


Article

Quickly Finding the Semantically Optimal Presentation Order for a Set of Text Artifacts

Daniel S. Soper 

Department of Information Systems & Decision Sciences, California State University, Fullerton, CA 92831, USA; dsoper@fullerton.edu

Abstract: This study considers how to quickly find the order in which to present a set of text artifacts on mobile apps or websites such that those artifacts are maximally semantically separated. Semantic separation is desirable because it ensures that users experience as much novelty as possible from one item to the next, thereby improving user attention and engagement. Since an exhaustive search of all possible sequences of text items becomes increasingly infeasible as the length of the sequence grows, a new algorithm is proposed to quickly find the semantically optimal presentation order for a set of text artifacts. The performance of the proposed algorithm is evaluated using an extensive set of experiments involving three different types of text artifacts, seven different sequence lengths, and more than 600 experimental trials. The results demonstrate that the proposed algorithm can select statistically optimal sequences of text artifacts extremely quickly, regardless of the type of text artifacts being used as input or the length of the sequence. App and website developers who are seeking to hold users' attention and improve user engagement may therefore find the proposed algorithm very attractive in comparison to an exhaustive search.

Keywords: text mining; semantic separation; content presentation order; web design; app design; pagination; infinite scroll; user engagement

1. Introduction

It is very common for modern apps and websites to present content in the form of a sequence of items through which users can browse. Video-sharing apps, for example, present users with a sequence of video previews, while news websites present users with a sequence of news article headlines. Streaming television and music apps present users with sequences of available shows and recordings, while social media platforms present users with sequences of photos, messages, and other posts. Similarly, search engines present information to users in the form of a sequence of items, as do online retailers. These sequences of content items are typically displayed to users either through the interface design pattern known as "pagination" or through the pattern known as "infinite scroll" [1,2]. Both of these interface design patterns allow users to browse through a subset of available content items, and if none of the currently displayed items is of interest, to fetch another sequence of content items, either by loading the next page of results in the case of pagination [3], or by scrolling downward in the case of infinite scroll [4].

Regardless of whether an app or website implements pagination or infinite scroll, each sequence of content items is finite in length. No universal agreement exists regarding the optimal number of items that should be included in each sequence, but sequences of length 10 have been observed to be common for search results [5]. For any set of content items of length $k > 1$, app and website developers must determine the order in which the items will



Academic Editor: Fei Liu

Received: 5 November 2024

Revised: 7 January 2025

Accepted: 14 January 2025

Published: 16 January 2025

Citation: Soper, D.S. Quickly Finding the Semantically Optimal Presentation Order for a Set of Text Artifacts.

Information **2025**, *16*, 59. <https://doi.org/10.3390/info16010059>

Copyright: © 2025 by the author. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

be presented to users, thus transforming the unordered set into a sequence. Search engines, for example, commonly order content items according to their relevance to the user’s query, while news articles are typically presented in reverse chronological order. Alphabetical order, numerical order, ordering by popularity, and ordering by user ratings are just a few of the many other ways in which the presentation order for content items is often determined.

For apps and websites that rely on an advertising-based revenue model, a common design goal is to keep users engaged on the app or website as long as possible, thus allowing more advertisements to be displayed and more revenue to be generated [6]. It follows, of course, that in order to keep a user engaged, the app or website must hold the user’s attention. Past research has demonstrated that novelty plays a significant role in human attention processes, with novel stimuli leading to increased attention [7,8]. Physiologically, novel stimuli trigger the release of the neurotransmitter dopamine, which heightens attention through cognitive pleasure and reward mechanisms that make the novel stimuli more enticing [9,10]. Ensuring that users experience novelty while browsing from one content item to the next can therefore be expected to heighten their attention and bolster their engagement with an app or website. This theoretical connection between novelty and user engagement is supported by a variety of empirical studies, including research that found positive impacts of novelty on user engagement on streaming media platforms [11], research connecting novelty to user engagement with online news articles [12], and research showing how novelty improves user engagement with blog posts [13]. By contrast, a lack of novelty, as manifested through excessive redundancy or similarity from one content item to the next, can be expected to lead to waning attention and lower user engagement.

Many of the content items that appear on apps and websites take the form of text artifacts, including, for example, search results, product descriptions, reviews, news headlines, social media messages, blog posts, frequently asked questions (FAQs), and tutorials and how-to articles, among many others. In light of the discussion above, developers of apps and websites on which text artifacts appear as sequences of content items should consider the extent to which those items exhibit novelty from one item to the next. Given a set of text artifacts, maximal novelty can be achieved by ensuring that the semantic distance between the artifacts in the sequence is as large as possible, with semantic distance referring to the extent to which the meaning of two text artifacts differs [14]. For a finite set of text artifacts, however, many different sequences are possible. To maximize the novelty experienced by the user, the app or website must therefore find the specific sequence of those artifacts that exhibits the largest total semantic distance, as determined by the sum of the semantic distances between each adjacent pair of text artifacts. To date, however, no systematic study has been undertaken that considers how to identify the semantically optimal presentation order for a set of text artifacts in an efficient manner. The current paper seeks to fill this gap in the extant literature.

To better clarify the nature of this semantic similarity problem, consider a set of three text artifacts $\{A, B, C\}$, the possible sequences of which are shown in Table 1.

Table 1. Possible sequences for a set of three text artifacts $\{A, B, C\}$.

Sequence	
$A \rightarrow B \rightarrow C$	$C \rightarrow B \rightarrow A$
$A \rightarrow C \rightarrow B$	$B \rightarrow C \rightarrow A$
$B \rightarrow A \rightarrow C$	$C \rightarrow A \rightarrow B$

As a general rule, there are $k!$ possible sequences (i.e., permutations) of length k for a set containing k elements. In the context of semantic similarity, however, it is important to recognize that the semantic distance between any two text artifacts is symmetric, such that

the semantic distance between artifact A and artifact B is identical to the semantic distance between artifact B and artifact A . By extension, a sequence of text artifacts is bidirectional insofar as the sum of the semantic distances for a sequence $A \rightarrow B \rightarrow C$ is identical to the sum of the semantic distances for its symmetric counterpart $C \rightarrow B \rightarrow A$. Since sequences of text artifacts are semantically bidirectional, the symmetric version of each sequence can be excluded from consideration when using the sequence as a whole as the unit of analysis. The three sequences in the rightmost column of Table 1, for example, can be ignored because they are all symmetric versions of the three sequences that appear in the leftmost column of the table. Excluding symmetric versions, the number of possible sequences of length k for a set of k text artifacts is thus equal to $k!/2$.

