Content-based Layout Optimization

Balaji Vasan Srinivasan
balsrini@adobe.com
Adobe Research
Bangalore, India

Vishwa Vinay
vinay@adobe.com
Adobe Research
Bangalore, India

Niyati Chhaya
nchhaya@adobe.com
Adobe Research
Bangalore, India

ABSTRACT

Effective personalization of web experiences constitutes matching the intent and interest of a user (or a group of users) to content they consume, while optimizing a set of target engagement metrics. With improved content consumption tracking via web analytics, such personalization is not only feasible but also valuable for a content publisher/owner with large volumes of content to choose from. However the multitude of media (desktop, mobile, etc.) and the diversity of users' interests necessitates automation in this process of constructing personalized content experiences. In this paper, we propose a genetic algorithm based framework that chooses a subset of content items (from a large collection) that are relevant to a given user and determines their respective sizes and relative positions to construct a layout that is optimized for a chosen engagement metric. Comparisons against existing frameworks based on crowd-sourced annotations indicate improved prominence of key content (based on historic engagement metrics) by the proposed approach, while improving the information diversity of the content presented in the layout.

CCS CONCEPTS

• **Human-centered computing** → **Human computer interaction (HCI)**; • **Computing methodologies** → *Learning to rank*.

KEYWORDS

layout; content ranking; optimization

ACM Reference Format:
Balaji Vasan Srinivasan, Vishwa Vinay, and Niyati Chhaya. 2019. Content-based Layout Optimization. In *Joint Proceedings of the ACM IUI 2019 Workshops, Los Angeles, USA, March 20, 2019.* , 8 pages.

1 INTRODUCTION

With the evolution of digital technology, content is increasingly being consumed in a variety of environments ranging from desktops to mobiles to wearable devices. An enterprise with an online presence would like to engage with its customers across these channels. To do so, it would need to create engaging content and experiences appropriate for each channel, and this can be a non-trivial task. An engaging experience requires relevant, non-redundant and diverse information [26]. Once the content is available, it needs to be transformed and delivered based on the mode of consumption. Every enterprise would want to optimize certain **commercial metrics** via its digital experiences while maintaining a **high engagement level** with the target audience. The commercial metrics vary with the kind of enterprise, e.g. number of subscription sign-ups triggered from the experience for a media house, number of orders for a retailer, etc.

Currently, an enterprise employs a team of web designers and developers who put together an experience from a repository of content in a manner suitable for consumption on a chosen device by a user segment [21]. The designer/developer starts with a target metric (e.g. orders, signups) to be optimized and short-lists content items that have historically performed well on that metric. The set of content items are then laid out into an experience appropriate for the corresponding medium - for e.g., mobile layouts may be narrower with popular content items placed at the top, while desktop layouts can accommodate more items with high user interest areas reserved for the popular items. This allocation of content items to locations in the experience is an iterative process - items that gather more engagement from users may progressively be promoted to more prominent positions. Similarly, extensive experimentation (A/B testing) may be involved in identifying truly good quality content rather than items that attract user attention by merely being placed in attractive locations.

This manual process limits the extent of personalization achievable by an enterprise given the sheer volume of available content to choose from, the multitude of media for experience delivery and the variety of target consumer preferences. The effort involved multiplies when the experience is dynamic and changes in response to user activity. This calls for a mechanism to automate the process of rendering content in a given layout while optimizing a target metric across various consumers or audience segments. Such a mechanism can both provide assistance to content developers and designers, as well as aid personalized experience delivery at scale by quickly putting together renditions for varied target audience settings.

We consider the problem of automatically constructing layouts of content items (also referred to as fragments in the rest of the paper). These fragments can be of different sizes (Figure 2) and we refer to each size of the fragments as one of its variants. The target is a *home page* like experience similar to Figure 1 which displays summaries of these individual fragments that a user can click through for a 'detailed information page'. We aim to personalize this experience by automatically choosing a subset of the content from the reference collection optimized for certain metrics and also choose the appropriate size variant depending on the required emphasis for the selected content. The proposed approach is a genetic algorithm-based framework [14] where the fitness function captures various factors that contribute to the engagement on a given layout - prominence of popular items, relevance to the target user and diversity of the displayed content. We concentrate on rectangular grid-like layouts - which are common in several home page templates. The algorithm starts with a metric that we choose to maximize, a reference collection

IUI Workshops'19, March 20, 2019, Los Angeles, USA

of fragments (and their variants) and the target layout configuration (such as the size), and proceeds to select and lay out the appropriate variants of the content on the grid.

Figure 1: A sample home page layout for desktop, mobile and tablets. As would be expected, the layout, emphasis and choice of fragments changes from the desktop to mobile versions.

