	0.36	0.34	0.23	0.27
	0.25	0.59	0.69	0.64	0.39	0.34	0.36	0.33	0.24	0.28
	0.5	0.58	0.67	0.62	0.39	0.32	0.35	0.32	0.25	0.28
	1	0.58	0.68	0.62	0.36	0.31	0.33	0.34	0.24	0.28
Override	0	0.56	0.98	0.72	0.85	0.11	0.19	0.8	0.11	0.2
	0.25	0.56	0.99	0.72	0.74	0.1	0.17	0.93	0.1	0.18
	0.5	0.56	0.99	0.72	0.8	0.16	0.26	0.72	0.06	0.11
	1	0.55	0.98	0.7	0.56	0.07	0.12	0.79	0.02	0.03
Speaker	0	0.62	0.96	0.75	0.62	0.11	0.19	0.5	0.1	0.17
	0.25	0.58	0.97	0.73	0.61	0.11	0.19	0.75	0.06	0.11
	0.5	0.58	0.99	0.73	0.74	0.14	0.23	1	0.04	0.07
	1	0.6	0.97	0.74	0.38	0.06	0.1	0.5	0.04	0.07

Figure 4.8: Table of all the metrics across the gradation of experiments. Here, the average precision (P), recall (R), and F1 scores are calculated between the predicted and actual aphasia type of the input transcripts. The input transcripts vary along an increasing proportion of pseudoword replacement (0.0, 0.25, 0.5, 1.0). The class-wise P, R, and F1 for the major aphasia classes are reported accordingly.

Note that since the input dataset made an assumption that most utterances by a participant showed features of their respective Aphasia types, we decided to analyse our results in 3 different ways:

1. Sentence: sentence-wise comparison of actual and predicted labels.

2. Speaker: for each speaker, our ground truth was the label participant was assigned in the AphasiaBank transcripts, and predicted labels were the majority labels from the sentences under each speaker, for each speaker.

3. Override: for each speaker, we changed the predicted labels of the sentences under that speaker to the majority label obtained for them, then did a sentence-wise comparison.

Table (captured by Figure 4.8) summarise the same.

The results (visualised in Figure 4.9) show us that there is consistent decrease in performance of Broca sentence-wise classification (and an overall decrease across other metrics as well) while Anomic remained stable across all metrics (possibly because of the bias in data proportions.), we also see that Wernicke (the lowest proportion of data) was unaffected on sentence-wise classification.

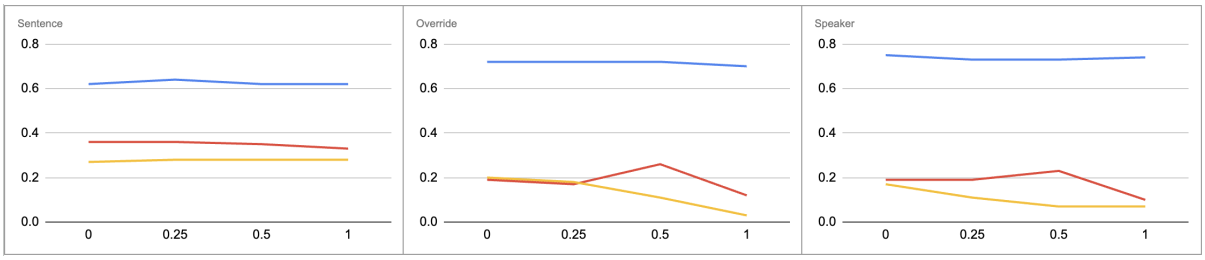


Figure 4.9: Visualisation of F1 scores from Fig. 4.8 as max proportion of masking allowed, is increased (x-axis). Blue:Anomic, Red:Broca, and Yellow:Wernicke.

Figure 4.10 displays some examples that help us explain what possibly happened on replacing pseudowords and how on the sentencewise-classification task and why Broca was probably the only category that kept falling as the masking increased:

