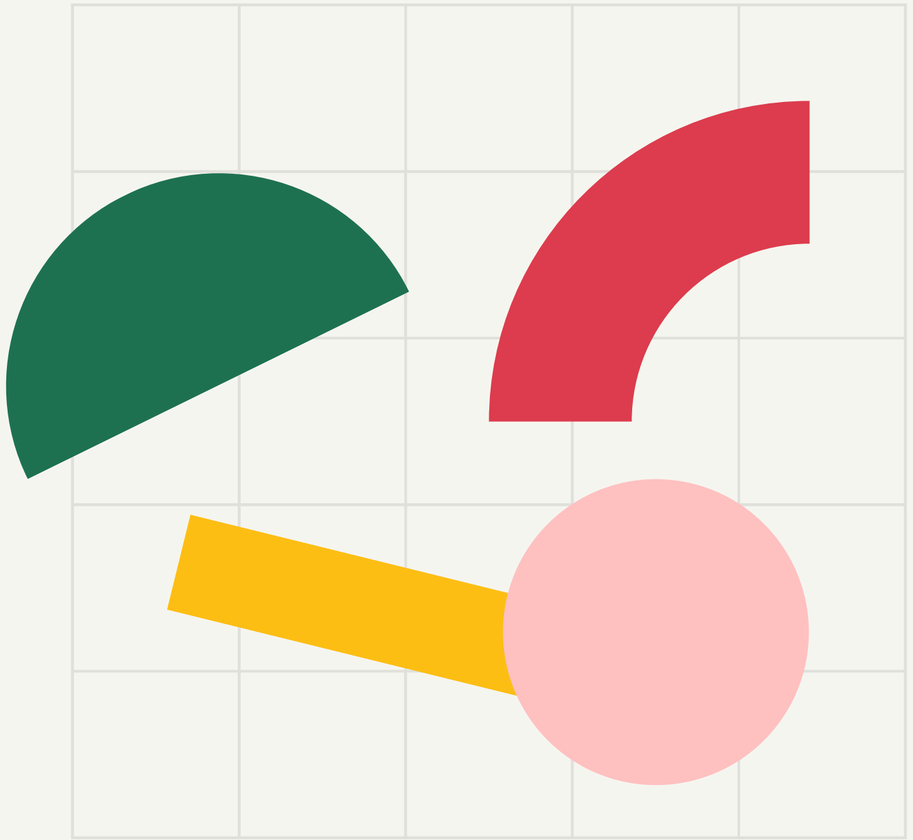

# HTML Essentials

Course for Beginners

# Unit Goals

what we'll cover

- HTML basics
- elements vs tags
- common HTML elements
- intro to MDN
- HTML boilerplate

