ViTOR: Learning to Rank Webpages Based on Visual Features

Paper Written by Bram van den Akker et al, WWW 2019

CSCE 670 Paper Breakdown by **Ping Lu**,
Texas A&M

Motivation

- Opportunities: Use the **design and visual appearance** of a webpage to improve the performance of **learning to rank (LTR)**.
- Challenge 1: Relatively **little** is **known** about the the **relation** between visual appearance and user perception of a webpage.
- Challenge 2 : **Methods to extract** visual features is limited
- Challenge 3: **No appropriate dataset** available to support research on LTR with visual features.

Motivation

- The perceived relevance of a page is dictated by the **layout of the page and formatting: people tend to read in an "F-shape"** (on the right graph)
- Users like to scan **first lines** in a page
- Then **first few words** on the left of each line (shorter bar in F)
- Finally, users scan the content's very left side in a **vertical movement** (stem of F)
- Users like to scan **bold** words, **headings** and subheadings, and ***different formatting*** words.
- Advice: **Prioritize and format words** carrying the most relevant information in the **"F-shape"** and in different formats

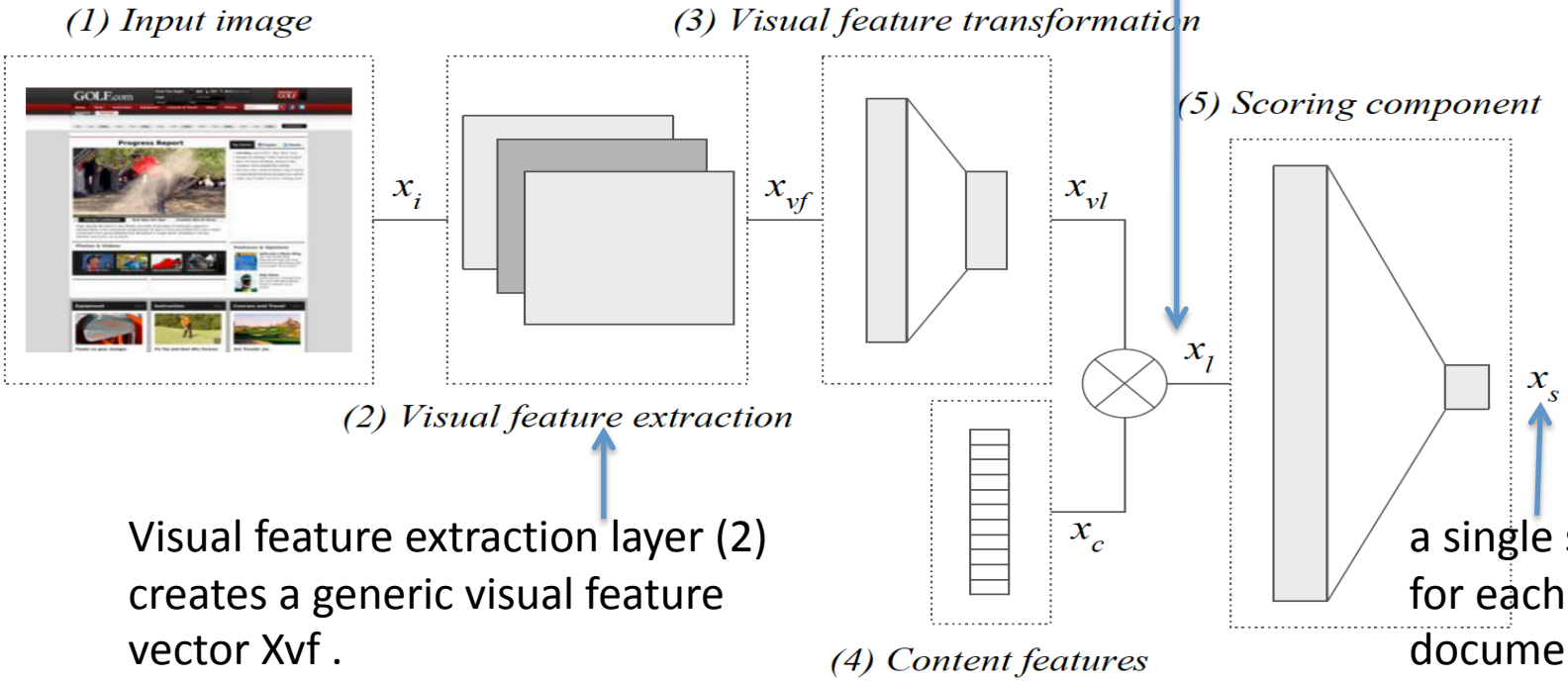
Figure 1: Heat map in an **F-shaped** pattern from an eye tracking research 2016, www.nngroup.com/articles/f-shaped-pattern-reading-web-content

Main Technical Contribution

- There was a study by Fan et al. that used visual information to rank pages by building a model called ViP.
- In our paper here, the authors proposed the Visual learning TO Rank (**ViTOR**) model that integrates the state-of-the-art ***visual features extraction methods***, and it has shown to significantly improved LTR performance and outperformed ViP.
- First, the authors extract visual features from webpage snapshots using **transfer learning** and, in particular, by adopting **the VGG-16 and ResNet-152 models pre-trained on ImageNet**.
- Second, they introduce **a novel set of visual features extracted from synthetic saliency heatmaps**, which explicitly model how users view webpages.

Main Technical Contribution-Model Architecture

The visual feature vector Xvl are combined with other content features Xc and form the final feature vector Xl.



Visual feature extraction layer (2) creates a generic visual feature vector Xvf .

a single score Xs for each query-document pair.