Although a sequence in which all pairs of adjacent text artifacts are equidistantly spaced may be theoretically appealing, in the absence of any duplicate artifacts, such an outcome would be highly improbable in the real world because the semantic content of each artifact in the sequence would differ in some way from the semantic content of every other artifact in the sequence. Given a finite set of distinct text artifacts as input and the objective of maximizing the semantic novelty from one artifact to the next, the best we can do is therefore to identify the sequence for which the sum of semantic distances between each pair of adjacent artifacts is as large as possible. This notion is illustrated for two different sets of text artifacts in Figure 1 below, with the optimal sequences highlighted in green.

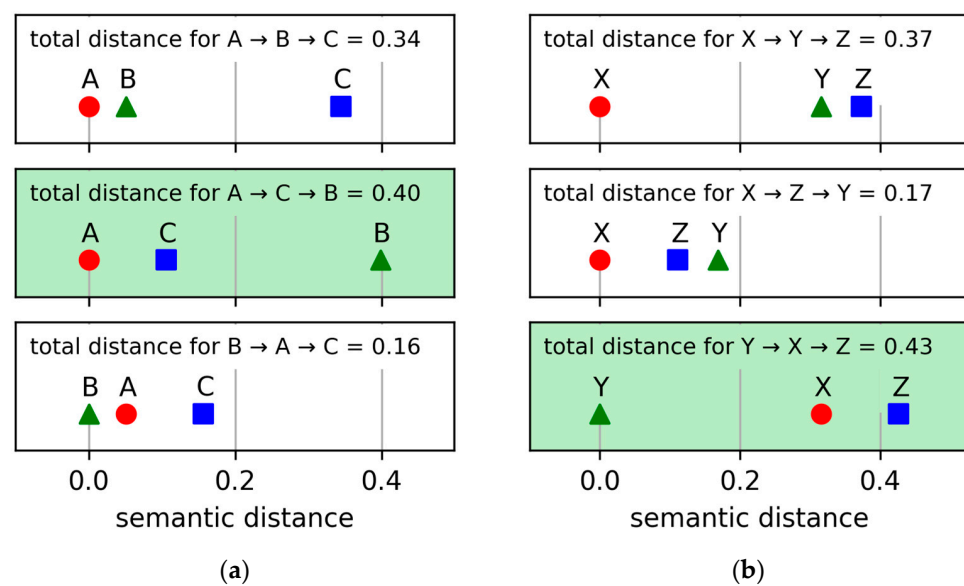


Figure 1. Semantic distances for all possible bidirectional sequences of two sets of text artifacts. The optimal presentation order is the sequence for which the sum of the semantic distances between each adjacent pair of items in the set is as large as possible. (a) The optimal sequence for the set of text artifacts $\{A, B, C\}$ is $A \rightarrow C \rightarrow B$ (or its symmetric equivalent, $B \rightarrow C \rightarrow A$). (b) The optimal sequence for the set of text artifacts $\{X, Y, Z\}$ is $Y \rightarrow X \rightarrow Z$ (or its symmetric equivalent, $Z \rightarrow X \rightarrow Y$).

As shown in Figure 1, the sum of semantic distances for a sequence of text artifacts depends on the order in which those artifacts appear in the sequence. In panel (a) of Figure 1, for example, the total semantic distance for the sequence $B \rightarrow A \rightarrow C$ is 0.16, while the total semantic distance for the same text artifacts when arranged into the sequence $A \rightarrow C \rightarrow B$ is 0.40. Since the semantic distance between each pair of adjacent artifacts reflects the degree to which their topical content differs, the transition from artifact C to artifact B would be experienced by a user as a more substantial or novel shift in content than the transition from, say, artifact B to artifact A , whose small semantic distance suggests a much greater degree of topical redundancy or overlap. For a finite set of text artifacts, the greatest

possible amount of novelty that the user can experience is obtained from the sequence with the largest total semantic distance, and for that purpose, the sequence with the greatest length can be considered optimal.

While finding the best possible sequence should be of great interest to developers of apps and websites, it is also critical to be able to find the best sequence quickly in order to ensure a high-quality user experience. As noted previously, the total number of possible bidirectional sequences of length k for a finite set of text artifacts containing k elements is equal to $k!/2$. Finding the semantically optimal presentation order for a set of text artifacts can therefore be easily and quickly accomplished using an exhaustive search when the number of artifacts in the set is small. For a larger set of text artifacts, however, providing a high-quality user experience would be infeasible using an exhaustive search due to exponential growth in both the number of possible sequences and the wall-clock time that would be required to examine those sequences. A set of five text artifacts, for example, could only be arranged into $5!/2 = 60$ different bidirectional sequences, while a total of $10!/2 = 1,814,400$ bidirectional sequences would be possible for a set of 10 artifacts. For a set of 15 text artifacts, however, a total of $15!/2 = 653,837,184,000$ bidirectional sequences would be possible, making an exhaustive search computationally infeasible in any reasonable amount of wall-clock time.

In light of the discussion immediately above, the goal of the current paper is to develop and evaluate the performance of an algorithm that can quickly find the semantically optimal presentation order for a set of text artifacts, even when the number of artifacts in the set is comparatively large. If proven tenable, the resulting algorithm could be used by developers of a wide variety of apps and websites to present content items to users in a sequence that is intentionally designed to hold the users' attention, thus increasing both user engagement and its attendant revenues.