2 RELATED WORK

In this section, we provide an overview of papers addressing problems similar to the one described here.

Layout optimization is the process of finding a good layout efficiently and attempts to find techniques that create well serving layouts for different purposes. Gonzalez et al. [15] explored the problem of optimizing the content in a newspaper layout using simulated annealing. Barbrand [4] extended this work using a genetic algorithm framework. Duarte et al. [11] use a tree-map based structure to layout images efficiently. However, all these works focus on reducing the gaps/white-spaces in between the fragments and do not account for the content that is being laid out. Often the layouts are dictated by the historic performances and the information presented in the fragments. In our approach, we use a genetic algorithm framework to account for these content factors along with the engagement metrics to optimize the layout distribution.

Kumar et al. [19] create web pages from sample templates by identifying $1:1$ mappings between layout elements. Srinivasan et al. [27] extend this and propose *ESCORT* that maps important content to important parts of the layout based on a predefined notion of the 'criticality' of content items. However, the importance of a fragment is often determined by multiple factors and the approach in [27] cannot scale for compound importance functions. Our approach allows for such scaling across multiple content related characteristics as components of a composite 'fitness function'. We use ESCORT as one of our baselines and show improvement with our framework in both information diversity and prominence of key fragments.

Bin packing [3, 9] is a class of algorithms for the assignment of pre-selected items to locations. The underlying optimization problem starts with a set of rectangles (or any other shape) and finds the optimal way of 'packing' them into specified locations, i.e., 'bins'. Bin packing has been used for various applications including supply chain management [12], scheduling problems [22], and video-on-demand applications [31].

Bin packing can be particularly well-suited for grid layouts, due to parallels between the bins and the cells in the grid layouts. In our proposed approach, we utilize the First Fit Decreasing approximation of the bin packing algorithm [17] to define a *decoder* (defined later) and pack an ordered list of fragments into a grid. As we show later, such an approach aids in placing prominent fragments in prominent locations of the grid. Similar to methods in the earlier section, bin packing requires that each fragment be associated with a single *weight* that dictates its priority. Our method not only removes the need for such a singular characterization of the content, but also allows for considering interactions between fragments.

Another closely related problem is that of **Search Engine Result Page (SERP) composition** [29] in the context of *Aggregated Search* [18], where results from multiple verticals are blended into a single experience. The verticals are for example web, images, videos and local. Typically, one of the verticals (e.g. the web) is nominated as primary and results from the other verticals are introduced at pre-selected slots. The pre-selected slots along the primary vertical defines the layout, and results from each source need to be mapped to the slots/bins. There have been studies on what constitutes a good SERP [2, 28], with factors including the specific verticals that the results are from, and coherence across the verticals. Arguello [1] provides a comprehensive overview, including the related problem of evaluating a given SERP layout. The scenario described in the current paper shares many of the same concerns as above for two dimensional layouts [8] and we also consider the impact of inter-fragment diversity, as well as allowing for individual fragments to not be of uniform size.

Closest in spirit to the current paper are the works described in [23–25]. The application scenario considered in these papers are interactivity assistance in the construction of graphic designs. With input from design principles, factors capturing design quality can be encapsulated into an *energy function* that can then be optimized. The fitness function commonly used within a genetic algorithm setup serves a similar purpose. In the current paper, we have not considered aesthetics - focusing instead on individual item properties (relevance) and inter-item considerations (diversity). An advantage of the framework described here is however that it is easily extensible by extending the fitness function to include other factors.

While layout optimization and content distribution in a 2-d space have been independently explored, to the best of our knowledge, automatic laying-out of content with simultaneous optimization of engagement and content metrics across mediums is less studied and is our primary contribution.

3 GENETIC ALGORITHM FRAMEWORK

In this section, we describe the problem in detail so as to motivate the use of genetic algorithms as the solution approach. Revisiting the problem definition, we have a set of candidate content items, each of which is called a 'fragment'. Each fragment is a representation of the content to be used for the experience and a subset of these fragments need to be chosen for the target experience. We assume that each fragment includes some content - e.g. text in the form of a title or description, or images and other visual content. Other metadata,

Figure 2: Content Element/Fragment and its different variants considered in our analysis. Due to the proprietary nature of the underlying content used for the layout, we use an 'indicative' image and text here. However, all our experiments included the actual content from a popular media platform.

such as the historical performance of the content in terms of user engagement, may also be available. The fragments are considered to have different variants - each of different size. Figure 2 shows different variants of the fragments considered in our paper. Such variants would typically be defined via a template used commonly across several fragments. Layout optimization involves identifying a subset of the provided fragments that make the layout, as well as the position and relative sizes of the individual fragments. Note that we take the fragments and their constituent content to be a given, and optimize their assembly into a layout. Optimizing the aesthetics of the layout is beyond the scope of our exploration.

