Week 2 - Markup Languages & Scientific Literature

Hongzheng Chen

Nov 22, 2019

1 Markup Languages

According to Wikipedia, a markup language is a system for annotating a document in a way that is syntactically distinguishable from the text. That is, you use a *easy way* to generate *complex visual text*.

XML (eXtensible Markup Language) may be the most general markup language. You organize contents in a tree-like structure with each content in a pair of tags (`<tag>content</tag>`). For an introduction to XML, you can refer to Runoob's tutorial. Since XML is quite easy, only comprehending the core idea is enough.

Based on XML, Resource Description Framework (RDF) is further used to describe resources on the web. The basic introduction of RDF can be found here. For more about RDF, you can refer to the following paper in the future, which is also a research direction of graph computing.

- Siyuan Wang, Chang Lou, Rong Chen, and Haibo Chen, *Fast and Concurrent RDF Queries using RDMA-assisted GPU Graph Exploration*, Proceedings of 2018 USENIX Annual Technical Conference (ATC), 2018
- Jiaxin Shi, Youyang Yao, Rong Chen, Haibo Chen and Feifei Li, *Fast and Concurrent RDF Queries with RDMA-based Distributed Graph Exploration*, Proceedings of 2016 Usenix Symposium on Operating System Design and Implementation (OSDI), 2016

Also, RDF is tightly connected with semantic networks and knowledge graph. They will not be expanded here, and are left to be discovered by yourselves in the future research.

JavaScript Object Notation (JSON) and YAML (YAML Ain't Markup Language) are two data serialization languages. Since they are commonly used and very similar to markup languages, we also introduce them here. C++/Java/Python all have extensions for them enabling programmers to better manage and transfer data. For their difference, you can see this discussion from Stack Overflow. YAML is also used as configuration files and can be loaded externally. Thus, configuring the environment in a new machine may become easy, since only simple editing on the YAML file is needed. Decoupling program and data also prevents sensitive information from leaking.

HyperText Markup Language (HTML) is not our focus, but you should know it, since it is one of the three front-end kits (HTML, CSS, Javascript) and is fundamental when you

develop your webpage. Also, to obtain original data on Internet, you will write spider that basically parses HTML pages, grab important information based on tags, and do data collections. There are lots of tutorials about HTML on Chinese Internet, such as this and this. But you should notice that HTML is only the basic of webpage / app development, for nowadays front-end developers, they may not directly write HTML. They leverage front-end *frameworks* like Vue or React to make faster developments.

Markdown is a user-friendly markup language to generate simple but tidy HTML webpage. The widest usage of Markdown maybe the README page on Github[1], which is the first page of each project. Many online discussion forums like Zhihu may initially support Markdown. As daily usage, Markdown is also very convenient to mark down notes or organize your schedule. There are lots of tutorials about Markdown on Chinese and English Internet, like this and this, which can be easily obtained by search engine. Also there are lots of online editors like Dillinger and StackEdit and offline support like Markdown Preview Enhanced for VS Code. Due to its simplicity, you can quickly code a parser from Markdown to HTML, as CSP 201703-3 indicates.

TEX and LATEX are widely used in paper writing, and will be covered in the next few courses. Combining with Markdown, they becomes the first choice for many CS students to take notes.

For history and other information about markup languages, you can refer to the Wiki page.

2 Scientific Literature

2.1 Ranking

The recommended conference and journal list proposed by China Computer Federation (CCF) is the basic evaluation standard in China. Application for a M.S. or Ph.D. or graduation from university will all be judged by this standard. Top-tier conferences and journals should be paid the most attention.

Since the development of CS is so fast that traditional journal publication cannot meet increasing academic communication requirements nowadays, conferences held each year may be the best places for researchers to share their ideas. Thus, conferences may be more important than journals in CS, and lead the research in each minor area.

CS Rankings is one of the objective ranking systems that rank universities all over the world based on their publications on CS-related areas. It is updated every day, so you can easily follow up and find what universities or professors do so well recently.

[1] About writing a good README page, you can refer to this template.

2.2 Literature Resources

Commonly, you can just type a paper's title and Google it, then you can find all the information on this paper.

The Institute of Electrical and Electronics Engineers (IEEE, pronounced i triple e) is the world's largest professional society with engineers and professionals from different backgrounds with a lot of strong volunteers around the world. Association for Computing Machinery (ACM) is more focused towards computing and CS-related fields like Data Analytics, Programming, Data Mining, Web and Software. Both of them are the organizers or publishers of many conferences and journals. IEEE Xplore and ACM DL are the digital libraries storing papers published by IEEE / ACM.

arXiv (pronounced "archive" - the X represents the Greek letter chi χ) is a repository of electronic preprints (known as e-prints) approved for posting after moderation, but not full peer review. arXiv covers various fields including mathematics, physics, chemistry, biology, computer science, etc. Especially, artificial intelligence researchers prefer to post their manuscripts onto arXiv, while researchers other subfields in CS may be less interested in arXiv partly due to the rigorous review process of conferences in these subfields.

The major use case of arXiv is for disseminating manuscripts that you also publish in a journal or conference. By posting a preprint on arXiv, people can find your research, build on it, cite it, and give you feedback on it immediately, while at the same time the same work goes through the (sometimes slow) peer-review process. Some of these papers will fall out of the peer review pipeline at some point, and only appear on arXiv, but that doesn't necessarily mean that they are less useful, important, or sound[2].

Notice to access most of the `pdf` files of these papers, you should log in these databases using the university Internet.

To see all the researches done by some specific researchers, you can find them in dblp (a CS bibliography database), Google Scholar, LinkedIn (more common to people in enterprise), or their personal webpages.

2.3 Literature Structure

Scientific research papers may have good structures. Basically, they contain the following parts.