The balance of this paper is organized as follows: the next section presents the proposed algorithm for quickly finding the semantically optimal presentation order for a set of text artifacts, as well as descriptions of the data, experiments, and methods of evaluation that were used to assess the algorithm's performance. Section 3 presents the results obtained from the experiments, and those results are subsequently discussed in Section 4. This paper concludes in Section 5 with a summary, a discussion of this study's limitations, and opportunities for future research.

2. Materials and Methods

2.1. Encoding and Measuring the Semantic Distance Between Text Artifacts

While there are many methods of capturing the semantic content of text artifacts, the current study adopted the common and well-established approach in which the semantic content of each artifact is encoded into a vector in a shared n -dimensional space [15]. A simple two-dimensional example of this vector-space model is illustrated in Figure 2 below.

Representing text artifacts as vectors of length n in a shared n -dimensional space is convenient because it allows the degree of semantic similarity (or dissimilarity) among the text artifacts to be measured as a function of the distance between their corresponding vectors. For the current study, the cosine distance metric was used to quantify the semantic distance between text artifacts because its values are naturally constrained to the closed interval $[0.0, 1.0]$, with a value of 0.0 indicating that two text artifacts are semantically identical and a value of 1.0 indicating that two artifacts are semantically orthogonal [16]. All text artifacts used in the current study were encoded into vectors using the pretrained "all-MiniLM-L6-v2" model, which is freely available as part of the Sentence Transformers library in Python [17]. This particular model maps each text artifact to a vector space consisting of 384 dimensions.

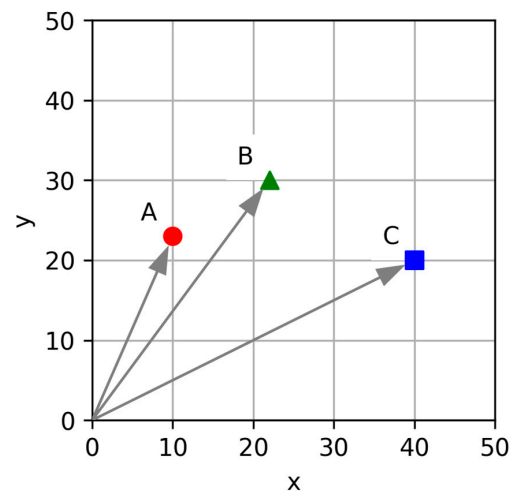


Figure 2. A set of three text artifacts {A, B, C} represented as vectors in a two-dimensional space.

2.2. The Proposed Algorithm

Having described the methods that were used in the current study to encode text artifacts into vectors and measure the semantic distance between them, we are now equipped to proceed with a complete description of the proposed algorithm for quickly finding the optimal presentation order for a set of text artifacts. As noted previously and as illustrated in Figure 1, the optimal presentation order is defined as the bidirectional sequence for which the sum of the semantic distances between all adjacent pairs of text artifacts in the sequence is as large as possible. The steps in the algorithm (Algorithm 1) below are described textually rather than symbolically in order to ensure clarity.

Algorithm 1: Quickly finding the semantically optimal presentation order for a set of text artifacts

1. Encode all text artifacts that will appear in the sequence into vectors in a shared high-dimensional space.
 2. Compute the cosine distances between all possible pairs of text artifact vectors.
 3. Initialize storage locations to hold the best observed sequence of text artifacts and the total semantic distance of that sequence.
 4. For each text artifact in the set, complete the following:
 - a. Initialize storage locations to hold a new candidate sequence and the total semantic distance of the candidate sequence.
 - b. Use the current text artifact as the first item in the new sequence.
 - c. While text artifacts remain to be added to this candidate sequence, complete the following:
 - i. Among the remaining (unused) text artifacts, find the artifact whose cosine distance from the most recent item in the sequence is largest, and add that artifact to the sequence.
 - ii. Update the total semantic distance of the candidate sequence to include the cosine distance between the two most recent items.
 - d. If the current candidate sequence has the largest total semantic distance thus far observed, complete the following:
 - i. Update the value of the largest total semantic distance thus far observed.
 - ii. Store the current candidate sequence as the best sequence thus far observed.
 5. Return the best sequence (i.e., the sequence with the largest total semantic distance that was observed during the search).
-

A simplified graphical representation of the proposed algorithm is provided in Figure 3 below.

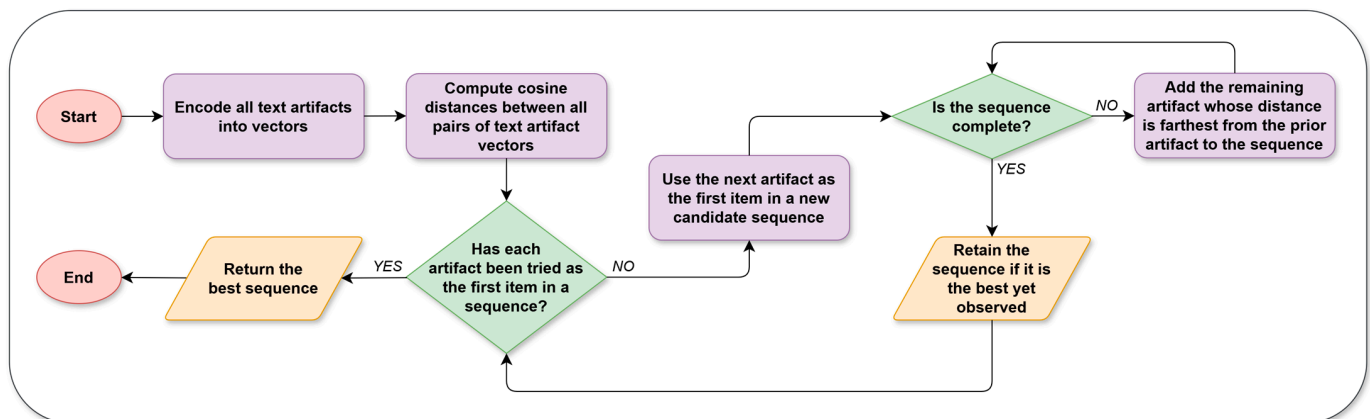


Figure 3. A simplified graphical representation of the proposed algorithm.

