Improving Aggregated Search Coherence

Jaime Arguello

University of North Carolina at Chapel Hill
jarguello@unc.edu

Abstract. Aggregated search is that task of blending results from different search services, or *verticals*, into the core web results. Aggregated search coherence is the extent to which results from different sources focus on similar senses of an ambiguous or underspecified query. Prior research studied the effect of aggregated search coherence on search behavior and found that the query-senses in the vertical results can affect user interaction with the web results. In this work, we develop and evaluate algorithms for *vertical results selection*—deciding which results from a particular vertical to display. Results from a large-scale user study suggest that algorithms that improve the level of coherence between the vertical and web results influence users to make more productive decisions with respect to the web results—to engage with the web results when at least one of them is relevant and, to a lesser extent, to *avoid* engaging with the web results otherwise.

1 Introduction

Commercial search portals such as Google, Bing, and Yahoo! provide access to a wide range of specialized search services or *verticals*. Example verticals include search engines for a specific type of media (images, videos, books) or a specific type of search task (search for news, local businesses, on-line products). The goal of *aggregated search* is to integrate results from different verticals into the core web results. From a system perspective, aggregated search is a two-part task: (1) predicting *which* verticals to present for a given query (*vertical selection*) and (2) predicting *where* to present those verticals selected (*vertical presentation*). Typically, a vertical is presented by blending a few of its top results somewhere in the first page of web results.

In this work, we study a phenomenon called *aggregated search coherence.* Given an ambiguous or underspecified query (e.g., "saturn"), a common strategy for a search engine is to diversify its results (e.g., to return results about "saturn" the planet, the car, and the Roman god). Aggregated search coherence is the extent to which results from different sources focus on similar senses of the query. Suppose that a user enters the query "saturn" and the system decides to integrate image vertical results into the web results. If the web results focus on the car, but the blended images focus on the planet, then the aggregated results have a *low* level of coherence. Conversely, if both sets of results focus on the same query-sense(s), then the aggregated results have a *high* level of coherence.

Prior work investigated the effects of aggregated search coherence on search behavior. Specifically, Arguello and Capra [2, 4, 3] found that users are more likely to interact with the web results when the vertical results are more consistent with the user's intended query-sense. That is, a user looking for "saturn" the car is more likely to interact with the web results if the vertical results blended on the SERP include results about the car versus the planet. This is referred to as a "spill-over" effect. The spill-over effect suggests that while the vertical results come from a completely independent system, they can still influence user engagement with other components on the SERP (e.g., the web results).

Modeling cross-component effects is an important, yet understudied problem in aggregated search. If a user wants results from multiple sources (e.g., vertical and web results) or wants web results instead of vertical results, it is important for the system to display vertical results that show how the vertical is relevant to the query, but do not negatively affect user engagement with other components on the SERP. In this paper, we evaluate algorithms for *vertical results selection*—deciding which results from a particular vertical to display. We focus on algorithms that improve the level of coherence between the vertical and web results and show that these methods avoid negatively affecting user engagement with the web results.

There are two ways in which incoherent vertical results can negatively affect user engagement with the web results. First, if the vertical results contain the user's intended query-sense, but the web results do not, then the vertical results may influence the user to engage with the web results in vain. A more productive decision would be to quickly reformulate the query. Second, if the vertical results *do not* contain the intended query-sense, but the web results do, then the vertical results may influence the user to unnecessarily reformulate the query. A more productive decision would be to engage with the web results. If we treat user engagement with the web results as a binary decision, then these two situations represent false-positive and false-negative decisions by the user, respectively.

We evaluate several different vertical results selection algorithms across four verticals: images, news, shopping, and video. Results from a large-scale user study suggest that algorithms that improve the level of coherence between the vertical and web results influence users to make more productive decisions with respect to the web results—to engage with the web results when there is a relevant web result on the SERP and, to a lesser extent, to *avoid* engaging with the web results otherwise.