Figure 1. Model Architecture

The resulting model is trained end-to-end using **a pair-wise hinge loss with L2 regularization**

Main Technical Contribution-Visual Feature Extractors

- The authors use the **VGG-16** [Karen Simonyan 2014] and **ResNet-152** [Kaiming He 2016] Models as the visual feature extraction models which both have **pre-trained parameters**.
- VGG-16 and ResNet-152 have **convolutional layers** that **extract features** from an input image, which are in turn used by a **fully connected layer** to **classify each image**.
- We use these **convolutional layers** as the **visual feature extraction layer**, which transforms Xi to Xvf . All parameters of these **convolutional layers** are **frozen** during training.
- The **fully connected layers** can be altered and retrained in order to be used with new inputs and tasks. We utilize them as a **visual feature transformation layer** within the ViTOR architecture to produce LTR specific features Xvl.
- VGG-16 has 1 fully connected layer. ResNet-152 transformation layer has 3 layers.

Main Technical Contribution- Saliency Heatmaps Advantage

- The authors propose to explicitly **model the user viewing pattern** through synthetic saliency heatmaps.
- First, the synthetic saliency heatmaps explicitly learn to predict how users perceive webpages by **training end-to-end model on actual eye-tracking data**. We expect this information to better correlate with webpage relevance compared to raw snapshots.
- Second, saliency heatmaps **reduce the average storage** requirements by up to 90%, because they are gray-scale images.
- The authors use a two-stage transfer learning model that **convert a raw snapshot into a synthetic saliency heatmap**. This heatmap is then used as an **input imag**e Xi for the ViTOR model.

Interesting Experimental Results- Three Types of Input and Output Baseline

- ViTOR highlights use visual features extracted from snapshots of webpages **with highlighted query terms** (2nd column on the right).
- ViTOR saliency uses visual features extracted from **synthetic saliency heatmaps.**
- The **vanilla snapshots, highlights and saliency heatmaps** for each model respectively are used as the input image.
- The authors compare the proposed ViTOR model to the ViP model by Fan et al., and a number of content-based ranking methods.

Figure 2: Examples of a vanilla snapshot, a red highlighted snapshot, and a saliency heatmap from left to right, respectively.

Interesting Experimental Results

Table 1 Results for the ViTOR model using only content features (baseline), vanilla snapshots, highlighted snapshots, and saliency heatmaps. All results significantly improve over the ViTOR baseline. Best results are shown in bold.

	p@1	p@10	ndcg@1	ndcg@10	MAP
ViTOR baseline	0.338	0.370	0.189	0.233	0.415
VGG snapshots	0.514	0.484	0.292	0.324	0.442
ResNet snapshots	0.550	0.452	0.310	0.301	0.437
VGG highlights	**0.560**	**0.520**	0.323	**0.346**	**0.456**
ResNet highlights	0.530	0.463	0.305	0.312	0.440
VGG saliency	0.554	0.453	0.310	0.302	0.422
ResNet saliency	**0.560**	0.476	**0.333**	0.321	0.442

The highlighted snapshots carry more information compared to vanilla snapshots.

The saliency heat maps with ResNet-152 match and outperform VGG-16 with highlighted snapshots when looking at ndcg@1.

Interesting Experimental Results

Table 2 Results for the VGG-16 with highlighted snapshots, ResNet-152 with saliency heatmaps, and baselines. † indicates a significant decrease in performance compared to VGG highlights and ‡ indicates a significant decrease in performance compared to both ViTOR implementations. Best results are shown in bold.

	p@1	p@10	ndcg@1	ndcg@10	MAP
BM25	0.300‡	0.316‡	0.153‡	0.188‡	0.350‡
RankBoost	0.450	0.444	0.258	0.288†	0.427
AdaRank	0.290‡	0.357‡	0.149‡	0.227‡	0.398
LambdaMart	0.470	0.420†	0.256	0.275†	0.418
ViP snapshots	0.392‡	0.398‡	0.217‡	0.254‡	0.421‡
ViP highlights	0.418‡	0.416‡	0.239‡	0.269‡	0.422‡
VGG highlights	**0.560**	**0.520**	0.323	**0.346**	**0.456**
ResNet saliency	**0.560**	0.476	**0.333**	0.321	0.442

The proposed ViTOR model **outperforms** baselines, whether they are supervised or unsupervised, use visual features or not.

My Thoughts-The Good

- ***The genius 1:*** The architecture of this model **separate the visual feature extraction layer and visual feature transformation layer**, this step made the computation way faster and cost-efficient in memory. Since the model is pretrained and the parameters are all frozen during the extraction step, the extraction step results are query independent for the snapshots input and saliency heatmaps input and can be stored on disk. The real time query dependent LTR happens in the feature transformation layer and is **fast to compute**.
- ***The genius 2***: Utilized a **model trained on actual eye-tracking data** to **synthesize saliency heatmaps**, which **explicitly model how users view webpages**. The authors then use the resulting heatmaps as an input Xi. The input is query independent and relate to what information the users actually get from the webpage better than vanilla snapshots.

My Thoughts-Suggestions

- However, to achieve consistent significant improvements compared to the state-of-the-art LTR methods, **different loss functions** within the ViTOR model have to be investigated.
- Other state-of-the-art **feature extraction methods** can be implemented for further exploration, such as the CapsuleNet [Sara Sabour 2017] model.
- Another promising direction is to **combine multiple visual features**, i.e., visual features extracted from vanilla snapshots, snapshots with highlights and saliency heatmaps.
- Other methods of **combining visual and textual features** might also be worth exploring.