Notably, for a set of k text artifacts, the proposed algorithm constructs and considers just k candidate sequences, with each text artifact being used exactly once as the first item in a sequence. After placing the first text artifact in a candidate sequence, the proposed algorithm greedily uses whichever of the remaining artifacts has the largest cosine distance from the most recent item as the next item in the sequence. This process continues until the candidate sequence is complete. The newly completed candidate sequence is then retained if it has the largest total semantic distance thus far observed or is discarded otherwise. After constructing and considering each of the k candidate sequences, the proposed algorithm returns the best-observed sequence and then terminates. Since the number of candidate sequences considered by the proposed algorithm corresponds directly to the number of text artifacts in the input set, it follows that the wall-clock time required by the algorithm scales linearly with the size of the set (or equivalently, with the length of the sequence). Given that the proposed algorithm has an $O(n)$ complexity, while an exhaustive search has an $O(2^n)$ complexity, it is reasonable to expect that the total execution time of the proposed algorithm will be very fast in comparison to an exhaustive search.

2.3. Data

If found to be viable, the proposed algorithm could be usefully applied to determine the semantically optimal sequence of a set of text artifacts in a very wide variety of scenarios. As a means of evaluating and demonstrating this broad applicability, the data set that was used to evaluate the performance of the algorithm was intentionally designed to contain text artifacts from three distinct domains, including (1) synopses of movie plots, (2) news articles about many different topics, and (3) lyrics for popular songs. In total, 30 of each of these different types of text artifacts were included in the data set, yielding a total of 90 unique text artifacts for use in the experiments.

The movie synopses were obtained from the Internet Movie Database (IMDb), which describes itself as “the world’s most popular and authoritative source for information on movies, TV shows, and celebrities” [18]. Specifically, the IMDb synopses for the 30 highest-grossing movies of all time were used in the experiments, according to the worldwide gross box office revenue figures published by Box Office Mojo as of October 2024 [19]. The 30 news articles used in the experiments were derived from the freely available Reuters Corpus that is included as part of the Natural Language Toolkit (NLTK) suite of Python libraries [20]. The 30 specific news articles that were included in the data set were chosen at random from among the 10,788 articles contained in the NLTK Reuters Corpus. Finally, the

song lyrics included in the data set were for the 30 greatest songs of all time as of October 2024, according to the songs' performances in the United States on the Billboard Hot 100 Singles Chart [21], with the song lyrics themselves being obtained from LyricFind's Lyrics Database [22]. Interested readers may download all of the raw data used in the experiments by following the instructions in the Data Availability Statement that appears near the end of this paper.

2.4. Experiments

Three sets of experiments were conducted in order to evaluate the performance of the proposed algorithm, with each set of experiments using one of the text sources described immediately above (i.e., movie synopses, news articles, or song lyrics). Each set of experiments examined sequences of text artifacts of length $k = \{5, 6, \dots, 11\}$. A total of 30 trials were carried out for each combination of text source and sequence length (i.e., for each experimental condition) in order to ensure that the distributions of the resulting performance metrics would be approximately Gaussian, per the central limit theorem [23]. The k specific text artifacts used in each trial were chosen at random from among the set of 30 artifacts for the trial's corresponding source of text. The experiment design thus involved three sources of text and seven different sequence lengths, yielding 21 unique experimental conditions. Since each experimental condition was tested 30 times, a total of $21 \times 30 = 630$ experimental trials were conducted in the current study. All of the experimental trials were carried out sequentially using identical hardware on the Google Colaboratory platform [24].

The proposed algorithm itself was assessed in two different ways, both of which involved comparing the algorithm's performance in each experimental trial against that of a corresponding exhaustive search that used the same text artifacts and sequence length. First, the wall-clock time required by the algorithm was compared against that of an exhaustive search, and second, the quality of the sequence chosen by the proposed algorithm in each trial was compared against the globally optimal sequence for that trial, as determined by the exhaustive search. This involved both a comparison of the total semantic distance of the sequence chosen by the proposed algorithm against the total semantic distance of the globally optimal sequence, as well as calculating the percentile of the algorithm's chosen sequence in comparison to all possible sequences for the current experimental trial. Specific details about the methods that were used for evaluating the proposed algorithm are provided in the following subsection. Interested readers may also download all of the Python source code that was used to conduct the experiments by following the instructions in the Data Availability Statement that appears near the end of this paper.

2.5. Methods for Evaluating the Proposed Algorithm

As noted above, the performance of the proposed algorithm was evaluated both in terms of its wall-clock execution time relative to an exhaustive search and in terms of the quality of the algorithm's chosen sequence of text artifacts. With respect to wall-clock time, the total execution time of the proposed algorithm and its corresponding exhaustive search were recorded for each of the 30 trials that were carried out in each of the 21 different experimental conditions. Welch's t -tests were then used to compare the wall-clock times from the proposed algorithm and its matching exhaustive searches for each experimental condition. Welch's t -tests were used for this purpose because they allow for unequal variances, and there was no reason to expect that the variances of the distributions of the wall-clock times for the proposed algorithm and the exhaustive searches would be equal [25].

Comparing the wall-clock times of the proposed algorithm and an exhaustive search was necessary but insufficient for fully assessing the algorithm’s viability since being faster than an exhaustive search would be of little value if the quality of the sequences chosen by the proposed algorithm were poor. For this reason, the total semantic distances of the sequences of text artifacts chosen by the proposed algorithm were statistically compared against those of the globally optimal sequences for each experimental condition, as determined by a matching set of exhaustive searches. Again, Welch’s *t*-tests were used for this purpose because there was no reason to expect that the variances in the distributions of semantic distances for the sequences chosen by the algorithm and those identified as optimal via an exhaustive search would be equal [25]. For each experimental condition, if the proposed algorithm was found to require significantly less wall-clock time than an exhaustive search while simultaneously choosing sequences of text artifacts that were statistically indistinguishable from their corresponding optimal sequences, then it could be reasonably concluded that the proposed algorithm is superior to an exhaustive search.

