DLTSR: A Deep Learning Framework for Recommendations of Long-tail Web Services

ABSTRACT

- With the growing popularity of web services, more and more developers are composing multiple services into mashups.
- Developers show an increasing interest in non-popular services (i.e., long-tail ones), however, there are very scarce studies trying to address the long-tail web service recommendation problem.
- The major challenges for recommending long-tail services accurately include severe sparsity of historical usage data and unsatisfactory quality of description content
- To tackle the problem of unsatisfactory quality of description content, we use stacked denoising auto encoders (SDAE) to perform feature extraction.
- Additionally, we impose the usage records in hot services as a regularization of the encoding output of SDAE, to provide feedback to content extraction.

EXISTING SYSTEM

- With the growing popularity of web services, more and more developers are composing multiple services into mashups.
- Developers show an increasing interest in non-popular services (i.e., long-tail ones), however, there are very scarce studies trying to address the long-tail web service recommendation problem.
- The major challenges for recommending long-tail services accurately include severe sparsity of historical usage data and unsatisfactory quality of description content

DISADVANTAGES

- Frequently used services
- high-performance long-tail

PROPOSED SYSTEM

- we propose to build a deep learning framework to address these challenges and perform accurate long-tail recommendations.
- To tackle the problem of unsatisfactory quality of description content, we use stacked denoising auto encoders (SDAE) to perform feature extraction.
- Additionally, we impose the usage records in hot services as a regularization of the encoding output of SDAE, to provide feedback to content extraction.
- To address the sparsity of historical usage data, we learn the patterns of developers' preference instead of modeling individual services.
- Our experimental results on a real-world dataset demonstrate that, with such joint auto encoder based feature representation and content-usage learning framework, the proposed algorithm outperforms the state-of-the-art baselines significantly.

ADVANTAGES

- Mashup creation
- tackle the problem of unsatisfactory quality of descriptions
- high accuracy

HARDWARE REQUIREMENTS

- Processor - Intel
- Speed - 1.1 GHz
- RAM - 256 MB(min)
- Hard Disk - 20 GB
- Monitor - SVGA

SOFTWARE REQUIREMENTS

- Operating System - Windows 7/8
- Front - End - ASP.NET with C#
- Tools used - Visual studio 2010
- Back – End - SQL SERVER 2005

REFERENCES

- X. Liu and I. Fulia, “Incorporating user, topic, and service related latent factors into web service recommendation,” in Web Services(ICWS), 2015 IEEE International Conference on. IEEE, 2015, pp.185–192.
- Y. Zhong, Y. Fan, K. Huang, W. Tan, and J. Zhang, “Time-aware service recommendation for mashup creation,” IEEE Transactionson Services Computing, vol. 8, no. 3, pp. 356–368, 2015.
- S. M. McNee, J. Riedl, and J. A. Konstan, “Being accurate is notenough: how accuracy metrics have hurt recommender systems,”in CHI’06 extended abstracts on Human factors in computing systems ACM, 2006, pp. 1097–1101.