The target layout is described in terms of its shape and size, and here we focus on grid-shaped rectangular layouts defined by the number of rows (N_R) and columns (N_C). Such a layout can accommodate a maximum of $L\ \ N_R \times N_C$ fragments, each of unit size. The set of displayed fragments might be fewer than L if larger-sized fragments have been chosen - for example, popular/highly relevant content might deserve more screen area. The algorithm also simultaneously optimizes an engagement metric like the expected number of clicks on fragments.

Fragments that are likely to make larger contributions to the engagement are preferred in regions of the screen that attract most attention. Existing literature (e.g. [5]) provides guidance about where these might be. A strategy to maximize engagement would be to place the content items with most historical engagement in the locations corresponding to high user attention. While intuitive, this greedy strategy has the downside that it does not account for inter-content characteristics and might result in a situation where two very similar fragments are placed next to each other - this can be undesirable from a user's perspective.

The optimization problem is therefore complicated by having to account for interactions among content items. Two layouts with the same set of fragments that are different only in terms of the positions of the fragments might have very different characteristics from the users' perspective due to the difference in the user attention to different fragments [5].

A concrete illustration of the scenario considered in the current paper is the creation of genre/category level pages at a site like IMDB. In fact, the experiments described in Section 4 address this setting. Given a genre (e.g. 'action', 'comedy', 'drama'), we could utilize a historical metric (e.g. revenue or ratings) to pick movies in order of their popularity. Given constraints on the number of items that could be included (due to the size of the layout), we might decide to display the top-$\mathcal{L}$ most popular movies. If the list is to be personalized, the global popularity needs to be balanced against relevance to the individual. We might decide to allow some of the movies to be present in tiles of a larger size - e.g. allow the top-ranked movie to take up 2 slots in the grid layout and therefore drop the bottom ranked item from the shortlist. Also, to ensure diversity in the displayed fragments, an item of lower popularity might be promoted into the layout. Therefore, the layout construction algorithm has to not only choose the set of items, but also their individual positions and sizes, to ensure that the resulting experience is engaging.

The discontinuous nature of the perceived quality of the layout with respect to these configurations, coupled with the large search space means that this problem does not lend itself easily to traditional optimization approaches. To account for this, we propose a genetic algorithm framework. Genetic algorithms [14, 30] are a class of computational approaches that are typically concerned with problems that are non-linear, and where it is not possible to treat each parameter as an independent variable that can be solved in isolation from the other variables. The proposed genetic algorithmic framework simultaneously optimizes the performance metrics of experience fragments along with content-affinities and diversity properties.

This framework begins with the assignment of a **Chromosome** - which is a representation of the layout that is being optimized. We initialize this to a vector of size of $\mathcal{L}$ and with each entry being an identifier to a unique fragment. The vector is initialized to the set of fragments in descending order of their respective metrics. Since the distribution is pivoted on this initialization, we ensure that the initial distribution has the best (engagement) metric combination.

Central to the genetic algorithm framework is the notion of a **fitness function**, which takes the current configuration, represented by the Chromosome, as input and outputs a number indicating the quality of the layout that would result from displaying this set of fragments laid out as controlled by the **decoder** (described in detail later). Apart from the metric tied to each fragment, the fitness function also accounts for the inter-content affinities, expected relevance to the user, overall content diversity, and coverage of all possible aspects and concepts across the content in the input fragment set, motivated by their use in information retrieval scenarios [10]. The resulting fitness value gives an objective measure of the goodness of the distribution of fragments in the layout across the content parameters and the target metric. The fitness function is thus a mechanism within the framework that allows easy extensibility. Note that the different factors affecting the fitness may not agree with each other. For example, the best collection of fragments as per the individual metrics might have very low inter-fragment diversity - and hence may not be the optimal experience for the consumer (due to redundant information). Our fitness function addresses this by including a variety of factors to determine the 'goodness' of the collection of fragments.

The decoder takes an ordered list of fragments along with the layout configurations to distribute the fragments currently within the Chromosome on to the layout. To ensure that a sufficient variety of configurations of fragments is explored, a **mutation** operation needs to be defined that makes alterations to the existing Chromosome. If a particular mutation leads to an increase in the fitness value, the corresponding change is retained. In the proposed framework, the purpose of mutation is 2-fold. Firstly, it aids in the exploration of different permutations of the input set of fragments thus evolving the chromosome/layout to a distribution that has better overall fitness. Additionally, a factor is used to insert fragments that would have otherwise not featured in the distribution. This not only allows newer fragments with no performance metrics to feature in the distribution but can also introduce better content diversity. After exploring alternative configurations in this manner across many iterations/generations, the Chromosome with the highest fitness can be identified, and this is taken to be the optimized layout of the content.