Finally, the quality of the sequence of text artifacts chosen by the proposed algorithm in each experimental trial was also quantified by computing the percentile of the chosen sequence relative to all possible sequences for that trial. Specifically, the percentile of a chosen sequence was calculated as $1 - (\text{number of superior sequences} / \text{total sequences})$, with superior sequences being defined as those whose total semantic distance was greater than that of the sequence chosen by the proposed algorithm. For example, a set of six text artifacts would have a total of $6!/2 = 360$ possible bidirectional sequences. If two of those sequences had a larger total semantic distance than the sequence chosen by the algorithm, then the quality of the sequence chosen by the algorithm would be quantified as $1 - (2/360) \approx 0.994$, yielding a percentile for the algorithm’s chosen sequence of 99.4%. The wall-clock times, total semantic distances, and percentiles of the proposed algorithm’s chosen sequences are presented in the following section.

3. Results

The results of the experiments that were described in the previous section are presented in Tables 2–4 below, with the tables respectively showing the comparative performance of the proposed algorithm relative to an exhaustive search for the movie synopses, news articles, and song lyrics text artifacts.

Table 2. Proposed algorithm performance on movie synopses relative to an exhaustive search.

Sequence Length	Mean Wall-Clock Time (Seconds)		Mean Cosine Distance of Best Sequence		Average Quality of Algorithmic Sequence
	Proposed Algorithm	Exhaustive Search	Proposed Algorithm	Exhaustive Search	
5	0.00028 ***	0.00316	3.52583	3.53478 ^{ns}	0.99278
6	0.00031 ***	0.00797	4.44218	4.45140 ^{ns}	0.99722
7	0.00029 ***	0.02623	5.29985	5.32438 ^{ns}	0.99827
8	0.00041 ***	0.23103	6.27516	6.31719 ^{ns}	0.99788
9	0.00050 ***	2.16796	7.14110	7.19904 ^{ns}	0.99986
10	0.00057 ***	22.07154	8.12200	8.16449 ^{ns}	0.99998
11	0.00072 ***	256.39432	9.08820	9.19619 ^{ns}	0.99996

*** $p < 0.001$, ^{ns} $p =$ not significant.

Beginning with wall-clock time, it is immediately evident in Tables 2–4 that the average time required by the proposed algorithm to choose a sequence of text artifacts is much less than the average wall-clock time required by an exhaustive search. Moreover, this difference was highly significant at $p < 0.001$ for all 21 combinations of text sources and sequence lengths considered in the experiments. As expected, the average wall-clock time required by the proposed algorithm was observed to grow linearly with the length of the sequence

(or, equivalently, with the number of text artifacts in the input set), while the average wall-clock time required by an exhaustive search was observed to grow exponentially. These characteristics are illustrated in Figure 4 below.

Table 3. Proposed algorithm performance on news articles relative to an exhaustive search.

Sequence Length	Mean Wall-Clock Time (Seconds)		Mean Cosine Distance of Best Sequence		Average Quality of Algorithmic Sequence
	Proposed Algorithm	Exhaustive Search	Proposed Algorithm	Exhaustive Search	
5	0.00036 ***	0.00074	3.59040	3.59219 ^{ns}	0.99500
6	0.00026 ***	0.00363	4.50808	4.51811 ^{ns}	0.99833
7	0.00034 ***	0.02710	5.39192	5.41109 ^{ns}	0.99923
8	0.00044 ***	0.23080	6.37339	6.44631 ^{ns}	0.99881
9	0.00053 ***	2.05910	7.27462	7.33826 ^{ns}	0.99959
10	0.00063 ***	22.24218	8.30702	8.39426 ^{ns}	0.99992
11	0.00070 ***	256.20882	9.17250	9.29327 ^{ns}	0.99997

*** $p < 0.001$, ^{ns} $p =$ not significant.

Table 4. Proposed algorithm performance on song lyrics relative to an exhaustive search.

Sequence Length	Mean Wall-Clock Time (Seconds)		Mean Cosine Distance of Best Sequence		Average Quality of Algorithmic Sequence
	Proposed Algorithm	Exhaustive Search	Proposed Algorithm	Exhaustive Search	
5	0.00021 ***	0.00061	2.48525	2.49249 ^{ns}	0.99278
6	0.00045 ***	0.00373	3.06495	3.09022 ^{ns}	0.99324
7	0.00037 ***	0.02884	3.70566	3.74845 ^{ns}	0.99595
8	0.00041 ***	0.23391	4.39472	4.46422 ^{ns}	0.99781
9	0.00051 ***	2.12178	4.98578	5.08293 ^{ns}	0.99752
10	0.00064 ***	22.08317	5.77806	5.92306 ^{ns}	0.99865
11	0.00068 ***	255.74082	6.30305	6.49257 ^{ns}	0.99877

*** $p < 0.001$, ^{ns} $p =$ not significant.

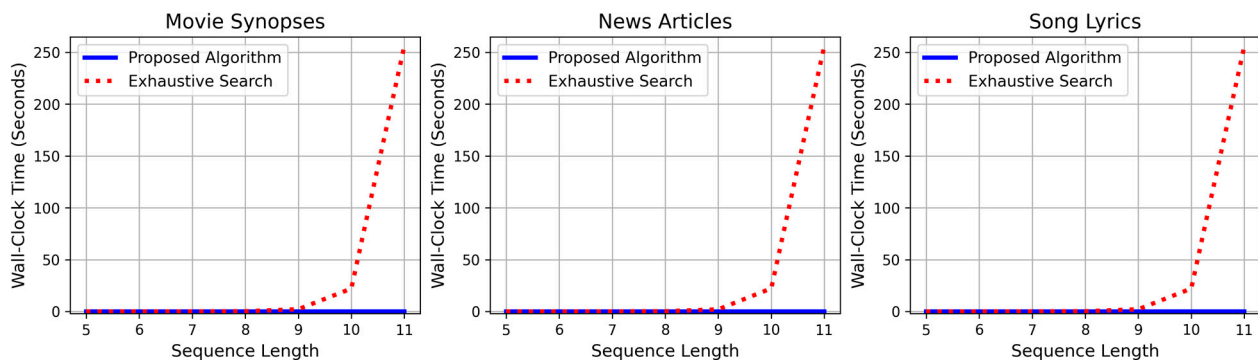


Figure 4. Average wall-clock time required by the proposed algorithm in comparison to an exhaustive search.