Participant ID	Original Transcript	Masked Transcript	Replaced Transcript	Ground Truth	Predicted Label
whiteside18a	y you guys had uh not had not th yet started your semester so	[MASK] you guys had uh not had not [MASK] yet started your semester so	So you guys had uh not had not not yet started your semester so	Broca	Broca
williamson06a	uh s i don't wanna .	uh [MASK] i don't wanna .	uh no i don't wanna .	Broca	Wernicke
whiteside06a	and he and i said w this must be the thing with them .	and he and i said [MASK] this must be the thing with them .	and he and i said , this must be the thing with them .	Wernicke	Wernicke
MSU08b	but it's i think it's very s tlli slowly .	but it's i think it's very [MASK] [MASK] slowly .	but it's i think it's very slowly very slowly .	Broca	Anomic
BU03a	um um she eventually got help from the uh godbu bu godmother .	um um she eventually got help from the uh [MASK] [MASK] godmother .	um um she eventually got help from the uh her fairy godmother .	Broca	Wernicke
kurland02g	and she was cinderella was happy to prince .	and she was [MASK] was happy to prince .	and she was she was happy to prince .	Wernicke	Broca
UNH05a	well uh when i f-f-firfirst went there uh i didn't know many people .	well uh when i [MASK] went there uh i didn't know many people .	well uh when i first went there uh i didn't know many people .	Anomic	Wernicke

Figure 4.10: Some examples showing results of classification on the masked transcripts. The original transcripts are obtained from the AphasiaBank dataset, from which a varying proportion (0.0, 0.25, 0.5, 1.0) of pseudowords are masked to produce the masked transcripts. The masked transcripts are passed through a RoBERTa model using the MLM task to generate the replaced transcripts, which are used to finetune respective DistilBERT models for the aphasia-type classification.

1. The Good (greens): Although both of these are not correct predictions, they are marked green as masking turned out well. For the first sentence, replacing a *th* with a *not* made it the transcript 'look' more Broca-like, as the word repetition is with meaningful part/whole of the word. Similarly, for the second one, replacing made it very structured and thus Wernicke-like. Thus the masking is being done well, across transcripts.

2. The Accidentally OK (yellow): There are some results of masking (like this one) where predictions match up with the ground truth, even though replacement changed the structure a lot.

3. The Bad (reds): All of these are mis-classified examples, showing examples of various ways in which the masking was bad:

   (a) Participant MSU08b: here the participant possibly tried to form the word *still* but the masking model replaced that with *very slowly*. Although this still looks like a Broca-like example, the classification model predicts Anomic.

   (b) Participant BU03a: here we see a classic example of Broca where *godbu* is a part of the word *godmother* (albeit the nasal 'm' is de-nasalised to 'b'), and it would be an interesting experiment to eventually see if tokenisers/sub-word-embeddings for the classification model could learn this as a feature for Broca, but the masking replaced the pseudowords *godbu* & *bu* with *her fairy* which is the perfect replacement and thus makes it a non-Broca-like (in fact Wernicke-like) and a grammatically accurate utterance.

   (c) Participant kurland02g: this example shows that we have to be very careful with what is considered as a pseudoword. Since a basic English lexicon (for general NLP usecases) won't have a lot Propoer nouns like *cinderella* but Aphasia prompts might (as they are generally easy to recognise situations, like children's stories), it is important to not replace them.

   (d) Participant UNH05a: Similar to BU03a above, a phonemic repetition of the first phoneme of the word, while trying to find it, was masked as it now made *fffirst* a pseudoword. This eventually changed back to *first* but now a more well-formed sentence.

In summary, while the models might not yet be learning how to extract information from pseudowords by either searching for the closest word, or by trying to see if they are a modified and/or incomplete version of a close word in the neighbourhood of the sentence, it is still important to preserve them. This is as deleting them can either make an incoherent sentence more incoherent to the system or replacing them (as seen above) with an expected token automatically, could make it more coherent than it was meant to be. As an alternate approach understanding how pseudowords work and making NLP models capable of handling them and gathering more information from them, could have helped solve a lot of the issues listed above!

Chapter 5

Conclusion

5.1 Summary

The first exploration concluded with a possibly language-agnostic pipeline, which generated Hindi & English pseudowords. Built after the principles of language modelling and subsequent generation, this pipeline (PseudRNN) uses an LSTM trained on sequences of phonemes in a lexicon to generate more of them. Out of which a simple subtraction of word sequences (by cross-checking the lexicon) led us to pseudowords in both languages. We also check if only orthographical sequences would be sufficient in generating pseudowords language-agnostically and found out systematic issues with abugida style scripts like Devanagari. This leads us to the next chapter of finding ways to evaluate & compare pseudowords & their generation methods.

The second exploration describes how to design a behavioral experiment to collect human judgements on a pseudoword's wordlikeness and how to evaluate the results from such a task. It adds on to the existing work and also shows ways to make that more langauge-agnostic. This is done by following the existing work and expanding on them and recording the results on current works' generated pseudowords for future comparison by another work. While it encourages more work in the area, it also shows results obtained on comparing both the PseudRNN outputs on existing metrics. It showed that although there is a lot to be fixed in the existing metric framework, lingusitically, Hindi pseudowords generated by PseudRNN are comparable, while English ones are ambiguous with respect to the older metrics. This chapter then concludes by making public the first Hindi pseudoword dataset and also enriches it with features used for psycholinguistic datasets of words & pseudowords.