2 Related Work

Current methods for aggregated search prediction and evaluation do not *explicitly* favor coherent results. Algorithms for vertical selection and presentation use machine learning to combine a wide range of features. Prior work investigated features derived from the query string [6, 11, 14, 15], from the vertical results [5, 6, 10, 11], from the vertical query-log [5, 6, 10, 11], and from historic click-through rates on the vertical results [14, 15]. None of these features consider the relationship between the vertical results and those from other components on the SERP. Evaluation methods for aggregated search fall under three categories: on-line, test-collection, and whole-page evaluation methods. On-line methods are used to evaluate systems in a live environment using implicit feedback (i.e., vertical *clicks* and *skips*). One limitation of these methods is all false positive vertical predictions (signaled by a *skip*) are treated equally. Prior work found that, depending on the vertical results, displaying a non-relevant vertical can also affect engagement with other components on the SERP [2, 4, 3]. An aggregated search test-collection includes a set of queries, cached results from different sources, and relevance judgements. Zhou *et al.* [21] proposed an evaluation metric that considers three distinguishing properties between verticals: (1) its relevance to the task, (2) the visual salience of the vertical results, and (3) the effort required to assess their relevance. Our research suggests a fourth aspect to consider: the expected spill-over from the vertical results to other components. Bailey *et al.* [7] proposed an evaluation method that elicits human judgements on the whole SERP. While cross-component coherence is mentioned an important aspect of whole-page quality, its effect on search behavior was not investigated.

Incoherent results occur when the different aggregated components focus on different senses of an ambiguous query. A natural question is: How often does this happen? Sanderson [17] analyzed a large commercial query-log and found that about 4% of all unique queries and 16% of all unique head queries corresponded to ambiguous entities in Wikipedia and WordNet. This result suggests that ambiguous queries are common. Given an ambiguous query, incoherent results are more likely when results from different sources favor different senses. The analysis by Santos *et al.* [19] suggests that this is often the case. Santos *et al.* considered the different senses for a set of ambiguous entities and compared their frequencies in query-logs from a commercial web search engine and three verticals. Results found that different sources are often skewed towards different senses (e.g., the shopping vertical had more queries about "amazon" the company, while the images vertical had more queries about the rainforest).

One strategy for improving aggregated search coherence is to diversify results from different components across similar query-senses. Approaches for search result diversification fall under two categories: *implicit* and *explicit*. Implicit approaches diversify results by minimizing redundancy in the top ranks [8]. Explicit approaches diversify results by directly targeting results about different aspects of the query. Prior work investigated predicting the different query-aspects using topic categorization [1], a clustering of the collection [9], query reformulations in a query-log [16], and query suggestions from an on-line "related queries" API [18]. In this work, we focus on methods for selecting vertical results on the same query-senses as the web results and include Maximal Marginal Relevance [8] (an implicit diversification method) as a baseline for comparison.

3 Algorithms for Vertical Results Selection

Preliminaries. We describe our algorithms using the following notation. First, we assume that each vertical v is associated with some number τ_v of results that are blended into the web results if the vertical is presented. We considered four verticals. For the images, shopping, and video verticals, $\tau_v = 5$. For the news vertical, $\tau_v = 3$. Let $\mathcal{R}_q^v$ denote the original retrieval from vertical v in response to query q. All the algorithms described below take $\mathcal{R}_q^v$ as the input and produce a new ranking denoted as $\tilde{\mathcal{R}}_q^v$. The goal for the system is to decide which t_v results from $\mathcal{R}_q^v$ to include in $\tilde{\mathcal{R}}_q^v$ and in what order. Next, let $\mathcal{R}_q^w$ denote the top 10 web results for query q and $\tilde{\mathcal{R}}_q^w$ denote a diversified re-ranking of $\mathcal{R}_q^w$. One of our algorithms uses $\tilde{\mathcal{R}}_q^w$ internally to diversify the vertical results. Finally, let $\mathcal{R}_q^*(k)$ denote the result at rank k in $\mathcal{R}_q^*$.

All the algorithms described below require measuring the similarity between pairs of web and/or vertical documents. This similarity function is denoted as $\phi(d_i, d_j)$ and is explained later.

Maximal Marginal Relevance. MMR diversifies results by minimizing redundancy in the top ranks [8]. Given an initial ranking $\mathcal{R}_q$, it constructs a new ranking $\tilde{\mathcal{R}}_q$ by iteratively appending documents that are similar to the query (relevant) and dissimilar to those already in $\tilde{\mathcal{R}}_q$ (novel).

Our implementation of MMR assumes that the relevance of every vertical result in $\mathcal{R}_q^v$ is constant. Thus, vertical results are appended to $\tilde{\mathcal{R}}_q^v$ solely based on their dissimilarity to those already in $\tilde{\mathcal{R}}_q^v$. We first initialize $\tilde{\mathcal{R}}_q^v$ by appending the top vertical result in $\mathcal{R}_q^v$ and then iteratively append vertical results from $\mathcal{R}_q^v$ with the lowest similarity with the most similar ones already in $\tilde{\mathcal{R}}_q^v$.