Repeating this sequence of steps for every target user (or user segment) for the respective metric (based on the consumption environment) yields the personalized experience for every possible combination. Algorithm 1 outlines the different steps in the proposed framework. In the subsequent subsections, we elaborate the key components of our framework: decoder, fitness computation and mutation.

Algorithm 1: Genetic Algorithm Framework for content-based layout optimization

```
Input: 𝓕: Set of all fragments, 𝓜: metric to optimize, 𝓛:
       Layout size to optimize
Output: Decoded Layout
Layout ← Chromosome𝓕, 𝓜, 𝓛;
Layout.fitness ← computeFitnessLayout;
curGeneration ← 0;
repeat
    tempLayout ← mutateLayout;
    tempLayout.fitness ← computeFitnesstempLayout;
    if (tempLayout.fitness-Layout.fitness)>0 OR
     RandomNumber < δ then
        Layout ← tempLayout;
    end
    curGeneration ;
until curGeneration > totalGeneration;
return decodeLayout
```

3.1 Decoder

The Chromosome corresponds to a set of input candidate fragments that need to be allocated a location in the layout and is an ordered set of fragments. The decoder controls how the linear ordered collection of fragments is arranged according to the target layout dimensions provided by the designer. It takes the fragments (each of a currently chosen size) in the order defined by the Chromosome along with the layout dimensions and outputs the arranged layout. This allocation can be simple (e.g. via a reverse raster scan), but should be able to handle the fact that fragments potentially are of differing sizes. In this paper, we use a decoder inspired by the bin packing algorithm[20] which is a variant of the knapsack algorithm in 2−dimensions. We extend the complete bounding box-based strategy using the First Fit Decreasing (FFD) algorithm[16].

The FFD algorithm starts with a ranked set of fragments (as defined by the Chromosome) and iteratively creates an updated layout (bin dimension) and an updated fragment list (of all fragment variants that have not been placed on the layout yet) till the point all locations on the layout have been filled. We leverage the study in [5] to assign an exponentially-decaying weight/importance to every bin in the layout based on their 2-dimensional location. The bins are then filled in the order of the assigned importance - thus ensuring that important parts of the layout are filled before the other parts. Utilizing the ranking from the Chromosome ensures achieving the target layout with the key fragments in key locations. Figure 3 shows a schematic diagram for the algorithm.

It is easy to see that the decoding is a highly discontinuous operation. Making a minor alteration in the Chromosome - e.g. via a mutation - potentially alters the fitness of the resulting layout in non-trivial ways. The proposed genetic algorithm framework addresses this by decoupling the optimization and fitness computation. Running the framework across several generations explores various layout against the fitness function to achieve a stable optimized layout. The proposed framework is generic since it allows the fitness function to be a black box, and ensures that the resulting layout puts the key fragments in key locations.

3.2 Fitness function

Given a spatial arrangement of fragments (i.e., output of the Decoder), the fitness function produces a numeric value indicating how desirable the given configuration is. In that sense, the fitness computation needs to encapsulate factors that affect the quality of a particular layout. We use [10] to help design the components of this fitness function. In [10], set in the Information Retrieval domain, an initial set of results for a user's query are available. Based on this evidence, we may decide to add new terms to the query that were not provided explicitly by the user ('*Query Expansion*'), expecting that these will likely return more relevant results. The risk of doing so is that the added terms do not match the user's intent, thereby

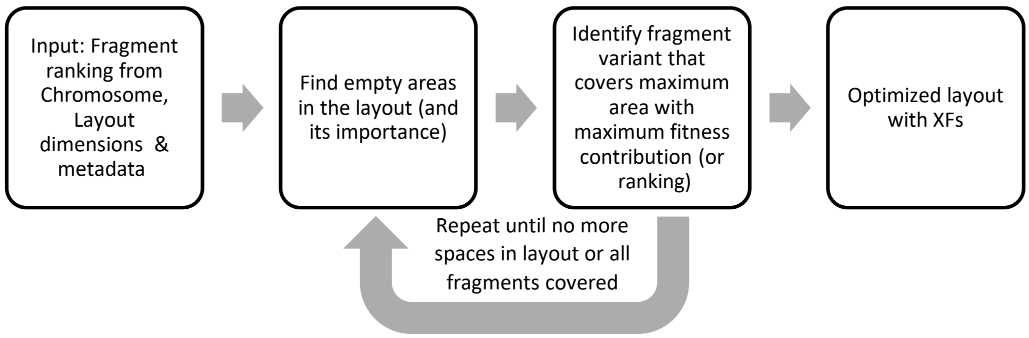

Figure 3: Sequence of steps in the Bin Packing based decoder

leading to a drop in quality of results. The author uses 'coverage' and 'balance' across topics/aspects, and trades these off against relevance. We follow a similar template. Our proposed algorithm includes the following factors in our fitness computation to account for different aspects of the layout.

