You Don't Have Time to Read This: An Exploration of Document Reading Time Prediction

Orion Weller,[1] **Jordan Hildebrandt,**[1] **Ilya Reznik,**[2] **Chris Challis,**[2]
E. Shannon Tass,[1] **Quinn O. Snell,**[1] **Kevin Seppi**[1]
[1]Brigham Young University
[2]Adobe
orionw@byu.edu

Abstract

Predicting reading time has been a subject of much previous work, focusing on how different words affect human processing, measured by reading time. However, previous work has dealt with a limited number of participants as well as word level only predictions (i.e. predicting the time to read a single word). We seek to extend these works by examining whether or not document level predictions are effective, given additional information such as subject matter, font characteristics, and readability metrics. We perform a novel experiment to examine how different features of text contribute to the time it takes to read, distributing and collecting data from over a thousand participants. We then employ a large number of machine learning methods to predict a user's reading time. We find that despite extensive research showing that word level reading time can be most effectively predicted by neural networks, larger scale text can be easily and most accurately predicted by one factor, the number of words.

1 Introduction

Understanding how we read and process text has proven a large area of both cognitive science and natural language processing (NLP) research (Graesser et al., 1980; Liversedge et al., 1998; Frank et al., 2013a; Busjahn et al., 2014; Weller and Seppi, 2019, 2020). Online content providers and consumers are also interested in this research; in the increasingly busy world of today, consumers lack the time to read long articles, prompting content creators to aim for specific reading lengths. Many providers[1] have even examined traffic patterns in order to determine the ideal content length, with the general consensus finding 3-7 minutes of content optimal. Thus, having established the optimal content length, article writers now face the next hurdle: when has their post reached the ideal length? A news article about last night's football game may be easier to read than a technical post about NLP. Perhaps the font type or size influences the consumer's comprehension, slowing down the reading process. There are many factors, both textual and stylistic, that quickly come to mind when considering the potential reading time of an article.

Although there has been an extensive body of work on reading time prediction applied to single words (Frank, 2017; Willems et al., 2015; Shain, 2019; van Schijndel and Linzen, 2018), to the best of our knowledge there has been no research into understanding these effects on document sized text. In this paper, we seek to address this area by building models to predict, understand, and interpret factors that could affect an article's reading time. Our contributions to this area include a methodically designed statistical study, consisting of 1130 experimental trials and 32 different articles, experimental results for a broad collection of machine learning algorithms on this novel task, and discussion of potential reasons why more complex models fail. To the best of our knowledge, this is the largest experimental study for reading time research, in terms of participants and breadth of factors. All code and datasets are publicly available.[2]

2 Related Work

Researchers have made significant progress in predicting the reading time of single words, illustrating the effect of different words on the human brain (Frank et al., 2013b; Shain, 2019; Goodkind and Bicknell, 2018) for many different texts (Futrell et al., 2018; Kennedy et al., 2003). Although this

Work done as part of a capstone course with Adobe
[1]Medium's study can be found here.

[2]The code and datasets for our experiments can be found at http://github.com/orionw/DocumentReadingTime

Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 1789–1794

effort is focused more on the cognitive effects of words, these results show that scientists can accurately predict the reading time of individual words in context. With the rise in popularity of machine learning techniques, many scientists have found the most success through these methods, with the most recent research showing significant improvements from combining neural networks as language models with linear mixed models (LMMs) (Goodkind and Bicknell, 2018; de Vries et al., 2018; van Schijndel and Linzen, 2018). However, all previous research has been confined to the effect of a specific word in context, which naturally leads to the question of how this research generalizes.

A separate but similar line of research, readability, measures the reading difficulty of a body of text. This research area has investigated effects of readability in a plethora of areas: online vs paper (Kurniawan and Zaphiris, 2001), color and contrast (Legge et al., 1990), and writing style (Bostian, 1983). The most famous readability metric for English, the Flesch–Kincaid (Kincaid et al., 1975), uses the number of syllables and words to determine readability. Other scientists have attempted to improve upon this simple metric, showing success in reading level classification with unigram language models (Si and Callan, 2001) or SVM models built on top of these basic textual characteristics (Pitler and Nenkova, 2008). As previous metrics seem to be sufficient, recent research has focused on evaluating and comparing the diverse metrics on different domains (Sugawara et al., 2017; Redmiles et al., 2019). We use these readability works to influence our choice of features, as readability seems inherently interwoven with reading time. We employ the *py-readability-metrics* package to include 7 state-of-the-art metrics that we add to our data for the modeling task (Section 4, Appendix B).