Moving next to a consideration of sequence quality, it is evident in Tables 2–4 that, on average, the total semantic distance of the sequence of text artifacts chosen by the proposed algorithm is slightly less than the total semantic distance of the optimal sequence identified via an exhaustive search. Importantly, however, this difference was not observed to be statistically significant for any of the 21 different combinations of text sources and sequence lengths evaluated in the experiments. Thus, although the total semantic distances of the sequences of text artifacts identified by the proposed algorithm are nominally smaller than those identified via an exhaustive search, they are, in fact, statistically identical in all cases to the optimal sequences. It can therefore be concluded that, on average, the proposed algorithm selects sequences of text artifacts that are statistically optimal.

Finally, Tables 2–4 reveal that the average quality of the sequences chosen by the proposed algorithm—as measured by their percentiles relative to all possible sequences—not only consistently exceeds 99% but also tends to approach 100% as the length of the sequence grows. Put differently, the average performance of the proposed algorithm in terms of the quality of the sequences of text artifacts that it selects tends to improve as the sequence grows longer. The observed improvement in the relative quality of the sequences selected by the algorithm is attributable to the exponential growth in the total number of sequences that are possible as the length of the sequence grows. If, for example, the algorithm chose the second-best option out of 100 possible sequences, then the quality of the chosen sequence would be 99.0%. If the algorithm chose the second-best option out of 1000 possible sequences, however, the quality of the chosen sequence would be 99.9%. This behavior is illustrated in Figure 5 below, which shows how the average quality of the proposed algorithm’s chosen sequences approaches 100% according to a logarithmic function.

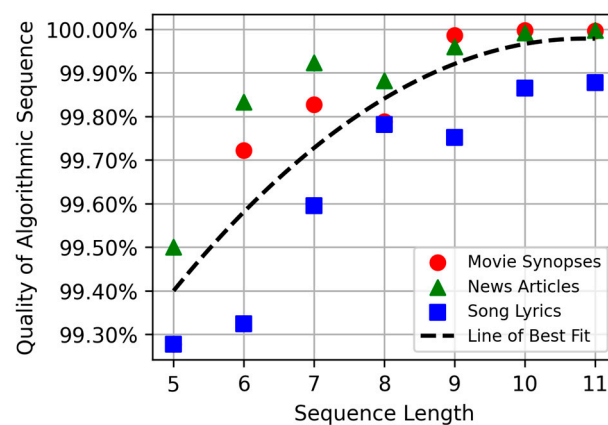


Figure 5. Average quality of sequences identified by the proposed algorithm relative to all possible sequences of the same length.

The results obtained from the experiments and their implications for practitioners are discussed in greater detail in the following section.

4. Discussion

Collectively, the results of the experiments reported in the previous section suggest that the proposed algorithm runs very quickly and yields sequences of text artifacts whose total semantic distances are statistically optimal. The fact that the proposed algorithm is consistently able to deliver statistically optimal sequences of text artifacts in a very short amount of time has important implications for developers of apps and websites that present sequences of textual content to users. Namely, unless the set of input text artifacts is very small, finding the semantically optimal sequence of artifacts using an exhaustive search simply requires too much wall-clock time to provide a high-quality experience to users. In the experiments, for example, the proposed algorithm was consistently observed to find a statistically optimal sequence in an average of less than 0.001 s, regardless of the number of text items in the sequence. By contrast, an exhaustive search running on identical hardware typically required more than 20 s to find the optimal sequence for a set of 10 text artifacts and required more than 250 s to perform the same task for a set of 11 text artifacts. It would be very difficult indeed to argue that a mobile app or data-driven website provides a high-quality user experience if it requires its users to wait for more than 20 s while new content is being loaded, let alone waiting for more than 250 s!

Its dramatic superiority in wall-clock time notwithstanding, the proposed algorithm was also observed to identify sequences of text artifacts whose total semantic distances were statistically indistinguishable from the globally optimal sequences. Put differently, there were no statistically significant differences between the quality of the sequences chosen by the proposed algorithm and the optimal sequences identified via an exhaustive search. Moreover, the quality percentiles of the sequences of text artifacts selected by the proposed algorithm were observed to improve according to a logarithmic function as the length of the sequences grew, increasing from an overall average of 99.35% for sequences of length $k = 5$ to an overall average of 99.96% for sequences of length $k = 11$. Given that the quality of the sequences selected by the proposed algorithm improves as the length of the sequence grows, and given that the amount of wall-clock time required for an exhaustive search increases exponentially under the same conditions, the proposed algorithm should be very attractive to app and website developers, particularly when working with larger sets of text artifacts.

In addition to the above general observations, the proposed algorithm was also observed to work equally well for all of the different types of textual content that were included in the experiments, and there is no theoretical reason to expect that the proposed algorithm would perform in any notably different manner for any other type of textual content. Moreover, the vector-space model upon which the proposed algorithm relies ensures that the algorithm will perform well regardless of how much semantic variation exists among the text artifacts in the input set. We might, for example, expect the total amount of semantic variation in a set of news articles about the FIFA World Cup to be less than the total amount of semantic variation in a set of news articles about a wider variety of topics such as politics, technology, business, and entertainment. The degree of semantic variation among the set of input articles will, of course, naturally constrain the total amount of semantic separation that can be achieved, but as long as that degree of variation is greater than zero, the proposed algorithm can be relied upon to find a high-quality solution.