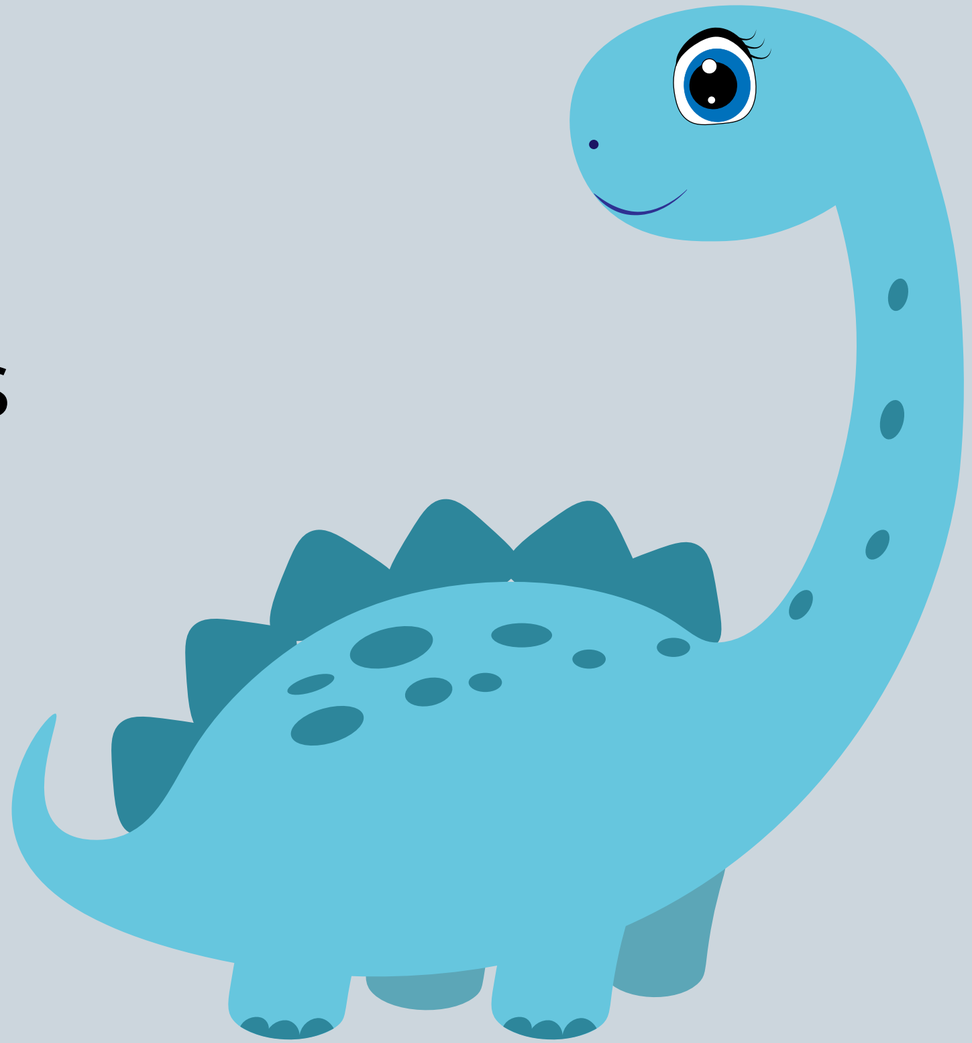


**THE**

**BLUE** ---------- CSS - adjectives

**DINO** ---------- HTML - nouns

**SMILED** ---- JS - verbs

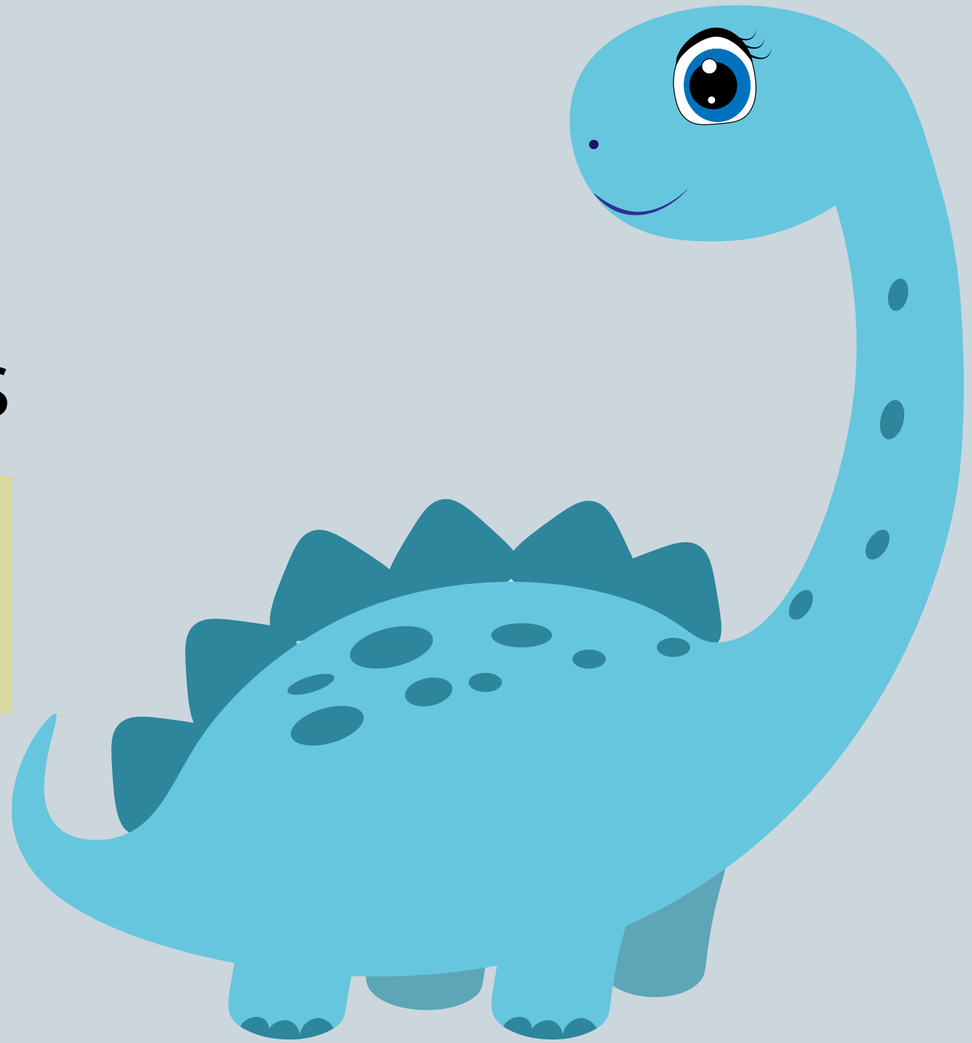


**THE**

**BLUE** ---------- CSS - adjectives

**DINO** ---------- HTML - nouns

**SMILED** ---- JS - verbs



# HTML

IS A MARKUP LANGUAGE

# MARKUP LANGUAGE

How would you describe this paper's structure to someone over the phone so that they could reproduce it?

## Learning to Cluster Web Search Results

Hua-Jun Zeng[1] Qi-Cai He[2] Zheng Chen[1] Wei-Ying Ma[1] Jinwen Ma[2]

[1]Microsoft Research, Asia
49 Zhichun Road
Beijing 100080, P.R.China

{hjzeng, zhengc, wyma}@microsoft.com

[2]LMAM, Department of Information Science,
School of Mathematical Sciences, Peking University,
Beijing 100871, P. R. China

heqicai@pku.edu.cn,
jwma@math.pku.edu.cn

**ABSTRACT**

Organizing Web search results into clusters facilitates users' quick browsing through search results. Traditional clustering techniques are inadequate since they don't generate clusters with highly readable names. In this paper, we reformalize the clustering problem as a salient phrase ranking problem. Given a query and the ranked list of documents (typically a list of titles and snippets) returned by a certain Web search engine, our method first extracts and ranks salient phrases as candidate cluster names, based on a regression model learned from human labeled training data. The documents are assigned to relevant salient phrases to form candidate clusters, and the final clusters are generated by merging these candidate clusters. Experimental results verify our method's feasibility and effectiveness.

**Categories and Subject Descriptors**

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval - *Search process*, *Clustering*, *Selection process*; G.3 [**Probability and Statistics**]: Correlation and Regression Analysis

**General Terms**

Algorithms, Experimentation

**Keywords**

Search result organization, document clustering, regression analysis

### 1. INTRODUCTION

Existing search engines such as Google [4], Yahoo [15] and MSN [12] often return a long list of search results, ranked by their relevancies to the given query. Web users have to go through the list and examine the titles and (short) snippets sequentially to identify their required results. This is a time consuming task when multiple sub-topics of the given query are mixed together. For example, when a user submits query "jaguar" into Google and wants to get search results related to "big cats", s/he should go to the 10th, 11th, 32nd and 71st results.