(1) **Metric-contribution** indicates the total expected value of the engagement metric for this chosen set of fragments displayed in the given arrangement. Since we would like to pick a layout that maximizes the metric value, we would choose to place fragments that have historically seen larger values of the metric at locations that are most likely to receive user attention. We compute a weighted average of individual fragment metrics, where the weight is based on its current assigned location in the 2-dimensional and represents user attention based on an exponential decay function as in the user study in [5]:

$$\exp - \frac{i^2 \quad j^2}{{N_R}^2 \quad {N_C}^2} * M \qquad (1)$$

for an item that the Decoder is currently placing at row i and column j on the grid, and M is the value of the engagement metric for this item.

(2) While the metric captures historical performance across a set of users, it is important to ensure that all the fragments that are part of the current display include those that are relevant to the current user. The fitness function therefore includes the fragment's **relevance** to ensure that utility of the fragments to the target user. This component helps achieve personalization since the computed fitness is now dependent on a particular user. When the fragments have textual content associated with them, relevance can be computed by standard measures (e.g. tf-idf) with respect to the user profile - based on historic preferences.

(3) The candidate set of fragments span topical concepts (extracted from the text in the title and description). The fitness computation attempts to ensure that the chosen set in the layout (represented by L) is representative of the larger candidate set (represented by C) in the sense that:

(a) the important topic/concepts in the candidates are present in the chosen set, this is termed as *coverage*. We define coverage as

$$\frac{{}_{i\in L}\;{}_{a\in A}\; x_{ia}}{||L||||A||} \qquad (2)$$

where x_{ia} is the weight of concept attribute a to item i

(b) where possible, all topics/concepts in the full set of candidates are represented in the final chosen set, this is termed as *balance*. We define balance as

$$_{a\in A}\left(\frac{{}_{i\in C}\; x_{ia}}{||C||} - \frac{{}_{i\in L}\; x_{ia}}{||L||}\right) \qquad (3)$$

We extract the key entities in the fragment (from the title and description) as its concepts. The combination of coverage and balance over the attributes yields a diverse selection of fragments that covers information about different aspects of the initial set. Increasing diversity is a well known strategy that reduces the risk of producing a layout with no relevant content for a given user [7].

We compute the fitness as a weighted-sum of these factors to get an estimated utility/quality of the layout. Note that our framework is generic - other factors that we expect to contribute to the quality of a particular layout can be readily incorporated, with the rest of the mechanism within the framework designed to optimize the net fitness.

3.3 Mutation

Genetic algorithms operate by constructing multiple layout configurations, and evaluating each via the fitness function. Mutation is the process of taking the current configuration (represented by the Chromosome) and producing an altered version of it. This step is used to explore alternate configurations to hopefully identify a more optimal (as per the fitness function) layout. In the proposed approach, the mutation component includes one of the following 3 steps:

(1) With probability α_1, choose fragment in the top 20-percentile of the fragments ordered by their individual metrics, and select its variants whose $Mcost > \Sigma$, where M is the metric value as before, $cost$ is its size. The current nominated size of the fragment is taken to be the largest one that satisfies the above constraint. This ensures that apart from choosing a good set of fragments and their corresponding positions, promising fragments are given a larger share of the layout (screen) real estate.

(2) With probability α_2, exchange the position of two chosen fragments in the chromosome. Via the Decoder, the fragment's location in the final displayed layout will now be different. This alternative, aids in exploring different permutations of fragments in the same Chromosome, i.e., alternate locations for each fragment.
(3) With probability α_3, swap one of the fragments in the chosen set with a fragment that is in the larger candidate set but not in the layout right now. This alters the set of fragments that will be in the layout.

In our experiments, we empirically set $\alpha_1 \quad 0.3, \alpha_2 \quad 0.6, \alpha_3 \quad 0.1$. Exploring different sets of fragments, as well as different configurations, attempts to ensure that the final chosen layout is of high quality. The key idea in the genetic algorithmic framework is to run the mutations over a large number of iterations (generations) and this will yield a stable generation that has an optimal fitness across the various parameters of evaluation.