MMR may improve coherence if the web results are diversified and the top vertical results are initially skewed towards a particular query sense. However, MMR selects vertical results *independently* from the web results. The next three approaches explicitly select vertical results that are similar to the web results.

Web Similarity. WebSim (Algorithm 1) aims to diversify the vertical results in $\tilde{\mathcal{R}}_q^v$ across the same query-senses in the top τ_v web results. Specifically, it iteratively appends vertical results to $\tilde{\mathcal{R}}_q^v$ such that $\tilde{\mathcal{R}}_q^v(k)$ corresponds to the vertical result in $\mathcal{R}_q^v$ most similar to $\mathcal{R}_q^w(k)$ (lines 3-6).

Algorithm 1 Web Similarity

```
   WebSim($\mathcal{R}_q^v, \mathcal{R}_q^w, \tau_v$)
1: $\tilde{\mathcal{R}}_q^v \leftarrow \emptyset$; $k \leftarrow 1$
2: while $|\tilde{\mathcal{R}}_q^v| < \tau_v$ do
3:     for all $d_i \in \mathcal{R}_q^v$ do
4:         $sim(d_i) \leftarrow \phi\left(d_i, \mathcal{R}_q^w(k)\right)$
5:     end for
6:     $d^* \leftarrow \arg\max_{d_i} sim(d_i)$
7:     $\tilde{\mathcal{R}}_q^v \leftarrow \tilde{\mathcal{R}}_q^v \cup \{d^*\}$; $\mathcal{R}_q^v \leftarrow \mathcal{R}_q^v \backslash \{d^*\}$; $k \leftarrow k + 1$
8: end while
9: return $\tilde{\mathcal{R}}_q^v$
```

A possible disadvantage of WebSim is that the top τ_v web results may not cover all the query-senses in the top 10 web results. For example, the top 10 web results may include results about “saturn” the planet and the car, but the top τ_v web results may all be about the planet. The next two approaches attempt to address this issue.

Web Similarity MMR. WebSimMMR (Algorithm 2) is almost identical to WebSim. However, instead of selecting the vertical results most similar to the top τ_v results in $\mathcal{R}_q^w$, it first uses MMR to re-rank $\mathcal{R}_q^w$ into $\tilde{\mathcal{R}}_q^w$ (line 1). Then, it iteratively appends vertical results to $\tilde{\mathcal{R}}_q^v$ such that $\tilde{\mathcal{R}}_q^v(k)$ corresponds to the vertical result in $\mathcal{R}_q^v$ most similar to $\tilde{\mathcal{R}}_q^w(k)$ (lines 3-6). The goal of internally re-ranking the web results using MMR is to have the top τ_v results in $\tilde{\mathcal{R}}_q^w$ represent different query-senses present in the top 10 web results.

Algorithm 2 Web Similarity (MMR)

```
   WebSimMMR($\mathcal{R}_q^v, \mathcal{R}_q^w, \tau_v$)
1: $\tilde{\mathcal{R}}_q^v \leftarrow \emptyset$; $k \leftarrow 1$; $\tilde{\mathcal{R}}_q^w \leftarrow$ MMR($\mathcal{R}_q^w$) ▷ Re-rank top 10 web results ($\mathcal{R}_q^w$) with MMR.
2: while $|\tilde{\mathcal{R}}_q^v| < \tau_v$ do
3:     for all $d_i \in \mathcal{R}_q^v$ do
4:         $sim(d_i) \leftarrow \phi\left(d_i, \tilde{\mathcal{R}}_q^w(k)\right)$
5:     end for
6:     $d^* \leftarrow \arg\max_{d_i} sim(d_i)$
7:     $\tilde{\mathcal{R}}_q^v \leftarrow \tilde{\mathcal{R}}_q^v \cup \{d^*\}$; $\mathcal{R}_q^v \leftarrow \mathcal{R}_q^v \backslash \{d^*\}$; $k \leftarrow k + 1$
8: end while
9: return $\tilde{\mathcal{R}}_q^v$
```

A potential disadvantage of WebSimMMR is that the ordering of vertical results in $\tilde{\mathcal{R}}_q^v$ is somewhat arbitrary. Our final approach attempts to order the vertical results based on the proportion of top 10 web results on that query-sense.