A possible solution to this problem is to (online) cluster search results into different groups, and to enable users to identify their required group at a glance. Hearst and Pedersen [6] showed that relevant documents tend to be more similar to each other, thus the clustering of similar search results helps users find relevant results. In the above example for query "jaguar", if there is a group named "big cats", the four relevant results will be ranked high in the corresponding list (as shown in Figure 1). Several previous works [16][17][6][11][10] are conducted to develop effective and efficient clustering technology for search result organization. In addition, Vivisimo [14] is a real demonstration of this technique.


**Figure 1. An Example of Search Result Clustering**

Clustering methods don't require pre-defined categories as in classification methods. Thus, they are more adaptive for various queries. Nevertheless, clustering methods are more challenging than classification methods because they are conducted in a fully unsupervised way. Moreover, most traditional clustering algorithms cannot be directly used for search result clustering, because of some practical issues. Zamir and Etzioni [16][17] gave a good analysis on these issues. For example, the algorithm should take the document snippets instead of the whole documents as input, since the downloading of original documents is time-consuming; the clustering algorithm should be fast enough for online calculation; and the generated clusters should have readable descriptions for quick browsing by users, etc. We also follow these requirements to design our algorithm.

In this paper, we reformalize the search result clustering problem as a salient phrases ranking problem. Thus we convert an unsupervised clustering problem to a supervised learning problem. Although a supervised learning method requires additional training data, it enhances the performance of search result grouping significantly, and enables us to evaluate it accurately. Given a query and the ranked list of search results, our method first parses the whole list of titles and snippets, extracts all possible phrases (n-grams) from the contents, and calculates several properties for each phrase such as phrase frequencies,


*SIGIR'04*, July 25–29, 2004, Sheffield, South Yorkshire, UK.