4 EVALUATION

To demonstrate how the algorithm works, we used the set of fragments based on the top 1000 movies in IMDB for 2006 to 2016 from Kaggle[1]. In our dataset, every movie's data included its title, a short description about the movie, its genre, its IMDB Rating, IMDB User Votes along with its revenue. We used the movie revenue as a surrogate for its historic performance. The movie description and genre were used to determine the content level features in our fitness function. For our experiments, we defined variants of the fragment with sizes (height-by-width) - 2x2, 2x1, 1x2 and 1x1 as shown in Figure 2. Since the proposed algorithm does not modify the internals of the fragments (or its variants), we assume that the aesthetics of the fragments are not modified with these variants.

We constructed different layouts based on the movies belonging to the major genres (Action, Adventure, Comedy, Crime, Drama, Mystery, Romance, Thriller) in the dataset. Layouts of sizes suitable to a mobile environment (e.g. 4x3, 5x3) were defined with shorter columns and the desktop grids were defined with more columns. Every grid is then populated by each of the candidate methods. We extend the ESCORT approach in [27] as our primary baseline for comparisons. ESCORT defines a criticality/ranking for different 'slots' in the layout and we use [5] to define this in our setup. We use the Maximum-Marginal Relevance based diversified ranking [6] to define the criticality of the fragments. Additionally, we also include the version of the proposed Genetic Algorithm (GA) where our fitness function was based on the target metrics and relevance to the user (without the diversity considerations). Finally, we also included a random layout rendition - thus resulting in 4 distinct layouting methods.

To evaluate the 'goodness' of the proposed layout, we conducted two different experiments on Amazon Mechanical Turk[2]. Annotators were limited to those who have over 100 completed hits with 95% acceptance rates. Additionally, since the dataset is based on Hollywood movies, we limited the annotators' geo to US only.

As mentioned previously, the layout optimization algorithm for personalized delivery optimizes a target metric. This calls for the fragments that have performed well on the target metric to be placed in prominent locations. In our approach, a bin packing based decoder together with the fitness ranking ensures that the good-performing fragments are placed in key locations. To evaluate this, our first survey is aimed at capturing the perceived importance of the various fragments at different locations in the layout. Every layout produced by each of the methods were annotated by 5 distinct users - each user annotating the prominence of 4 fragments at different locations in the layout generated by all 4 candidate methods - resulting in 16 annotations for this category per annotator.

However, as discussed before, optimizing just based on good-performing fragments can lead to redundant layouts. Our approach avoids this by having additional content-based metrics in our fitness function to introduce diversity of the information in the fragments. To evaluate this, our second survey asked annotators to rate the 'information diversity' of layouts from two different algorithms. Repeating this across different combinations yields a preference ranking of different algorithms in terms of the diversity that they afford. Every method combination is annotated by 5 different annotators, with every annotator annotating all 6 possible combinations.

Thus every annotator recorded 22 total annotations. We paid 25 cents per worker, and the work took 4 minutes on an average. Initializing the different layouts based on one of the 8 genres mentioned above allows for debiasing any content-induced biases in the annotation. To avoid any grid-size induced bias, we expose every annotator to a single grid-size only. Finally, ordering of the methods in the two tasks were shuffled to safeguard against positional/order induced biases.

4.1 Prominence of key fragments

The first experiment measures the perceived importance of various fragments at different locations in a layout. Figure 4 shows a screen shot of the performed evaluation where we asked the annotators to annotate the perceived importance of a highlighted part of the layout on a scale of $0 - 100$.

For every layout, we calculate the Pearson's correlation ρ between the human-annotated perceived importance of the fragment in the layout and its actual metric. Table 1 shows the correlation between the perceived importance and the metric value for the proposed method (GA+diversity) against the various approaches including the one in [27] - ESCORT.

The average correlation indicate that the 3 approaches (ESCORT, GA-Relevance, GA-Diversity) exhibit a good correlation between the perceived importance of the fragments against its metrics unlike the Random layout (as would be expected). Further, note that both the frameworks based on the proposed approach (GA and GA+diversity) beat ESCORT [27] in terms of the overall correlation - indicating that the proposed approach does well in placing key fragments in prominent locations. Between GA and GA+Diversity, the framework without diversity performs better since it specifically optimizes the metric (prominence).