Web Cluster Similarity. WebClusterSim (Algorithm 3) first clusters the top 10 web results into τ_v clusters (line 1). We used complete-link agglomerative clustering. The resulting clusters ($\mathcal{C}_q^w$) are ordered by size such that $\mathcal{C}_q^w(k)$ corresponds to the kth largest cluster. Then, WebClusterSim iteratively appends vertical results to $\tilde{\mathcal{R}}_q^v$ such that $\tilde{\mathcal{R}}_q^v(k)$ corresponds to the vertical result in $\mathcal{R}_q^v$ with the greatest average similarity with the web results assigned to cluster $\mathcal{C}_q^w(k)$ (lines 3-6). The goal of WebClusterSim is to have vertical result $\tilde{\mathcal{R}}_q^v(k)$ be about the kth most frequent query-sense in the top 10 web results.

Algorithm 3 Web Cluster Similarity

```
   WebClusterSim($\mathcal{R}_q^v, \mathcal{R}_q^w, \tau_v$)
1: $\tilde{\mathcal{R}}_q^v \leftarrow \emptyset$; $k \leftarrow 1$; $\mathcal{C}_q^w \leftarrow Cluster(\mathcal{R}_q^w)$ ▷ Cluster top 10 web results into $t_v$ clusters.
2: while $|\tilde{\mathcal{R}}_q^v| < \tau_v$ do
3:     for all $d_i \in \mathcal{R}_q^v$ do
4:         $sim(d_i) \leftarrow \phi_{avg}\left(d_i, \mathcal{C}_q^w(k)\right)$          ▷ Compute average similarity.
5:     end for
6:     $d^* \leftarrow \arg\max_{d_i} sim(d_i)$
7:     $\tilde{\mathcal{R}}_q^v \leftarrow \tilde{\mathcal{R}}_q^v \cup \{d^*\}$; $\mathcal{R}_q^v \leftarrow \mathcal{R}_q^v \backslash \{d^*\}$; $k \leftarrow k + 1$
8: end while
9: return $\tilde{\mathcal{R}}_q^v$
```

Implementation Details. All of the above algorithms required measuring the similarity between pairs of web and/or vertical documents (denoted as function ϕ in Algorithms 1-3). To this end, we represented documents using their topical distribution.[1] First, we identified 128 second-level categories from the Open Directory Project (ODP) hierarchy and crawled 2,000 random webpages from each category.[2] Then, we trained 128 logistic regression classifiers using the Liblinear Toolkit.[3] We adopted a simple TF.IDF representation with stemming and stopwords removed, and normalized documents to unit length. Finally, we used the mass-normalized prediction confidence values from each classifier to generate a topical distribution for a each web and vertical document. Document similarity was measured using the symmetrized Kullback-Leibler divergence (KLD) [12].[4]

4 User Study

Experimental Protocol. Our goal was to study search behavior under the following scenario: First, a user has a particular search task in mind (e.g., "Find scientific information about Saturn the planet.") and enters an ambiguous query (e.g., "saturn"). Then, in response to this query, the system decides to integrate results from a particular vertical (e.g., images) into the web results. While the vertical results may be relevant to a different user, this particular user's information need is better satisfied by web results. Finally, based on the vertical and web results presented, the user must decide whether to engage with the web results or reformulate the query. We evaluate algorithms for deciding which vertical results to display. The goal is to influence the user to make a productive decision with respect to the web results—to engage with the web results if at least one of them is relevant and to *avoid* engaging otherwise.

The experimental protocol is shown in Figure 1. Participants were given a search task and were asked to use a live search engine to find a webpage containing the requested information. Search tasks had the form "Find information about `<entity>`", for example, "Find tourism information about Washington State." In order to do a controlled study of the scenario described above, participants were told that "to help get you started with the search task, you will be provided with an initial query and a set of results." This starting point SERP, called the *initial SERP*, is where the experimental manipulation took place.

The initial SERP included a search task description, an initial query, and a set of results, supposedly returned in response to the initial query. As described in detail below, the initial query was purposely ambiguous (e.g., "washington", which could mean the city, state, or historical figure) and the search results included web results and blended results from one of four verticals (images, news, shopping, or video). The web results corresponded to the top 10 results returned by the Bing Web Search API (in their original order) and the vertical results were determined by one of the algorithms described in Section 3. The vertical results were always blended between the third and fourth web result.

[1] All results had a textual representation. The web and news results had a title and summary snippet, while the image, shipping, and video results had a title.

[2] `http://www.dmoz.org/`

[3] `http://www.csie.ntu.edu.tw/~cjlin/liblinear/`

[4] KLD measures distance (i.e., smaller values indicate greater similarity). Thus, all of the above algoritms used the negative KLD to measure similarity.