3 Experimental Design

We collected our reading time data from a statistical survey performed on Amazon's Mechanical Turk. Since we were not physically present to observe the respondents we took a number of precautions and controls to ensure data quality. We note however, that the inclinations of Mechanical Turk users align with our target audience: we would expect most readers of online content to be of a younger demographic, tech-savy, and prone to read as fast as possible. In this section we will discuss our survey design, validation, and results.

3.1 Survey Design

In order to gather the maximum amount of information from a survey design, we implemented our survey following Fractional Factorial Design (FFD) (Box et al., 2005). This method of survey collection allows us to exploit the sparsity-of-effects principle, gleaning the most information while only using a fraction of the effort of a full factorial design, in terms of experimental runs and resources. This method works by defining two levels for each factor: for example, our factor *font size* had the levels 12 point and 16 point. We extracted 8 factors with 2 levels, consisting of 2^8 unique surveys ($2^{8-3} = 32$ using FFD) to design. When choosing factors and levels, we focused on areas that would provide the most contrast in order to illustrate potential differences in reading time.

Although there are an almost endless number of factors that could potentially influence article reading time, the number of surveys needed to explore those factors increases exponentially; thus, we chose eight crucial factors. Levels of the factor are indicated in parenthesis if applicable: font size (12 vs 16 point), font type (sans vs serif), subject matter (health vs. technology), genre (blog post vs news article), average syllables per word, number of words, average words per sentence, and average unigram frequency. We note that we further collected the original article's text so that additional factors could be easily extracted for future analysis. Again, these factors are not exhaustive but instead were chosen to give a representative sample for a specific area of online articles, while still showing contrast between documents (e.g. news articles vs blog posts or small vs large font).

To define the levels of our numeric features, such as unigram frequency or the average number of syllables, we collected 200 articles for the week of March 4th 2019, aggregating from different news and blog sources, but taking a maximum of three articles from each source (see a more comprehensive list on Github, as there are too many to list). We took these articles, extracted our feature characteristics, and found the median of the distribution. This number was then used as the cutoff between the two levels for that factor. Unigram frequencies were computed using the *wordfreq* library, aggregating frequencies from numerous sources.[3]

[3]Details on which text corpora were aggregated can be found at https://github.com/LuminosoInsight/wordfreq/

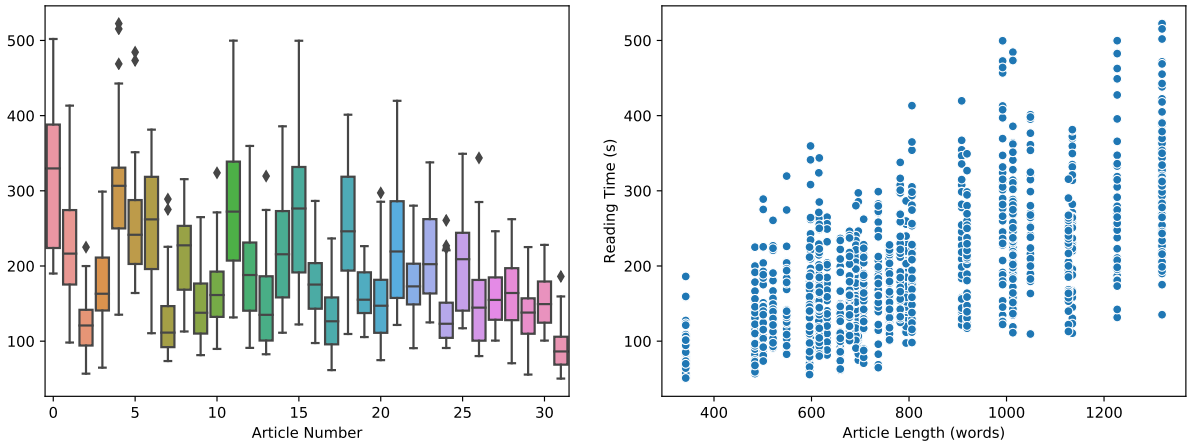

Figure 1: Left: boxplots for the results of each survey, with reading time in seconds. Right: a plot of the number of words vs. reading time. Note that lines in the x-axis are due to each of the 32 surveys having around 40 respondents each, for a total of 1130 respondents.

3.2 Survey Construction

With the requirements for each survey defined by the FFD, we gathered additional articles and parsed their features. We then matched each one of the 32 combinations from the FFD to a unique article that contained those features.