4.2 Content Diversity in the layout

The additional factors in our fitness function further ensures a diverse set of fragments being shown in the layout to avoid content redundancy. To measure this, we further asked the human annotators to

[1]https://www.kaggle.com/PromptCloudHQ/imdb-data
[2]https://www.mturk.com/

	ESCORT	GA-Relevance	GA-Diversity	Rand
All Grids	0.3161	0.4297	0.3777	0.0565
Mobile Grids	0.3874	0.5014	0.4785	0.0468
Web Grids	0.2319	0.3451	0.2587	0.0679

Table 1: Pearson's Correlation between 'perceived importance' of a movie fragment against its actual metric (Revenue) across multiple layout settings.

Figure 4: Survey to get the perceived importance of a fragment in the layout. Every annotator was asked to annotate the perceived importance of the indicated fragment based on their layout position. Table 1 shows the statistics around the ratings from the annotators.

compare the content diversity between two layouts. Figure 5 shows the corresponding human experiment.

Figure 5: A screen shot of the survey to compare diversity of layouts from various algorithms. Every annotator was asked to compare the layouts from two algorithms in terms of information diversity. Table 2 shows the statistics around the ratings from the annotators across different pairs.

We compute the WMW statistic (extended from learning-to-rank problems [13]) to measure the fraction of all possible layout pairs where one layout is marked more diverse than the other by the annotators. A lower WMW statistic indicates a relatively less diverse layout against the compared approach. The statistics for various configuration is summarized in Table 2. A value of WMW statistic > 0.5 indicate that more than half of the total annotations favored Method 1 over Method 2.

It can be seen that the proposed approach (GA+Diversity) consistently outperforms the other approaches in terms of information diversity. Across all the settings, approximately 60% of the annotators rated our approach to be more diverse over ESCORT [27]. Note that a random selection of fragments was rated to be more diverse against all candidate approaches. This is expected since the variety in the dataset yields such diversity. While diversity is preferrable, this comes at the cost of prominence to key fragments as indicated by Table 1.

Method 1 Method 2	GA-Diversity GA-Relevance	GA-Diversity ESCORT	GA-Relevance ESCORT	Random GA-Diversity	Random GA-Relevance	Random ESCORT
All Grids	0.6825	0.6508	0.7143	0.6825	0.7143	0.7460
Mobile Grids	0.7941	0.7941	0.9412	0.5588	0.6765	0.7059
Web Grids	0.5517	0.5272	0.5517	0.8276	0.7586	0.7931

Table 2: WMW Statistic between pairs of layouts as annotated by annotators towards diversity computation. Every cell indicates the WMW statistic (fraction of annotators rating the Method 1 to produce more diverse layout than the Method 2.

Tables 1 and 2 establish that the proposed approach produces layouts with diverse information without compromising on the prominence for key fragments.

5 CONCLUSION

We have presented an algorithm to automatically layout a set of fragments across different layouts by simultaneously optimizing for different metrics and content factors. Human annotations of the resulting distribution indicate superior performance of the proposed approach over existing baselines. Future work includes further qualitative experiments to extract aspects of layouts that govern users' engagement with them. Once these have been identified, these factors would need to be incorporated into the layout generation algorithm.