# MARKUP LANGUAGE

"Make this part bold"
"Make this part a link"
"Make this a paragraph"

## Learning to Cluster Web Search Results

Hua-Jun Zeng[1] Qi-Cai He[2] Zheng Chen[1] Wei-Ying Ma[1] Jinwen Ma[2]

[1]Microsoft Research, Asia
49 Zhichun Road
Beijing 100080, P.R.China

{hjzeng, zhengc, wyma}@microsoft.com

[2]LMAM, Department of Information Science,
School of Mathematical Sciences, Peking University,
Beijing 100871, P. R. China

heqicai@pku.edu.cn,
jwma@math.pku.edu.cn

**ABSTRACT**

Organizing Web search results into clusters facilitates users' quick browsing through search results. Traditional clustering techniques are inadequate since they don't generate clusters with highly readable names. In this paper, we reformalize the clustering problem as a salient phrase ranking problem. Given a query and the ranked list of documents (typically a list of titles and snippets) returned by a certain Web search engine, our method first extracts and ranks salient phrases as candidate cluster names, based on a regression model learned from human labeled training data. The documents are assigned to relevant salient phrases to form candidate clusters, and the final clusters are generated by merging these candidate clusters. Experimental results verify our method's feasibility and effectiveness.

**Categories and Subject Descriptors**

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval - *Search process*, *Clustering*, *Selection process*; G.3 [**Probability and Statistics**]: Correlation and Regression Analysis

**General Terms**

Algorithms, Experimentation

**Keywords**

Search result organization, document clustering, regression analysis

**1. INTRODUCTION**

Existing search engines such as Google [4], Yahoo [15] and MSN [12] often return a long list of search results, ranked by their relevancies to the given query. Web users have to go through the list and examine the titles and (short) snippets sequentially to identify their required results. This is a time consuming task when multiple sub-topics of the given query are mixed together. For example, when a user submits query "jaguar" into Google and wants to get search results related to "big cats", s/he should go to the 10th, 11th, 32nd and 71st results.

A possible solution to this problem is to (online) cluster search results into different groups, and to enable users to identify their required group at a glance. Hearst and Pedersen [6] showed that relevant documents tend to be more similar to each other, thus the clustering of similar search results helps users find relevant results. In the above example for query "jaguar", if there is a group named "big cats", the four relevant results will be ranked high in the corresponding list (as shown in Figure 1). Several previous works [16][17][6][11][10] are conducted to develop effective and efficient clustering technology for search result organization. In addition, Vivisimo [14] is a real demonstration of this technique.


**Figure 1. An Example of Search Result Clustering**

Clustering methods don't require pre-defined categories as in classification methods. Thus, they are more adaptive for various queries. Nevertheless, clustering methods are more challenging than classification methods because they are conducted in a fully unsupervised way. Moreover, most traditional clustering algorithms cannot be directly used for search result clustering, because of some practical issues. Zamir and Etzioni [16][17] gave a good analysis on these issues. For example, the algorithm should take the document snippets instead of the whole documents as input, since the downloading of original documents is time-consuming; the clustering algorithm should be fast enough for online calculation; and the generated clusters should have readable descriptions for quick browsing by users, etc. We also follow these requirements to design our algorithm.

In this paper, we reformalize the search result clustering problem as a salient phrases ranking problem. Thus we convert an unsupervised clustering problem to a supervised learning problem. Although a supervised learning method requires additional training data, it enhances the performance of search result grouping significantly, and enables us to evaluate it accurately. Given a query and the ranked list of search results, our method first parses the whole list of titles and snippets, extracts all possible phrases (n-grams) from the contents, and calculates several properties for each phrase such as phrase frequencies,


*SIGIR'04*, July 25–29, 2004, Sheffield, South Yorkshire, UK.

# HTML ELEMENTS

To write HTML, we pick from a set of standard elements that all browsers recognize

**Common elements include:**

- <p> element - represents a paragraph of text
- <h1> element - represents the main header on a page
- <img> element - embeds an image
- <form> element - represents a form

# HTML TAGS

We create elements by writing *tags*. Most (but not all) elements consit of an opening and closing tag.

```
<p> I am a paragraph </p>
```

# HTML TAGS

We create elements by writing *tags*. Most (but not all) elements consit of an opening and closing tag.

opening tag

```
<p> I am a paragraph </p>
```

closing tag


DEVELOPER NETWORK

# HTML SKELETON

We write our HTML in a standard "skeleton"

```
<!DOCTYPE html>
<html>
<head>
  <title>My First Page</title>
</head>
<body>
  <!-- Content Goes Here -->
</body>
</html>
```