In order to gather a large audience with similar characteristics to online readership, we distributed our survey through Amazon's Mechanical Turk using the Qualtrics platform. Our survey flow consisted of five short demographic questions including age, gender, education level, familiarity with the article subject matter (health or technology) and their perception of their reading speed on a five point Likert scale (slow to fast). They were then instructed to read the next page of the survey uninterrupted at their normal reading pace, after which they would be asked several basic comprehension questions for validation. Each comprehension question was created to be easily answered if the user had read the article but non-trivial for those that had not. See Appendix A for examples of comprehension questions. If the user failed to answer any of the control questions correctly, the survey was terminated and the data was not used.

3.3 Survey Validation and Controls

Due to the nature of Mechanical Turk, we employed various controls to ensure the quality of our data. Many Mechanical Turk workers are prone to take multiple surveys concurrently, leave the page of the survey open for long periods of time, or rush through surveys in order to maximize their earnings. However, the inclination to read through an article quickly is similar to that of online readers, thus, a crowdsourcer's work is acceptable as long as they pass our validation.

In order to control for these tendencies, we included many checks throughout each stage of the survey. If the answers to the demographic questions were unrealistic (such as age greater than 90 or less than 18), we rejected the survey. If the user failed to answer a validation question, such as asking the user to select a certain box before proceeding to the next page, they were disqualified. If the user spent an unrealistic amount of time on the reading page due to any reason (less than two minutes or greater than ten minutes[4] for a long article, as an example) or failed to answer any of the comprehension questions, their data was not used.

3.4 Experimental Results

The results from our surveys are plotted in Figure 1, consisting of 1130 respondents. Note that the results have significant variance, especially as the length of the article increases. More plots of the data can be found in our Github repository.

4 Modeling

With the data gathered and readability metrics calculated (see Section 2), we explore the results from a variety of different models. We employ three categories of models: models that only use

[4]These times were found by initially performing this survey on a limited number of respondents with no limits and then extending the min/max by an additional two minutes.

extracted features, models that only use the text, and models that stack textual-only models with model features. Basic extracted feature models include a vanilla Linear Regression (LR) with only the *number of words* variable ("word"), a Linear Regression model with all variables ("all"), Random Forests, K-Nearest Neighbors (KNN), and a Multi-Layered Perceptron (MLP). As using the entire article as input for the text only models is not computationally feasible, we use modern neural networks to embed the text as a document embedding, using a linear output layer for regression. We tried various state-of-the-art embedding models including roBERTa (Liu et al., 2019; Devlin et al., 2018), XLNet (Yang et al., 2019), and ELMo (Peters et al., 2018). The stacked models combine the document embedding with the extracted features, feeding them both into an MLP. Embeddings use the Flair (Akbik et al., 2018) and HuggingFace (Wolf et al., 2019) libraries.

We use two baselines: a commonly used rule-of-thumb for online reading estimates, 240 words per minute (WPM), and the sum of the word-level predictions (Surprisal-Sum) from a surprisal model in order to compare with recent works (van Schijndel and Linzen, 2018; Shain, 2019). For the Surprsial-Sum baseline predictions, we employ the model used in (van Schijndel and Linzen, 2018), where predictions are made by training a Linear Mixed Model over surprisal data.

5 Results

The results from our experiments are found in Table 1. We see that the most effective models were the simplest: the 240 WPM baseline, linear regression, k-nearest neighbors, and random forests. Using the word count only linear model, because of its easy interpretability, shows us an R^2 value of 0.40, meaning that 40% of the variance of reading time can be explained by the number of words in the article. We also see that scaling a regression model to include demographic and textual information (the "all" linear regression model) does not seem to provide significant improvements in prediction.

Given the amount of empirical evidence from word level reading time prediction, we were surprised to see a dearth of similar results for document level prediction. Models that provide strong results in word level prediction, such as varieties of neural networks, fail to be as effective as the simpler models. Perhaps this is due to the length of the document - small changes in word level reading time simply get evened out at the document level (for example, see the Surprisal-Sum model). Alternatively, the level of surprisal in online articles may remain constant with the number of words.