The proposed algorithm has the virtues of being simple, being very fast in comparison to an exhaustive search, and being able to select statistically optimal sequences of text artifacts. Developers seeking to implement the algorithm in real-world scenarios will, however, need to consider a few important points related to the vector-space model upon which the proposed algorithm relies. First, the dimensions of the vector-space model directly influence how much of the semantic content of the underlying text artifacts is captured by each vector. The current study obtained good results with a freely available 384-dimension model, but models with more or fewer dimensions may be appropriate based on the developer's specific use case. Second, developers must decide whether to compute vector representations for and cosine distances between text artifacts in real time or in advance. For text artifacts whose content is static, computing and storing vector representations and cosine distances just once may prove advantageous, but for text artifacts whose content is subject to change, more frequent or real-time calculations may be more appropriate.

Finally, it should be noted that the proposed algorithm can also be applied to content items that are not inherently textual in nature. Modern artificial intelligence tools, for example, can readily generate textual descriptions of images using computer vision techniques [26,27] and can easily extract spoken words from video or audio recordings using speech-to-text models [28]. Generative AI tools are also capable of writing accurate textual summaries of a wide variety of multimedia artifacts [29]. Using such tools to preprocess non-textual content items can thus transform almost any form of media into a textual proxy that is suitable for input into the proposed algorithm with very little effort. Indeed, as long as a set of non-textual content items—such as images, videos, songs, or other audio

recordings—can be reduced to textual descriptions, the proposed algorithm can be used to quickly find the sequence in which those items should be presented to users in order to maximize the overall novelty from one item to the next. In this way, the proposed algorithm can be applied to find the statistically optimal sequence for a wide and mixed variety of content items, thus dramatically enhancing its potential usefulness for developers of apps and websites.

5. Conclusions

The primary goal of this paper was to address the problem of how to quickly find the optimal order in which to present a set of text artifacts to users such that the artifacts exhibit the widest possible degree of semantic separation, thereby bolstering user engagement by maximizing novelty from one artifact to the next. Although an exhaustive search could be used for this purpose, the wall-clock time required to conduct an exhaustive search increases exponentially as the number of text artifacts in the sequence grows. Since past research has established that users are unwilling to wait for more than a few seconds before being presented with new content [30–32], a solution that relies on an exhaustive search is infeasible if app and website developers hope to provide a high-quality experience to their users. The current paper therefore proposed an algorithm for quickly finding the semantically optimal presentation order for a set of text artifacts. An extensive set of experiments involving 21 different experimental conditions and more than 600 experimental trials was carried out in order to evaluate the performance of the algorithm, the results of which revealed that the proposed algorithm consistently selects sequences of text artifacts that are statistically optimal while simultaneously being extremely fast.

Although careful and methodical efforts were taken to evaluate the performance of the proposed algorithm, this study nevertheless has several limitations that merit acknowledgment. First, the proposed algorithm was evaluated using text artifacts from just three different domains—movie synopses, news articles, and song lyrics. The results of the experiments indicate that the proposed algorithm performed equally well across all of these domains, but confidence in the proposed algorithm could be improved by testing it using text artifacts from other domains. Next, the performance of the proposed algorithm was evaluated only for sequences of text artifacts of length $k = \{5, 6, \dots, 11\}$. Longer sequences were not examined in the current study due to the excessive amount of wall-clock time that would be required to conduct exhaustive searches for such sequences. Evidence obtained from the experiments suggests that the quality of the sequences selected by the proposed algorithm tends to improve as the length of the sequence grows, but whether this pattern holds for sequences of length $k > 11$ cannot be known in the absence of additional exhaustive searches. In addition to testing the algorithm in other textual domains, evaluating the algorithm's performance on longer sequences of text artifacts also represents an opportunity for future research. Moreover, it is reasonable to assume that other non-exhaustive algorithmic approaches could be developed to address the problem of maximizing inter-item novelty in a sequence of text artifacts, making the development and comparison of such algorithms another specific avenue for future work in this area.

The limitations above notwithstanding, the algorithm proposed in this paper appears to be very promising. Holding users' attention is critical for apps and websites that rely on advertising-based revenue models, and since past research suggests that novelty is strongly associated with increased attention, taking steps to ensure that the content items presented to users are as novel as possible from one item to the next has obvious advantages. It is hoped that the algorithm proposed in this paper will prove useful in that regard.

Funding: This research was generously supported by a senior intramural grant from the Office of Research and Sponsored Projects at California State University, Fullerton.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The raw data, analytical results, and Python source code for the algorithm and experiments described in this paper can be downloaded from the following URL: <https://drive.google.com/file/d/1JjLpUxZINcttmr7c8TFkZU9xJhmrOJGG/view?usp=sharing>, accessed on 15 October 2024.

Acknowledgments: This work benefited greatly from informal conversations with the author's colleagues in the Department of Information Systems & Decision Sciences at California State University, Fullerton. Their thoughts and insights are very much appreciated.

Conflicts of Interest: The author declares no conflicts of interest.