REFERENCES

[1] Jaime Arguello. 2017. Aggregated Search. *Foundations and Trends in Information Retrieval* (2017).
[2] Jaime Arguello and Robert Capra. 2012. The Effect of Aggregated Search Coherence on Search Behavior. In *21st ACM International Conference on Information and Knowledge Management (CIKM)*.
[3] J. O. Berkey and P. Y. Wang. 1987. Two-Dimensional Finite Bin-Packing Algorithms. *Journal of the Operational Research Society* (1987).
[4] Gjermund Bø Brabrand. 2008. Dynamic layout optimization for newspaper web sites using a controlled annealed genetic algorithm. *Masters Thesis, Gjovik University College* (2008).
[5] Georg Buscher, Edward Cutrell, and Meredith Ringel Morris. 2009. What do you see when you're surfing?: using eye tracking to predict salient regions of web pages. In *SIGCHI Conference on Human Factors in Computing Systems*.
[6] Jaime Carbonell and Jade Goldstein. 1998. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 335–336.
[7] Harr Chen and David R. Karger. [n. d.]. Less is More: Probabilistic Models for Retrieving Fewer Relevant Documents. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*.
[8] Flavio Chierichetti, Ravi Kumar, and Prabhakar Raghavan. 2011. Optimizing Two-dimensional Search Results Presentation. In *4th ACM International Conference on Web Search and Data Mining (WSDM)*.
[9] E. G. Coffman, Jr., M. R. Garey, and D. S. Johnson. 1997. Approximation Algorithms for NP-hard Problems. Chapter Approximation Algorithms for Bin Packing: A Survey.
[10] Kevyn Collins-Thompson. [n. d.]. Reducing the Risk of Query Expansion via Robust Constrained Optimization. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management*.
[11] Felipe SLG Duarte, Fabio Sikansi, Francisco M Fatore, Samuel G Fadel, and Fernando V Paulovich. 2014. Nmap: A novel neighborhood preservation space-filling algorithm. *IEEE Transactions on Visualization & Computer Graphics* 12 (2014), 2063–2071.
[12] Ugur Eliiyi and Deniz Eliiyi. 2009. Applications of Bin Packing Models Through The Supply Chain. (2009).
[13] Glenn Fung, Rómer Rosales, and Balaji Krishnapuram. 2006. Learning Rankings via Convex Hull Separation. In *Advances in Neural Information Processing Systems 18*.
[14] David E Goldberg. 2006. *Genetic algorithms*. Pearson Education India.
[15] Jesús González, Ignacio Rojas, Héctor Pomares, Moisés Salmerón, and Juan Julián Merelo. 2002. Web newspaper layout optimization using simulated annealing. *IEEE Transactions on Systems, Man, and Cybernetics* (2002).
[16] Eric Huang and Richard E Korf. 2013. Optimal rectangle packing: an absolute placement approach. *Journal of Artificial Intelligence Research* (2013).
[17] Eric C Huang and Richard E. Korf. 2013. Optimal Rectangle Packing: An Absolute Placement Approach. *J. Artif. Intell. Res.* (2013), 47–87.
[18] Arlind Kopliku, Karen Pinel-Sauvagnat, and Mohand Boughanem. 2014. Aggregated Search: A New Information Retrieval Paradigm. *Comput. Surveys* (2014).
[19] Ranjitha Kumar, Jerry O Talton, Salman Ahmad, and Scott R Klemmer. 2011. Bricolage: example-based retargeting for web design. In *SIGCHI Conference on Human Factors in Computing Systems*. ACM.
[20] EG Co man Jr, MR Garey, and DS Johnson. 1996. Approximation algorithms for bin packing: A survey. *Approximation Algorithms for NP-Hard Problems* (1996).
[21] Jakob Nielsen. 1999. User Interface Directions for the Web. *Commun. ACM* 42, 1 (Jan. 1999), 65–72.
[22] Taiwo O. 2015. Bin Packing Algorithms with Applications to Passenger Bus Loading and Multiprocessor Scheduling Problems. (2015).
[23] Peter O'Donovan, Aseem Agarwala, and Aaron Hertzmann. 2014. Learning layouts for single-page graphic designs. *IEEE transactions on visualization and computer graphics* 20, 8 (2014), 1200–1213.
[24] Peter O'Donovan, Aseem Agarwala, and Aaron Hertzmann. 2015. Designscape: Design with interactive layout suggestions. In *Proceedings of the 33rd annual ACM conference on human factors in computing systems*. ACM, 1221–1224.
[25] Seonwook Park, Christoph Gebhardt, Roman Rädle, Anna Maria Feit, Hana Vrzakova, Niraj Ramesh Dayama, Hui-Shyong Yeo, Clemens N Klokmose, Aaron Quigley, Antti Oulasvirta, et al. 2018. AdaM: Adapting Multi-User Interfaces for Collaborative Environments in Real-Time. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM, 184.
[26] Filip Radlinski, Paul N. Bennett, Ben Carterette, and Thorsten Joachims. 2009. Redundancy, Diversity and Interdependent Document Relevance. *SIGIR Forum* 43, 2 (Dec. 2009).
[27] Balaji Vasan Srinivasan, Tanya Goyal, Varun Syal, Shubhankar Suman Singh, and Vineet Sharma. 2016. Environment Specific Content Rendering & Transformation. In *Companion Publication of the 21st International Conference on Intelligent User Interfaces*. ACM.
[28] Shanu Sushmita, Hideo Joho, Mounia Lalmas, and Robert Villa. [n. d.]. Factors Affecting Click-through Behavior in Aggregated Search Interfaces. In *19th ACM International Conference on Information and Knowledge Management (CIKM)*. 519–528.
[29] Yue Wang, Dawei Yin, Luo Jie, Pengyuan Wang, Makoto Yamada, Yi Chang, and Qiaozhu Mei. 2016. Beyond ranking: Optimizing whole-page presentation. In *Proceedings of the 9th ACM International Conference on Web Search and Data Mining*.
[30] Darrell Whitley. 1994. A genetic algorithm tutorial. *Statistics and computing* (1994).
[31] Eduardo Xavier and Flavio Keidi Miyazawa. 2006. The Class Constrained Bin Packing Problem with Applications to Video-on-Demand. (2006).