Features Only:	RMSE	(sd)	MAE	(sd)
240 WPM	**66.0**	10.7	**52.1**	8.3
Surprisal-Sum	141.5	42.8	118.4	35.8
MLP	84.8	10.5	67.2	7.0
Random Forest	**64.3**	7.7	**50.2**	5.6
LR (word)	**65.5**	10.7	**51.1**	7.9
LR (all)	**65.7**	9.8	**51.6**	8.0
KNN	**70.1**	9.6	**54.3**	7.1
Text-Only:	**RMSE**	**(sd)**	**MAE**	**(sd)**
XLNet	81.0	8.6	62.8	6.6
ELMo	84.3	13.1	66.7	8.6
roBERTa	83.2	13.9	66.3	9.1
Stacked:	**RMSE**	**(sd)**	**MAE**	**(sd)**
XLNet/MLP	80.3	10.4	62.9	8.0
ELMo/MLP	83.2	13.7	66.4	9.4
roBERTa/MLP	83.5	10.5	66.1	6.9

Table 1: Results on the reading time prediction task. RMSE and MAE are reported in seconds for the mean of a 10-fold cross validation. "sd" indicates one standard deviation for the previous metric. Best results in each column are in bold.

6 Conclusion

Given previous work in single word reading time prediction, we conducted a large novel study to test whether document level reading time could be predicted. We carefully designed an experiment containing a myriad of potential factors to measure reading time, distributed the survey to more than a thousand people, and collected the results into the first dataset of its kind. We then employed machine learning techniques to predict the time to read, finding that simpler models were the most competitive, with the number of words as the sole critical factor in predicting reading time. We hope this resource can benefit future research into developing techniques to model and understand human responses to document sized text.

Acknowledgements

We would like to thank Hayden Harris for his help and advice during the capstone project.

References

Alan Akbik, Duncan Blythe, and Roland Vollgraf. 2018. Contextual string embeddings for sequence labeling. In *COLING 2018, 27th International Conference on Computational Linguistics*, pages 1638–1649.

Lloyd R Bostian. 1983. How active, passive and nominal styles affect readability of science writing. *Journalism quarterly*, 60(4):635–670.

George EP Box, J Stuart Hunter, and William G Hunter. 2005. Statistics for experimenters. In *Wiley Series in Probability and Statistics*. Wiley Hoboken, NJ, USA.

Teresa Busjahn, Roman Bednarik, and Carsten Schulte. 2014. What influences dwell time during source code reading?: analysis of element type and frequency as factors. In *Proceedings of the Symposium on Eye Tracking Research and Applications*, pages 335–338. ACM.

Jeanne Sternlicht Chall and Edgar Dale. 1995. *Readability revisited: The new Dale-Chall readability formula*. Brookline Books.

Meri Coleman and Ta Lin Liau. 1975. A computer readability formula designed for machine scoring. *Journal of Applied Psychology*, 60(2):283.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *North American Chapter of the Association for Computational Linguistics*.

Rudolph Flesch. 1948. A new readability yardstick. *Journal of applied psychology*, 32(3):221.

Stefan Frank. 2017. Word embedding distance does not predict word reading time. In *CogSci*.

Stefan L. Frank, Irene Fernandez Monsalve, Robin L. Thompson, and Gabriella Vigliocco. 2013a. Reading time data for evaluating broad-coverage models of english sentence processing. *Behavior Research Methods*, 45(4):1182–1190.

Stefan L. Frank, Leun J. Otten, Giulia Galli, and Gabriella Vigliocco. 2013b. Word surprisal predicts n400 amplitude during reading. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 878–883, Sofia, Bulgaria. Association for Computational Linguistics.

Richard Futrell, Edward Gibson, Harry J Tily, Idan Blank, Anastasia Vishnevetsky, Steven Piantadosi, and Evelina Fedorenko. 2018. The natural stories corpus. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Adam Goodkind and Klinton Bicknell. 2018. Predictive power of word surprisal for reading times is a linear function of language model quality. In *Proceedings of the 8th Workshop on Cognitive Modeling and Computational Linguistics (CMCL 2018)*, pages 10–18, Salt Lake City, Utah. Association for Computational Linguistics.

Arthur C Graesser, Nicholas L Hoffman, and Leslie F Clark. 1980. Structural components of reading time. *Journal of Verbal Learning and Verbal Behavior*, 19(2):135–151.

Robert Gunning et al. 1952. Technique of clear writing.

Alan Kennedy, Robin Hill, and Joël Pynte. 2003. The dundee corpus. In *Proceedings of the 12th European conference on eye movement*.

J Peter Kincaid, Robert P Fishburne Jr, Richard L Rogers, and Brad S Chissom. 1975. Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel.

George R Klare. 1974. Assessing readability. *Reading research quarterly*, pages 62–102.

Sri Hastuti Kurniawan and Panayiotis Zaphiris. 2001. Reading online or on paper: Which is faster?