Finally, the study around the perception of a passwords' security negatively correlated with its perceived memorabilty with crucial arguments drawn from correlations with pseudowords in the middle, as a part of the stimuli of sample passwords. Thus apart from being useful as passwords in day-to-day use cases, this study shows how pseudowords are important to specific kinds of research as well. This is then followed up by a more direct analysis of their impact on Aphasia classification by a language model, where replacing them opened up a Pandora's box of issues apart from negatively affecting the

classification of certain types of Aphasia which are passively modelled by an LM as types which either use pseudowords (Broca) or don't (Wernicke).

We thus find that pseudowords can be efficiently generated and evaluated better with ways that cut across languages. We also see that pseudowords are a natural part of gradation between meaningful and unpronounceable side of a language (its full set of strings, which can be generated using the phonemes in that language) that is of import in studies not limited to psycholinguistics and are irreplaceable features of some like those on Aphasic text.

5.2 Limitations & Future Work

This work is a first step towards consolidating current work in computational phonological modelling, pseudoword generation, and pseudoword evaluation. It is thus limited in scope in terms of more research untouched in the respective domain or recent research that might have come up. Although it aims to contribute a way forward inspired from these works, it will be important to keep adding on more research left out or new research to this exploration in terms of improving generation, finding more features to learn or fallbacks to safegaurd against or to add more metrics for a better evaluation.

This work also aims to be cross-lingual, but is only run fully on Hindi and for comparison to previous metrics, on English. For resource creation in the domain and for understanding if the model's prowess cross languages, the next step would be run the generation + evaluation pipeline for more Indian languages and others. A limitation that pops up right away is the need for experts in these languages needed to bridge the phonology and orthography + and to rate on suitability metrics. A short term plan for this could be to work on languages with these resources and a longer term plan would be to adapt tools in the domain to grapheme to phoneme conversion to replace the need for humans at some parts of these pipelines. In terms of resources, a systematic study to understand how minimal and what kind of data can be optimal for PseudRNN to efficiently produce quality pseudowords, needs to be conducted as well.

Finally, Chapters 2 & 3 are a unit and produce Soodkosh, whereas the applications & impact shown in Chapter 4 is an independent explorations of general use of Psuedowords beyond psycholinguistic tasks like Lexical Decision task, it is important to note that such studies into the why of pseudowords is important to simulate more interest in the field as well as solve some roadblocks for NLP tasks like processing incoherent text. Thus more research in pseudowords across languages on Aphasic text or otherwise can be of help to NLP and could open more avenues for NLP to help areas-in-need like models mimicking brains that have Aphasia affected regions, explaining a models' semantic understanding of natural text, or even testing an NLP model's robustness etc.

Appendix A

Pseudowords & How to Generatuce some

A.1 Hyperparameters

Hyperparameter	Value (specified if different from default)
Input Embedding Size	34 (default: 200)
Hidden Embedding Size	200
Layer Count	2
Input Sequence Length	10 (default: 35)
Dropout	0.2
Learning Rate	20
Gradient Clipping	0.25
Max Epochs	40
Batch Size	20
Initialization Seed	1111

Table A.1: Model and training hyperparameters of the WordRNN model used.

Appendix B

Evaluadating Pseudowords

Ethics Statement (for pseudoword generation & evaluation)

As the author of the work, I don't anticipate any negative or harmful ethical impacts of this research. We started out to facilitate neurologists with Indian Language psycholinguistic batteries to better understand the linguistic disabilities of people and to help them in a more informed manner. Pseudowords turned out to be an artefact that could not be borrowed from more resource-rich languages. This is what motivated the work and is also why the generation, evaluation, and the resulting dataset are made public. Collaborative research & innovation in this area which contribute more psycholinguistic tools which are accessible to other low-resourced languages would be beneficial to society.

B.1 Appendix: Consent

Below is the text for Consent that the Participants read and accepted before continuing to the main experiment:

*Please carefully read the following information and click on **"Yes, I agree"** below if you consent to volunteer for the survey:*

***Risks:** There are no anticipated risks to the participants.*

***Confidentiality:** We will ask you to fill in your roll number to connect survey data to the experiment. We will anonymize all data we collate finally. Thus your participation will remain confidential and a fresh, explicit consent will be taken if any personal detail needs to be used for further analysis.*

***Participation and Withdrawal:** Participation is voluntary and choose to leave the survey at any point in time pre-experiment. You can also inform the researchers (no questions will be asked) to mark your data as confidential at any point in time, even after the research concludes. Once marked confidential/withdrawn, the information will not be used or disclosed.*

P.S. This data will not be used for any commercial purposes.