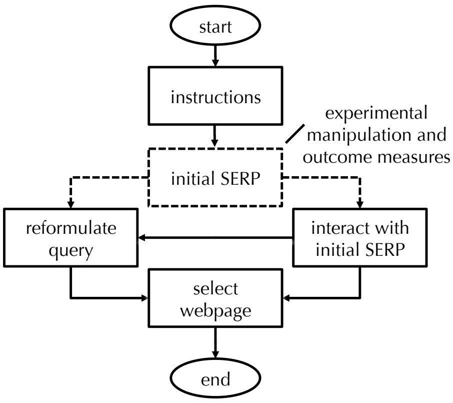
start
instructions
experimental manipulation and outcome measures
initial SERP
reformulate query
interact with initial SERP
select webpage
end

Fig. 1: Experimental protocol.

From the initial SERP, participants were asked to search naturally by examining the results provided or entering their own queries. Participant queries returned results using the Bing Web Search API without vertical results. Clicking on a result opened the landing page inside an HTML frame, with a button above the frame labeled: "Click here if the page contains the requested information." Clicking this button ended the search task. The goal of the study was disguised by telling participants that we were testing a new search engine.

Verticals. We experimented with four verticals: images, news, shopping, and video. Results for the images, news, and video verticals were obtained using Bing APIs and results for the shopping vertical were obtained using the eBay API. Vertical results were presented similarly to how they are presented in commercial systems. For the image, shopping, and video verticals, we blended five results horizontally on the SERP ($\tau_v = 5$), and for the news vertical, we blended three results vertically ($\tau_v = 3$). Image results were presented using thumbnails; news results were presented using the article title, summary, news source, and publication age; shopping results were presented using the product title, price, condition, and a thumbnail of the product; and video results were presented using the title, duration, and a keyframe of the video.

Search Tasks. Each vertical was associated with its own set of search tasks. For the purpose of our study, we extended the set of search tasks used in Arguello and Capra [4]. Next, we describe how the original search tasks were created and how we added new tasks.

Each search task was associated with two components: the search task description and the initial query. The search task description was a simple request for information and the initial query was purposely ambiguous. Arguello and Capra [4] created 300 search tasks (75 per vertical) using the following process. The first step was to gather a large set of ambiguous queries. To this end, the authors identified all entities associated with a Wikipedia disambiguation page that also appear as a query in the AOL query-log. The next step was to identify queries with a strong orientation towards one of the four verticals considered. To accomplish this, each candidate initial query was issued to Bing and four (possibly overlapping) sets of queries were gathered based on whether the query triggered the image, news, shopping, and/or video vertical in the Bing results. Finally, the authors identified 75 queries per vertical that returned multiple senses from its corresponding vertical search API. For each query, the search task was constructed about one of the senses in the vertical results.

To conduct a more robust evaluation, we aimed to double the number of search tasks. For each initial query, we tried to create a new search task based on a different query-sense in the vertical results. We were unable to construct a new search task for 29 initial queries because the other query-senses in the vertical results were too obscure. We ended up with a total of 571 search tasks. In order to study the spill-over effect from the vertical to the web results, search tasks were designed to require web results instead of vertical results. See Arguello and Capra [4] (Table 1) for a few example tasks from the original set.

User Study Implementation. The study was run as a remote study using Amazon's Mechanical Turk (MTurk). Each MTurk Human Intelligence Task (HIT) was associated with a single search task. We evaluated a total of five algorithms: the four algorithms described in Section 3 and, as a baseline for comparison, an approach that simply presented the top τ_v results returned by the corresponding vertical API for the initial query. Additionally, we collected data by showing participants only the web results (without any vertical results). In total, this resulted in 3,426 experimental conditions (571 search tasks × (5 algorithms + 1 no vertical) = 3,426). Finally, we collected data from 6 redundant participants for each experimental condition, for a total of 3,426 × 6 = 20,556 trials or HITs. Each HIT was priced at $0.10 USD.

Our HITs were implemented as *external* HITs, meaning that everything besides recruitment and compensation was managed by our own server. Hosting our HITs externally allowed us to control the assignment of MTurk workers to experimental conditions. Workers were assigned to experimental conditions randomly, except for two constraints. First, participants were not allowed to complete search tasks for the same initial query. Second, in order to obtain interaction data from a large number of participants, workers were not allowed to complete more than 60 HITs. We collected data from 1,135 participants.

MTurk studies require quality control and we addressed this in three ways. First, we restricted our HITs to workers with a 95% acceptance rate or greater. Second, to help ensure English language proficiency, we limited our HITs to workers in the US. Finally, using an external HIT design allowed us to do quality control dynamically. Prior to the experiment, we conducted a preliminary study to judge the relevance of each web result on an initial SERP. During the experiment, participants who selected three non-relevant web results from an initial SERP as being relevant were not allowed to do more HITs.