Gordon E Legge, David H Parish, Andrew Luebker, and Lee H Wurm. 1990. Psychophysics of reading. xi. comparing color contrast and luminance contrast. *JOSA A*, 7(10):2002–2010.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv e-prints*, page arXiv:1907.11692.

Simon P Liversedge, Kevin B Paterson, and Martin J Pickering. 1998. Eye movements and measures of reading time. In *Eye guidance in reading and scene perception*, pages 55–75. Elsevier.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of NAACL-HLT*, pages 2227–2237.

Emily Pitler and Ani Nenkova. 2008. Revisiting readability: A unified framework for predicting text quality. In *Proceedings of the conference on empirical methods in natural language processing*, pages 186–195. Association for Computational Linguistics.

Elissa Redmiles, Lisa Maszkiewicz, Emily Hwang, Dhruv Kuchhal, Everest Liu, Miraida Morales, Denis Peskov, Sudha Rao, Rock Stevens, Kristina Gligorić, et al. 2019. Comparing and developing tools to measure the readability of domain-specific texts. In

Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 4833–4844.

Marten van Schijndel and Tal Linzen. 2018. A neural model of adaptation in reading. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4704–4710.

Cory Shain. 2019. A large-scale study of the effects of word frequency and predictability in naturalistic reading. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4086–4094.

Luo Si and Jamie Callan. 2001. A statistical model for scientific readability. In *CIKM*, volume 1, pages 574–576.

Edgar A Smith and RJ Senter. 1967. Automated readability index. *AMRL-TR. Aerospace Medical Research Laboratories (US)*, pages 1–14.

Saku Sugawara, Yusuke Kido, Hikaru Yokono, and Akiko Aizawa. 2017. Evaluation metrics for machine reading comprehension: Prerequisite skills and readability. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 806–817.

Clarissa de Vries, W Gudrun Reijnierse, and Roel M Willems. 2018. Eye movements reveal readers' sensitivity to deliberate metaphors during narrative reading. *Scientific Study of Literature*, 8(1):135–164.

Orion Weller and Kevin Seppi. 2019. Humor detection: A transformer gets the last laugh. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3612–3616.

Orion Weller and Kevin Seppi. 2020. The rjokes dataset: a large scale humor collection. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 6136–6141, Marseille, France. European Language Resources Association.

Roel M. Willems, Stefan L. Frank, Annabel D. Nijhof, Peter Hagoort, and Antal van den Bosch. 2015. Prediction During Natural Language Comprehension. *Cerebral Cortex*, 26(6):2506–2516.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Transformers: State-of-the-art natural language processing.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, pages 5754–5764.

A Comprehension Questions

We designed our comprehension questions such that the answer would not be trivially obvious to those who did not read the article. In this example, an article about *Minecraft Mods*, we ask two questions that would even require someone familiar with Minecraft to read the article: asking them what the author's opinion was and what the term *mod* stood for in this specific context. We further put these questions on the page after the reading section of the survey and did not allow respondents to go back to re-read the text.

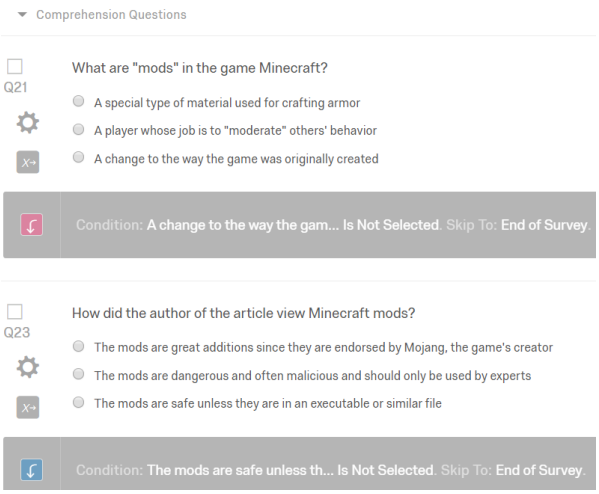

Figure 2: Example comprehension questions for an article about Minecraft

B Readability Metrics

We use the following metrics calculated from the py-readability-metrics package:

- Flesch-Kincaid (Kincaid et al., 1975)
- Flesch (Flesch, 1948)
- Gunning-Fog (Gunning et al., 1952)
- Coleman-Liau (Coleman and Liau, 1975)
- Dale-Chall (Chall and Dale, 1995)
- Ari (Smith and Senter, 1967)
- Linsear Write (Klare, 1974)