*By clicking **"Yes, I agree"**, you confirm that you have been satisfactorily explained the details of the procedure, the type and safety of data being collected via this study. Thus, you hereby consent to participate in the study.*

B.2 Appendix: Participant Demographics

Participant Demographic or Language Proficiency	Mean	Standard Deviation
Age	20.205	1.44
Reading	4.091	0.984
Speaking	4.386	0.689
Understanding	4.5	0.699
Say the days of the week	3.932	1.246
Name the months in a year	3.182	1.589
Use basic numbers (for asking prices or quantities etc.)	4.045	0.939
Name common fruits/vegetables/dishes	4.114	0.784
Retell a sequence of events/ recite a story	4.705	0.594
While talking to family older than you	4.5	1
While talking to family younger than you	4.477	1.045
While talking to friends and neighbours	4.727	0.499
While watching reels/series/movies	4.659	0.645
While texting friends	4.409	0.658
While arguing when it gets intense/ when you are angry	4.591	0.693
For writing a formal email or filling up a form	2.614	1.166

Table B.1: Demographic details & Self reported language proficiency ratings of the participants.

Please refer to table B.1 for the exhaustive list.

B.3 Appendix: Questionnaire adapted from LUQ

We collect the language proficiency and demographics via a Google Form as a pre-survey to the experiment for each participant. The below list details the same:

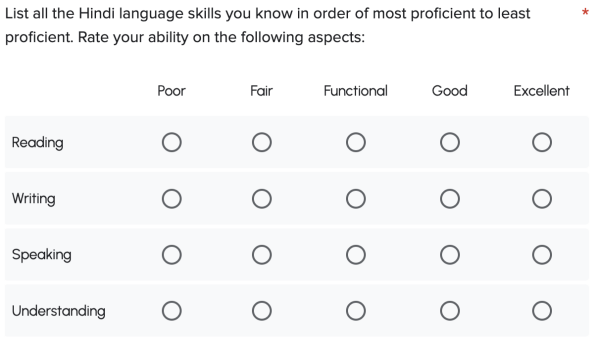


Figure B.1: Screenshot of the Google Form question for self-rating language proficiency.

- After Age & Gender we ask participants to rate self-proficiency on Hindi & English (Fig. B.1).
- We then ask participants to self report language proficiency in specific contexts that ranged from formal, informal, seniority difference etc. There was a block of instruction, followed by the question, and a scale to select response on, for different languages (like Fig. B.2).

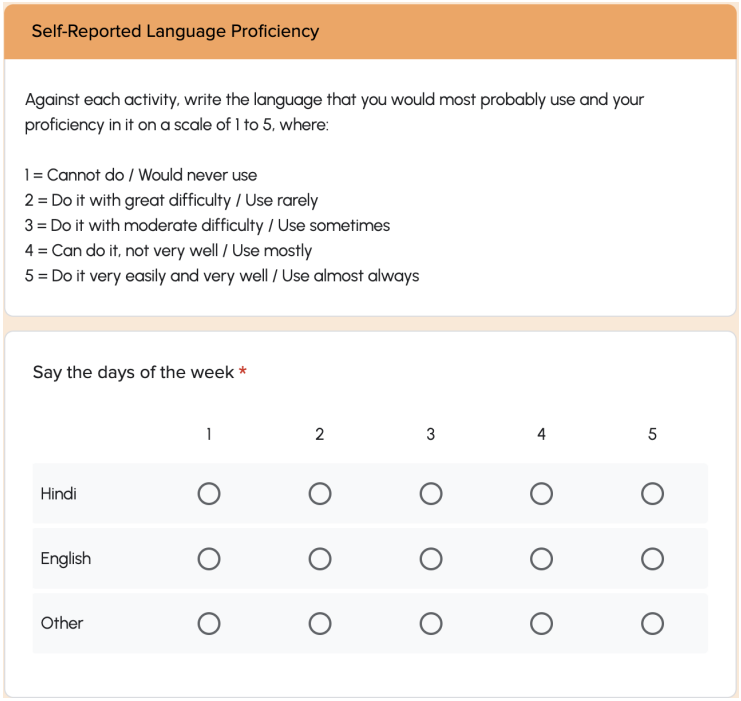


Figure B.2: Screenshot of the Google Form question displaying instructions and asking the participant to self-rate language proficiency for different languages, in a specific context.