Evaluation Methodology. We evaluate algorithms for deciding which results from a particular vertical to display. Algorithms were evaluated based on their ability to influence our study participants to make productive decisions with respect to the web results on the initial SERP. If we view user engagement with the web results as a binary decision, there are two ways users can make a productive decision: (1) they can engage with the web results if at least one of them is relevant or (2) they can *avoid* engaging with the web results otherwise. These correspond to true-positive and true-negative decisions, respectively.

To facilitate our analysis, it was first necessary to determine the relevance of each web result on an initial SERP. We collected relevant judgements using MTurk. We collected 10 redundant judgements per web-result/search-task pair for a total of 57,100 judgements (571 search tasks × 10 web results per task × 10 redundant judgements). The Fleiss' Kappa agreement was $\kappa_f = 0.595$, which is approaching *substantial* agreement (i.e., $\kappa_f = 0.600$) [13]. We aggregated relevance judgements using a majority vote—a web result was considered relevant if more than five MTurk workers marked it as relevant.

Engagement with the web results on an initial SERP was operationalized using clicks. We say that a participant engaged with the web results if he/she clicked on *at least one* and did not engage with the web results otherwise. Algorithms were evaluated using three metrics: (1) *accuracy* measures the percentage of true-positive and true-negative decisions (i.e., the participant clicked on a web result on the initial SERP and at least one of them was relevant *or* did not click on any and none of them were relevant), (2) the *true positive rate* measures the percentage of times there was a relevant web result on the initial SERP and the participant clicked on at least one, and (3) the *true negative rate* measures the percentage of times there were no relevant web results on the initial SERP and the participant did not click on any. Each experimental condition (i.e., search-task/algorithm pair) was completed by 6 redundant participants. We report performance by macro-averaging across search tasks and computed statistical significance using an approximation of Fisher's randomization test [20].

5 Results and Discussion

Results are presented in Tables 1-3 in terms of accuracy, true positive rate (TPR), and true negative rate (TNR). We were interested in measuring performance overall and for each vertical independently. Thus, we present macro-averaged performance across all search tasks (i.e., combining those from every vertical) and separately for those search tasks specific to each vertical. NoVertical gives the performance obtained from showing participants only the web results (without any vertical results) and Algo gives to the performance obtained from showing participants the top t_v results from the corresponding vertical search API. The Algo approach represents an aggregated search system that does not perform *vertical results selection*. The percentages indicate the percent change compared to NoVertical. The symbols $^\vartriangle(^\triangledown)$ denote a statistically significant increase(decrease) in performance compared to NoVertical and the symbols $^\blacktriangle(^\blacktriangledown)$ denote a statistically significant increase(decrease) in performance compared to Algo. The gray cells indicate the best performing algorithm within each column. Next, we discuss the differences in performance between algorithms, verticals, and evaluation metrics.

Algorithms. In terms of accuracy and TPR, ClusterWebSim was the best-performing algorithm. ClusterWebSim outperformed NoVertical for images, shopping and video, and performed only slightly worse for news (not significant). Moreover, ClusterWebSim outperformed Algo for shopping and video, performed at the same level for images, and only slightly worse for news (not significant).[5]

In terms of TNR, there was no clear winner—different algorithms performed better for different verticals. That said, ClusterWebSim was statistically indistinguishable from NoVertical and Algo for all verticals. It should also be noted that the differences between algorithms were less pronounced for TNR than for the other two metrics. We return to this point below.

It is also worth noting that ClusterWebSim outperformed MMR across all verticals and metrics. These two algorithms represent two different types of approaches to vertical results selection. ClusterWebSim selects results that are similar to the web results on the SERP and MMR selects results independently from the web results. Our results suggest that selecting vertical results that are similar to the web results can influence users to make more productive decisions with respect to the web results.

Verticals. In terms of accuracy and TPR (the metrics with the greatest variance), performance varied widely across verticals. The vertical results had a stronger effect for images and shopping than for news and video. For example, in terms of accuracy, the greatest improvement over NoVertical was greater for images (11.29%) and shopping (6.57%) than for news (2.74%) and video (3.78%). A similar trend was observed in terms of TPR. This trend is consistent with the results from Arguello and Capra [4]. Arguello and Capra found that users are more likely to interact with the web results when the vertical results are more consistent with the intended query-sense. However, the spill-over effect was only significant for images and shopping and not for news and video. Results from one of their studies suggests that images and shopping had more spill-over because their results are more salient and require less cognitive effort to process.