References

1. Kim, J.; Thomas, P.; Sankaranarayana, R.; Gedeon, T.; Yoon, H.-J. Pagination Versus Scrolling in Mobile Web Search. In Proceedings of the 25th ACM International on Conference on Information and Knowledge Management, New York, NY, USA, 24–28 October 2016.
2. MacDonald, D. *Practical UI Patterns for Design Systems: Fast-Track Interaction Design for a Seamless User Experience*; Apress: New York, NY, USA, 2019.
3. Tidwell, J. *Designing Interfaces: Patterns for Effective Interaction Design*; O'Reilly Media: Sebastopol, CA, USA, 2010.
4. Bernstein, G. *Unwired: Gaining Control over Addictive Technologies*; Cambridge University Press: Cambridge, UK, 2023.
5. Kelly, D.; Azzopardi, L. How Many Results Per Page? A Study of SERP Size, Search Behavior and User Experience. In Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, Santiago, Chile, 9–13 August 2015.
6. Claussen, J.; Kretschmer, T.; Mayrhofer, P. The Effects of Rewarding User Engagement: The Case of Facebook Apps. *Inf. Syst. Res.* **2013**, *24*, 186–200. [[CrossRef](#)]
7. Wu, F.; Huberman, B.A. Novelty and Collective Attention. *Proc. Natl. Acad. Sci. USA* **2007**, *104*, 17599–17601. [[CrossRef](#)] [[PubMed](#)]
8. Ernst, D.; Becker, S.; Horstmann, G. Novelty Competes with Saliency for Attention. *Vis. Res.* **2020**, *168*, 42–52. [[CrossRef](#)] [[PubMed](#)]
9. Costa, V.D.; Tran, V.L.; Turchi, J.; Averbeck, B.B. Dopamine Modulates Novelty Seeking Behavior During Decision Making. *Behav. Neurosci.* **2014**, *128*, 556–566. [[CrossRef](#)] [[PubMed](#)]
10. Duzskiewicz, A.J.; McNamara, C.G.; Takeuchi, T.; Genzel, L. Novelty and Dopaminergic Modulation of Memory Persistence: A Tale of Two Systems. *Trends Neurosci.* **2019**, *42*, 102–114. [[CrossRef](#)] [[PubMed](#)]
11. Ping, Y.; Li, Y.; Zhu, J. Beyond Accuracy Measures: The Effect of Diversity, Novelty and Serendipity in Recommender Systems on User Engagement. *Electron. Commer. Res.* **2024**, *1*–28. [[CrossRef](#)]
12. O'Brien, H.L. Exploring User Engagement in Online News Interactions. *Proc. Am. Soc. Inf. Sci. Technol.* **2011**, *48*, 1–10. [[CrossRef](#)]
13. Carmel, D.; Roitman, H.; Yom-Tov, E. On the Relationship Between Novelty and Popularity of User-Generated Content. *ACM Trans. Intell. Syst. Technol. (TIST)* **2012**, *3*, 1–19. [[CrossRef](#)]
14. Rips, L.J.; Shoben, E.J.; Smith, E.E. Semantic Distance and the Verification of Semantic Relations. *J. Verbal Learn. Verbal Behav.* **1973**, *12*, 1–20. [[CrossRef](#)]
15. Lee, D.L.; Chuang, H.; Seamons, K. Document Ranking and the Vector-Space Model. *IEEE Softw.* **1997**, *14*, 67–75. [[CrossRef](#)]
16. Deerwester, S.; Dumais, S.T.; Furnas, G.W.; Landauer, T.K.; Harshman, R. Indexing by Latent Semantic Analysis. *J. Am. Soc. Inf. Sci.* **1990**, *41*, 391–407. [[CrossRef](#)]
17. Reimers, N.; Gurevych, I. Sentence-BERT: Sentence Embeddings Using Siamese BERT-Networks. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing, Hong Kong, China, 3–7 November 2019; pp. 3982–3992.
18. IMDb. *Internet Movie Database*; Amazon: Seattle, WA, USA, 2024; Available online: <https://www.imdb.com> (accessed on 15 October 2024).
19. Box Office Mojo. *Worldwide Lifetime Gross Box Office Revenues*; IMDb Pro: Seattle, WA, USA, 2024; Available online: <https://www.boxofficemojo.com> (accessed on 15 October 2024).
20. Bird, S.; Loper, E.; Klein, E. *Natural Language Toolkit*; NLTK Project: Philadelphia, PA, USA, 2024; Available online: <https://www.nltk.org> (accessed on 15 October 2024).
21. Billboard Magazine. *Billboard Hot 100 Singles Chart*; Penske Media Corporation: Los Angeles, CA, USA, 2024.
22. LyricFind. *Song Lyrics Database*; LyricFind, Inc.: Toronto, ON, USA, 2024; Available online: <https://www.lyricfind.com> (accessed on 15 October 2024).
23. Wasserman, L. *All of Statistics: A Concise Course in Statistical Inference*; Springer: New York, NY, USA, 2013.

24. Google LLC. *Google Colaboratory*; Alphabet Inc.: Mountain View, CA, USA, 2024.
25. Welch, B.L. The Generalization of “Student’s” Problem When Several Different Population Variances Are Involved. *Biometrika* **1947**, *34*, 28–35. [[CrossRef](#)] [[PubMed](#)]
26. He, X.; Deng, L. Deep Learning for Image-to-Text Generation: A Technical Overview. *IEEE Signal Process.* **2017**, *34*, 109–116. [[CrossRef](#)]
27. Apostolidis, E.; Adamantidou, E.; Metsai, A.I.; Mezaris, V.; Patras, I. Video Summarization Using Deep Neural Networks: A Survey. *Proc. IEEE* **2021**, *109*, 1838–1863. [[CrossRef](#)]
28. Trivedi, A.; Pant, N.; Shah, P.; Sonik, S.; Agrawal, S. Speech to Text and Text to Speech Recognition Systems—A Review. *IOSR J. Comput. Eng.* **2018**, *20*, 36–43.
29. Epstein, Z.; Hertzmann, A.; Creativity, I.O.H.; Akten, M.; Farid, H.; Fjeld, J.; Frank, M.R.; Groh, M.; Herman, L.; Leach, N. Art and the Science of Generative AI. *Science* **2023**, *380*, 1110–1111. [[CrossRef](#)] [[PubMed](#)]
30. Nah, F.F.-H. A Study on Tolerable Waiting Time: How Long Are Web Users Willing to Wait? *Behav. Inf. Technol.* **2004**, *23*, 153–163. [[CrossRef](#)]
31. Lee, Y.; Chen, A.N.; Ilie, V. Can Online Wait Be Managed? The Effect of Filler Interfaces and Presentation Modes on Perceived Waiting Time Online. *MIS Q.* **2012**, *36*, 365–394. [[CrossRef](#)]
32. Hong, W.; Hess, T.J.; Hardin, A. When Filling the Wait Makes it Feel Longer: A Paradigm Shift Perspective for Managing Online Delay. *MIS Q.* **2013**, *37*, 383–406. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.