Related Publications

1. Choudhary, M., Srivatsa, K., Upadhyay, I., & Srivastava, P. (2021). Is convenient secure? Exploring the impact of metacognitive beliefs in password selection. *Proceedings of the Annual Meeting of the Cognitive Science Society*, 43. Retrieved from https://escholarship.org/uc/item/7v1654s9

2. Choudhary, M., Nikhil, E., Menon, A., Srivatsa, K., Surampudi, B., Sharma, D. (2023). PseudRNN: Automatic Generation of Pseudowords and Evaluation Strategies, *Association of Computational Linguistics*, Under-Review.

Bibliography

[1] Jemma Lynette König, Andreea S. Calude, and Averil Coxhead. Using character-grams to automatically generate pseudowords and how to evaluate them. *Applied Linguistics*, 41(6):878–900, 2019.

[2] Jeremy M. Needle, Janet B. Pierrehumbert, and Jennifer B. Hay. Phonotactic and morphological effects in the acceptability of pseudowords. In Andrea D. Sims, Adam Ussishkin, Jeff Parker, and Samantha Wray, editors, *Morphological Diversity and Linguistic Cognition*, page 79–112. Cambridge University Press, 2022.

[3] David A. Balota, Melvin J. Yap, Keith A. Hutchison, Michael J. Cortese, Brett Kessler, Bjorn Loftis, James H. Neely, Douglas L. Nelson, Greg B. Simpson, and Rebecca Treiman. The english lexicon project. *Behavior Research Methods*, 39(3):445–459, Aug 2007.

[4] William Milberg and Sheila E. Blumstein. Lexical decision and aphasia: Evidence for semantic processing. *Brain and Language*, 14(2):371–385, 1981.

[5] Taelin Karidi, Yichu Zhou, Nathan Schneider, Omri Abend, and Vivek Srikumar. Putting words in BERT's mouth: Navigating contextualized vector spaces with pseudowords. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10300–10313, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.

[6] Rowan Hall Maudslay and Ryan Cotterell. Do syntactic probes probe syntax? experiments with jabberwocky probing. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 124–131, Online, June 2021. Association for Computational Linguistics.

[7] Emmanuel Keuleers and Marc Brysbaert. Wuggy: A multilingual pseudoword generator. *Behavior Research Methods*, 42(3):627–633, Aug 2010.

[8] Kamil K Imbir, Tomasz Spustek, and Jarosław Żygierewicz. Polish pseudo-words list: dataset of 3023 stimuli with competent judges' ratings. *Front. Psychol.*, 6:1395, September 2015.

[9] Boris New, Christophe Pallier, Marc Brysbaert, and Ludovic Ferrand. Lexique 2 : A new french lexical database. *Behavior Research Methods, Instruments, & Computers*, 36(3):516–524, Aug 2004.

[10] Ludovic Ferrand, Boris New, Marc Brysbaert, Emmanuel Keuleers, Patrick Bonin, Alain Méot, Maria Augustinova, and Christophe Pallier. The French Lexicon Project: Lexical decision data for 38,840 French words and 38,840 pseudowords. *Behavior Research Methods*, 42(2):488–496, May 2010.

[11] Wouter Duyck, Timothy Desmet, Lieven P. C. Verbeke, and Marc Brysbaert. Wordgen: A tool for word selection and nonword generation in dutch, english, german, and french. *Behavior Research Methods, Instruments, & Computers*, 36(3):488–499, Aug 2004.

[12] Yousri Marzouki, Sara Abdulaziz Al-Otaibi, Muneera Tariq Al-Tamimi, and Ali Idrissi. Can the word superiority effect be modulated by serial position and prosodic structure? *Frontiers in Psychology*, 13:915666, Aug 2022.

[13] Yongeun Lee. A study of effects of frequency of phoneme sequences on wordlikeness judgments of korean nonce words. *Journal of English Language and Literature*, 51(2):215–234, 2009.

[14] Connor Mayer and Max Nelson. Phonotactic learning with neural language models. In *Proceedings of the Society for Computation in Linguistics 2020*, pages 291–301, New York, New York, January 2020. Association for Computational Linguistics.

[15] Joseph H. Greenberg and James J. Jenkins. Studies in the Psychological Correlates of the Sound System of American English. *WORD*, 20(2):157–177, January 1964.

[16] Manjari Ohala. *Aspects Of Hindi Phonology*. Motilal Banarsidass Publishers, 1983.