Table 1: Accuracy

	All Verticals	Images	News	Shopping	Video
NoVertical	0.573	0.549	0.583	0.578	0.582
Algo	0.587 (2.44%)	0.610 (11.11%)$^{\vartriangle}$	0.592 (1.54%)	0.569 (-1.56%)	0.577 (-0.86%)
MMR	0.580 (1.22%)	0.576 (4.92%)	0.575 (-1.37%)	0.578 (0.00%)	0.592 (1.72%)
WebSim	0.592 (3.32%)$^{\vartriangle}$	0.601 (9.47%)$^{\vartriangle}$	0.589 (1.03%)	0.588 (1.73%)	0.590 (1.37%)
WebSimMMR	0.581 (1.40%)	0.566 (3.10%)$^{\blacktriangledown}$	0.599 (2.74%)	0.574 (-0.69%)	0.582 (0.00%)
ClusterWebSim	0.602 (5.06%)$^{\vartriangle}$	0.611 (11.29%)$^{\vartriangle}$	0.580 (-0.51%)	0.616 (6.57%)$^{\vartriangle\blacktriangle}$	0.604 (3.78%)

Table 2: True Positive Rate (TPR)

	All Verticals	Images	News	Shopping	Video
NoVertical	0.395	0.422	0.393	0.377	0.386
Algo	0.415 (5.06%)	0.500 (18.48%)$^{\vartriangle}$	0.398 (1.27%)	0.374 (-0.80%)	0.375 (-2.85%)
MMR	0.408 (3.29%)	0.461 (9.24%)	0.381 (-3.05%)	0.383 (1.59%)	0.403 (4.40%)
WebSim	0.420 (6.33%)$^{\vartriangle}$	0.481 (13.98%)$^{\vartriangle}$	0.401 (2.04%)	0.405 (7.43%)	0.383 (-0.78%)
WebSimMMR	0.410 (3.80%)	0.458 (8.53%)$^{\blacktriangledown}$	0.415 (5.60%)	0.380 (0.80%)	0.379 (-1.81%)
ClusterWebSim	0.435 (10.13%)$^{\vartriangle}$	0.502 (18.96%)$^{\vartriangle}$	0.387 (-1.53%)	0.429 (13.79%)$^{\vartriangle\blacktriangle}$	0.415 (7.51%)

[5] For the video vertical, the improvement of ClusterWebSim over Algo was marginally significant in terms of accuracy ($p = 0.059$) and TPR ($p = 0.060$).

Table 3: True Negative Rate (TNR)

	All Verticals	Images	News	Shopping	Video
NoVertical	0.950	0.960	0.959	0.951	0.935
Algo	0.952 (0.21%)	0.965 (0.52%)	0.977 (1.88%)	0.932 (-2.00%)	0.941 (0.64%)
MMR	0.945 (-0.53%)	0.944 (-1.67%)	0.960 (0.10%)	0.941 (-1.05%)	0.935 (0.00%)
WebSim	0.957 (0.74%)	0.985 (2.60%)$^{\vartriangle}$	0.960 (0.10%)	0.929 (-2.31%)	0.964 (3.10%)$^{\vartriangle}$
WebSimMMR	0.942 (-0.84%)	0.914 (-4.79%)$^{\triangledown\blacktriangledown}$	0.963 (0.42%)$^{\blacktriangle}$	0.934 (-1.79%)	0.947 (1.28%)
ClusterWebSim	0.957 (0.74%)	0.960 (0.00%)	0.963 (0.42%)	0.963 (1.26%)	0.944 (0.96%)

Metrics. Performance across algorithms varied widely in terms of TPR, but was fairly stable in terms of TNR. There are two possible explanations for this. First, it may be that the vertical results had a stronger effect in causing participants to engage with the web results than in causing participants to *avoid* engaging with the web results. In other words, seeing the relevant query-sense in the vertical results may have a strong positive effect on users, but *not* seeing the relevant query-sense may have only a weak *negative* effect. Alternatively, the stability in TNR performance might be explained by our use of *clicks* as a proxy for user engagement with the web results. It may be that participants were often misled by incoherent vertical results, but were still effective at *not* clicking on a non-relevant web result based on its surrogate. Future work might consider a less conservative proxy for user engagement, for example, derived from browsing behavior (e.g., Did the participant scroll down the initial SERP?). A less conservative proxy might reveal greater differences in terms of TNR.