- **Title**: The highly compressed overview of what you do. The most important techniques or features of your work should be in the title.

[2]This paragraph comes from `https://academia.stackexchange.com/questions/75325/why-do-people-publish-on-arxiv-instead-of-other-places`

- **Author list**: Commonly[3], the authors are ordered by their contribution. The first one is called the first author who contributes most, including writing the manuscript and doing experiments. The corresponding author is the person who provides the idea and corrects the manuscript to meet the publication standard. And the last author is often the supervisor of this paper. More differences can be found in this article. Therefore, when you read a research paper, you should carefully pay attention to these authors and their affiliations. Moreover, try to dig out the relationship between the co-authors of different groups or research institutes, and you can find their research interests and their styles.
- **Abstract**: Briefly summarize why you make this work, what you have done, and the experimental results. 100 - 200 words are enough, but it often depends on what conferences or journals you submit.
- **Keywords**: As what it literally means, people can find your paper using these keywords.
- **Introduction**: Give a high overview of your field, the reason why you conduct this research, some previous researches and their problems, and the contribution of your work.
- **Background**: Provide basic knowledge for people to understand your paper, but you need NOT restate the well-known thing in your area. For example, if you propose a new generative adversarial network (GAN) structure in your work, you need not introduce the neural networks from perceptron, but you can directly skip to the previously GANs.
- **Motivation**: This is *extremely important*, since it relates why you do this research, if your research is an important problem most researchers focus nowadays, and whether your work can advance this field. It is one of the most important criteria when reviewers judge your paper. Thus, motivation should be stated clearly in the beginning of the paper. Researches in MICRO or ISCA may even open a "Motivation" section in the paper to detailedly present the experiments they done and thoroughly analyze the existing problems in previous researches.
- **Methodologies**: This is the core part of the research paper that provides how they conduct their work and how they solve the existing problems.
- **Experiments**: Firstly, they will present the experimental settings including the machine configuration, tools usage, hyperparameters, datasets, comparison baseline, etc. Please pay attention to settings since you may need to reproduce their work even when you conduct your own researches. Then, the paper will provide the experimen-

[3] In theoretical CS, they rank the authors in alphabetical order. There are many other fun ways to decide the author order, you can see this blog.

tal results using the proposed techniques. Pay attention to the data analysis part, and learn how to make thorough experiments related to one topic. Be careful of the hidden or unstated factors that may influence their results, which may be the flaw of their research.

- **Related work**: In this part, the authors list the work related to their researches, which can be served as further reading materials.
- **Conclusions & Discussions**: The authors conclude their work, state limitations, and provide guidance to future work.
- **Acknowledgments**: Commonly thank the anonymous reviewers for their great help, and list the funding number (refer to your supervisor).
- **References**: Different conferences or journals have different citation formats. Once your paper is accepted, they will tell you how to correctly format your paper. The easiest way is to keep a BibTEXfile, which will be discussed more in the next courses.

Different papers may have different organization structures, but they commonly cover the above topics. About how to read or write a good scientific paper, you can refer to Scientific Reading & Writing course lectured by Huawei Huang (2019 Fall in SYSU). The book for this course is Adrian Wallwork's *English for Writing Research Papers*.

By the way, when you write English, you can use Grammarly for spell checking, grammar checking, and proofreading.

2.4 Literature Management

When you start to read lots of papers, you cannot organize them by yourself using traditional files and folders. Thus a more efficient tool to manage the papers, notes, slides, and bibliographies is needed.

Zotero is for those usages. It is open-source with a big community and is very user-friendly. To import papers, you can drag papers to Zotero, and it will retrieve PDF metadata automatically. Or you can install a Zotero Connector in Chrome or other browsers to directly download papers to Zotero. You can explore yourself, and documents can be found here. A great introduction to Zotero can be found here.

3 Assignments

1. Add a README page for your assignment repository. Briefly describe what this repository is used for and the topic of each week.
2. Install Zotero and have a look at its functionality.
3. Generate ONE `.bib` file for ALL the following papers, which all comes from the top-tier conferences this year. You should first download the papers from those literature databases. The bibliographies should at least include a) the authors, b) the title of the paper, c) the *full name* of the conference, and d) the publication year.

ASPLOS'19 *Astra: Exploiting Predictability to Optimize Deep Learning* [AI/Sys]

MICRO'19 *Boosting the Performance of CNN Accelerators with Dynamic Fine-Grained Channel Gating* [AI/Arch]

ISCA'19 *TIE: Energy-efficient tensor train-based inference engine for deep neural network* [AI/Arch]

ICLR'19 *The Lottery Ticket Hypothesis: Finding Sparse, Trainable Neural Networks* (Best paper) [AI/Alg]

ATC'19 *NeuGraph: Parallel Deep Neural Network Computation on Large Graphs* [Graph/Sys]

SC'19 *Slim Graph: Practical Lossy Graph Compression for Approximate Graph Processing, Storage, and Analytics* [Graph/Sys]

PLDI'19 *Low-Latency Graph Streaming using Compressed Purely-Functional Trees* (Distinguished paper) [Graph/Alg]

KDD'19 *Network Density of States* (Best paper) [Graph/Alg]

4. Select ONE of the papers above, read its introduction / background / related work / conclusion section, and summarize

   - What are the problems that the paper targets to solve?
   - Why they did this research or what is the motivation of this work?
   - What have they done in this work? (one sentence)

   Use Markdown to style your summary, about 100 - 200 words in English is enough.

   Note: You need NOT read the whole paper and get deep into all the details. What you need is to have a basic overview of what the paper is doing. If you encounter some terms or concepts you don't understand, please Google them.

5. Remember to push the `.bib` file and the summary to your Github assignment repository.