[17] Rastle, Kathleen and Harrington, Jonathan and Coltheart, Max. 358,534 nonwords: The ARC Nonword Database. *The Quarterly Journal of Experimental Psychology Section A*, 55:1339–1362, 2002.

[18] Isabelle Dautriche, Kyle Mahowald, Edward Gibson, Anne Christophe, and Steven T. Piantadosi. Words cluster phonetically beyond phonotactic regularities. *Cognition*, 163:128–145, 2017.

[19] Sean Trott and Benjamin Bergen. Languages are efficient, but for whom? *Cognition*, 225:(Article ID) 105094, August 2022.

[20] Sean Trott and Benjamin Bergen. Why do human languages have homophones? *Cognition*, 205:(Article ID) 104449, December 2020.

[21] Spencer Caplan, Jordan Kodner, and Charles Yang. Miller's monkey updated: Communicative efficiency and the statistics of words in natural language. *Cognition*, 205:(Article ID) 104466, December 2020.

[22] Tiago Pimentel, Clara Meister, Simone Teufel, and Ryan Cotterell. On homophony and rényi entropy. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8284–8293, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.

[23] Richard Futrell, Adam Albright, Peter Graff, and Timothy J. O'Donnell. A generative model of phonotactics. *Transactions of the Association for Computational Linguistics*, 5:73–86, 2017.

[24] Nicole Mirea and Klinton Bicknell. Using LSTMs to assess the obligatoriness of phonological distinctive features for phonotactic learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1595–1605, Florence, Italy, July 2019. Association for Computational Linguistics.

[25] Sidsel Boldsen, Manex Agirrezabal, and Nora Hollenstein. Interpreting character embeddings with perceptual representations: The case of shape, sound, and color. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6819–6836, Dublin, Ireland, May 2022. Association for Computational Linguistics.

[26] Pramod Pandey. Phonology–orthography interface in devanāgarī for hindi. *Constraints on Spelling Changes*, 10(2):145–162, Dec 2007.

[27] Miikka Silfverberg, Lingshuang Jack Mao, and Mans Hulden. Sound analogies with phoneme embeddings. In *Proceedings of the Society for Computation in Linguistics (SCiL) 2018*, pages 136–144, 2018.

[28] Jackson L. Lee, Lucas F.E. Ashby, M. Elizabeth Garza, Yeonju Lee-Sikka, Sean Miller, Alan Wong, Arya D. McCarthy, and Kyle Gorman. Massively multilingual pronunciation modeling with WikiPron. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4223–4228, Marseille, France, May 2020. European Language Resources Association.

[29] Jimmy Wales. Wiktionary, the free dictionary. `https://en.wiktionary.org/`, 12 2002.

[30] Benoit Steiner, Zachary DeVito, Soumith Chintala, Sam Gross, Adam Paszke, Francisco Massa, Adam Lerer, Trevor Killeen, Gregory Chanan, Zeming Lin, Edward Yang, Alban Desmaison, Andreas Kopf, Alykhan Tejani, Andreas Kopf, Zachary DeVito, James Bradbury, Martin Raison, Luca Antiga, Martin Raison, Natalia Gimelshein, Sasank Chilamkurthy, Lu Fang, Trevor Killeen, Lu Fang, and Junjie Bai. Pytorch: An imperative style, high-performance deep learning library. *Neural Information Processing Systems*, 2019.

[31] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, November 1997.

[32] Grzegorz Kondrak. N-gram similarity and distance. In Mariano Consens and Gonzalo Navarro, editors, *String Processing and Information Retrieval*, pages 115–126, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

[33] Matthew Snover, Bonnie Dorr, Rich Schwartz, Linnea Micciulla, and John Makhoul. A study of translation edit rate with targeted human annotation. In *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas: Technical Papers*, pages 223–231, Cambridge, Massachusetts, USA, August 8-12 2006. Association for Machine Translation in the Americas.

[34] Garrett Nicolai and Grzegorz Kondrak. English orthography is not "close to optimal". In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 537–545, Denver, Colorado, May–June 2015. Association for Computational Linguistics.

[35] Shuxiao Gong and Jie Zhang. Gradient acceptability in mandarin nonword judgment. *Proceedings of the Annual Meetings on Phonology*, 7, 2020.

[36] Youngah Do and Ryan Ka Yau Lai. Incorporating tone in the modelling of wordlikeness judgements. *Phonology*, 38(3):535–535, August 2021.

[37] Adam Albright. Feature-based generalisation as a source of gradient acceptability. *Phonology*, 26(1):9–41, May 2009.