6 Conclusion

We developed and evaluated algorithms for vertical results selection—deciding which results from a particular vertical to display. Algorithms were evaluated based on their ability to influence users to make productive decisions with respect to the web results on the SERP. Results from our user study suggest the following trends. First, our best-performing algorithm (ClusterWebSim) selects vertical results that are similar to the web results. This algorithm performed better than simply presenting the top vertical results (Algo) and diversifying the vertical results independently from the web results (MMR). We treat this as evidence that improving the level of coherence between the vertical and web results can influence users to make more productive decisions with respect to the web results. Second, the vertical results had as stronger effect for some verticals (images, shopping) than others (news, video). This is consistent with prior work and may be due to the vertical surrogate representation. Finally, we observed that the vertical results had a greater effect on users discovering relevant web results on the SERP than on users avoiding non-relevant ones. We used clicks as a proxy for user engagement with the web results. It remains to be seen whether this trend holds true for a less conservative measurement of engagement.

Our findings have important implications for aggregated search. Current methods for vertical selection and presentation do not explicitly ensure coherence with other components on the SERP. We show that relatively simple algorithms for vertical results selection can help avoid negative cross-component effects. In this work, we focused on search tasks that favored web results and performed vertical results selection to ensure coherence with the web results. Future work will develop a unified framework that performs *results selection* to ensure coherence with the most confidently relevant component(s).

References

1. R. Agrawal, S. Gollapudi, A. Halverson, and S. Ieong. Diversifying search results. In *WSDM*, pages 5–14, 2009.
2. J. Arguello and R. Capra. The effect of aggregated search coherence on search behavior. In *CIKM*, pages 1293–1302, 2012.
3. J. Arguello and R. Capra. The effects of vertical rank and border on aggregated search coherence and search behavior. In *CIKM*, pages 539–548, 2014.
4. J. Arguello, R. Capra, and W.-C. Wu. Factors affecting aggregated search coherence and search behavior. In *CIKM*, pages 1989–1998, 2013.
5. J. Arguello, F. Diaz, and J. Callan. Learning to aggregate vertical results into web search results. In *CIKM*, pages 201–210, 2011.
6. J. Arguello, F. Diaz, J. Callan, and J.-F. Crespo. Sources of evidence for vertical selection. In *SIGIR*, pages 315–322, 2009.
7. P. Bailey, N. Craswell, R. W. White, L. Chen, A. Satyanarayana, and S. M. Tahaghoghi. Evaluating search systems using result page context. In *IIiX*, pages 105–114, 2010.
8. J. Carbonell and J. Goldstein. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *SIGIR*, pages 335–336, 1998.
9. B. Carterette and P. Chandar. Probabilistic models of ranking novel documents for faceted topic retrieval. In *CIKM*, pages 1287–1296, 2009.
10. F. Diaz. Integration of news content into web results. In *WSDM*, pages 182–191, 2009.
11. F. Diaz and J. Arguello. Adaptation of offline vertical selection predictions in the presence of user feedback. In *SIGIR*, pages 323–330, 2009.
12. H. Jeffreys. An invariant form for the prior probability in estimation problems. *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences*, 186(1007):pp. 453–461, 1946.
13. J. R. Landis and G. G. Koch. The measurement of observer agreement for categorical data. *Biometrics*, 33(1):159–174, 1977.
14. A. K. Ponnuswami, K. Pattabiraman, D. Brand, and T. Kanungo. Model characterization curves for federated search using click-logs: predicting user engagement metrics for the span of feasible operating points. In *WWW*, pages 67–76, 2011.
15. A. K. Ponnuswami, K. Pattabiraman, Q. Wu, R. Gilad-Bachrach, and T. Kanungo. On composition of a federated web search result page: Using online users to provide pairwise preference for heterogeneous verticals. In *WSDM*, pages 715–724, 2011.
16. F. Radlinski and S. Dumais. Improving personalized web search using result diversification. In *SIGIR*, pages 691–692, 2006.
17. M. Sanderson. Ambiguous queries: test collections need more sense. In *SIGIR*, pages 499–506, 2008.
18. R. L. T. Santos, C. Macdonald, and I. Ounis. Exploiting query reformulations for Web search result diversification. In *WWW*, pages 881–890, 2010.
19. R. L. T. Santos, C. Macdonald, and I. Ounis. Aggregated search result diversification. In *ITCIR*, pages 250–261, 2011.
20. M. D. Smucker, J. Allan, and B. Carterette. A comparison of statistical significance tests for information retrieval evaluation. In *CIKM*, pages 623–632, 2007.
21. K. Zhou, R. Cummins, M. Lalmas, and J. M. Jose. Evaluating aggregated search pages. In *SIGIR*, pages 115–124, 2012.