[38] Paul. Meara and Swansea University College's Centre for Applied Language Studies. Wales University. *EFL Vocabulary Tests*. Distributed by ERIC Clearinghouse [Washington, D.C.], 1992.

[39] Enes Avcu, Olivia Newman, Seppo P. Ahlfors, and David W. Gow. Neural evidence suggests phonological acceptability judgments reflect similarity, not constraint evaluation. *Cognition*, 230:(Article ID) 105322, January 2023.

[40] Kyle Gorman. Categorical and gradient aspects of wordlikeness. *Proceedings of the 43rd Annual Meeting of the North East Linguistics Society*, 2013.

[41] Dominique Makowski and Léo Dutriaux. Neuropsydia.py: A Python Module for Creating Experiments, Tasks and Questionnaires. *The Journal of Open Source Software*, 2(19):(Article ID) 259, November 2017.

[42] D Vasanta, Suvarna Alladi, J. Sireesha, and Bapi Surampudi. Language choice and language use patterns among telugu-hindi/urdu-english speakers in hyderabad, india. *International Conference on Language, Society, and Culture in Asian Contexts (LSCAC) Proceedings*, pages 57–67, 01 2010.

[43] Michael S. Vitevitch, Paul A. Luce, Jan Charles-Luce, and David Kemmerer. Phonotactics and Syllable Stress: Implications for the Processing of Spoken Nonsense Words. *Language and Speech*, 40(1):47–62, January 1997.

[44] Enrico Ripamonti, Claudio Luzzatti, Pierluigi Zoccolotti, and Daniela Traficante. Word and pseudoword superiority effects: Evidence from a shallow orthography language. *Quarterly Journal of Experimental Psychology*, 71(9):1911–1920, September 2018.

[45] Anthi Revithiadou, Dimitra Ioannou, Maria Chatzinikolaou, and Katerina Aivazoglou. Constructing pseudowords for experimental research: Problems and solutions. *Selected papers on theoretical and applied linguistics*, 21:356–365, 2016.

[46] James White, Faith Chiu, and Faith Chiu. Disentangling phonological well-formedness and attestedness: An erp study of onset clusters in english. *Acta Linguistica Academica*, 64(4):513–537, 2017.

[47] Ark Verma, Vivek Sikarwar, Himanshu Yadav, Ranjith Jaganathan, and Pawan Kumar. Shabd: A psycholinguistic database for Hindi. *Behavior Research Methods*, 54(2):830–844, August 2021.

[48] JASP Team. JASP (Version 0.16.4)[Computer software]. `https://jasp-stats.org/`, 2022.

[49] Stefan A. Frisch, Nathan R. Large, and David B. Pisoni. Perception of Wordlikeness: Effects of Segment Probability and Length on the Processing of Nonwords. *Journal of Memory and Language*, 42(4):481–496, May 2000.

[50] Todd M. Bailey and Ulrike Hahn. Determinants of Wordlikeness: Phonotactics or Lexical Neighborhoods? *Journal of Memory and Language*, 44(4):568–591, May 2001.

[51] Joakim Kävrestad, Markus Lennartsson, Marcus Birath, and Marcus Nohlberg. Constructing secure and memorable passwords. *Information & Computer Security*, 28(5):701–717, June 2020.

[52] Thomas O. Nelson. Metamemory: A theoretical framework and new findings. *Psychology of Learning and Motivation*, 26:125–173, 1990.

[53] Luna, Karlos. If it is easy to remember, then it is not secure: Metacognitive beliefs affect password selection. *Applied Cognitive Psychology*, 33:744–758, 2019.

[54] Stoet, Gijsbert. PsyToolkit: A software package for programming psychological experiments using Linux. *Behavior Research Methods*, 42:1096–1104, 2010.

[55] Stoet, Gijsbert. PsyToolkit. *Teaching of Psychology*, 44:24–31, 2016.

[56] Kaspersky. Kaspersky: Secure Password Check. `www.password.kaspersky.com/`, 2019.

[57] My1Login. Password Strength Test - My1Login. `www.my1login.com/resources/password-strength-test/`, 2019.

[58] Hearn, Mike and Wheeler, Dan. Use a Passphrase. `www.useapassphrase.com/`.

[59] Nahum, Mor and Vleet, Thomas M. Van and Sohal, Vikaas S. and Mirzabekov, Julie J. and Rao, Vikram R. and Wallace, Deanna L. and Lee, Morgan B. and Dawes, Heather and Stark-Inbar, Alit and Jordan, Joshua Thomas and Biagianti, Bruno and Merzenich, Michael and Chang, Edward F. Immediate Mood Scaler: Tracking Symptoms of Depression and Anxiety Using a Novel Mobile Mood Scale. *JMIR mHealth and uHealth*, 5:e44, 2017.

[60] Loutfi, Ijlal and Jøsang, Audun. Passwords are not always stronger on the other side of the fence. *Workshop on Usable Security*, 2015.

[61] Michael Stainbrook and Nicholas Caporusso. Comparative evaluation of security and convenience trade-offs in password generation aiding systems. *International Conference on Applied Human Factors and Ergonomics*, pages 87–96, June 2019.

[62] Lee, Kok-Wah and Ewe, Hong-Tat. Passphrase with Semantic Noises and a Proof on Its Higher Information Rate. *2007 International Conference on Computational Intelligence and Security Workshops (CISW 2007)*, pages 652–55, 2007.

[63] Nikola K. Blanchard, Clément Malaingre, and Ted Selker. Improving security and usability of passphrases with guided word choice. In *Proceedings of the 34th Annual Computer Security Applications Conference*, pages 723–732, San Juan PR USA, December 2018. ACM.

[64] Panferov, Eugene. An Observation About Passphrases: Syntax vs Entropy. *arXiv:1603.06133 [cs]*, 2016.

[65] Lila A. Loos, Michael-Brian Ogawa, and Martha E. Crosby. Impedances of memorable passphrase design on augmented cognition. In Dylan D. Schmorrow and Cali M. Fidopiastis, editors, *Augmented Cognition*, pages 84–92, Cham, 2019. Springer International Publishing.

[66] Andrew M. White, Katherine Shaw, Fabian Monrose, and Elliott Moreton. Isn't that Fantabulous: Security, Linguistic and Usability Challenges of Pronounceable Tokens. In *Proceedings of the 2014 New Security Paradigms Workshop*, pages 25–38, Victoria British Columbia Canada, September 2014. ACM.

[67] Pardon Blessings Maoneke, Stephen Flowerday, and Naomi Isabirye. Evaluating the strength of a multilingual passphrase policy. *Computers & Security*, 92:(Article ID) 101746, May 2020.

[68] Karen Renaud and Verena Zimmermann. Nudging folks towards stronger password choices: providing certainty is the key. *Behavioural Public Policy*, 3(2):228–258, 2019.

[69] Katherine Ann Dunfield and Günter Neumann. Automatic quantitative prediction of severity in fluent aphasia using sentence representation similarity. In *LREC 2020 Language Resources and Evaluation Conference 11-16 May 2020*, page 24, 2020.

[70] Brian MacWhinney and Davida Fromm. Language sample analysis with talkbank: An update and review. *Frontiers in Communication*, 7:(Article ID) 865498, 2022.

[71] Davida Fromm, Joel Greenhouse, Mitchell Pudil, Yichun Shi, and Brian MacWhinney. Enhancing the classification of aphasia: A statistical analysis using connected speech. *Aphasiology*, 36(12):1492–1519, September 2021.

[72] Gerasimos Chatzoudis, Manos Plitsis, Spyridoula Stamouli, Athanasia–Lida Dimou, Nassos Katsamanis, and Vassilis Katsouros. Zero-Shot Cross-lingual Aphasia Detection using Automatic Speech Recognition. In *Interspeech 2022*, pages 2178–2182. ISCA, September 2022.

[73] Sharice Clough and Jean K. Gordon. Fluent or nonfluent? part a. underlying contributors to categorical classifications of fluency in aphasia. *Aphasiology*, 34(5):515–539, 2020.

[74] Jean K. Gordon and Sharice Clough. How fluent? part b. underlying contributors to continuous measures of fluency in aphasia. *Aphasiology*, 34(5):643–663, 2020.

[75] Jean K Gordon and Sharice Clough. How do clinicians judge fluency in aphasia? *J Speech Lang Hear Res*, 65(4):1521–1542, March 2022.

[76] Marjory Day, Rupam Kumar Dey, Matthew Baucum, Eun Jin Paek, Hyejin Park, and Anahita Khojandi. Predicting severity in people with aphasia: A natural language processing and machine learning approach. In *2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, pages 2299–2302, 2021.

[77] Abhishek Shivkumar, Jack Weston, Raphael Lenain, and Emil Fristed. BlaBla: Linguistic Feature Extraction for Clinical Analysis in Multiple Languages. In *Interspeech 2020*, pages 2542–2546. ISCA, October 2020.

[78] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online, October 2020. Association for Computational Linguistics.