Intelligent Linux Information Access by Data Mining: the ILIAD Project

Timothy Baldwin,[♠] **David Martinez,**[♡] **Richard B. Penman,**[♠] **Su Nam Kim,**[♠]
Marco Lui,[♠] **Li Wang**[♠] and **Andrew MacKinlay**[♠]
♠ Dept of Computer Science and Software Engineering, University of Melbourne, Australia
♡ NICTA Victoria Research Laboratory

Abstract

We propose an alternative to conventional information retrieval over Linux forum data, based on thread-, post- and user-level analysis, interfaced with an information retrieval engine via reranking.

1 Introduction

Due to the sheer scale of web data, simple keyword matching is an effective means of information access for many informational web queries. There still remain significant clusters of information access needs, however, where keyword matching is less successful. One such instance is technical web forums and mailing lists (collectively termed “forums” for the purposes of this paper): technical forums are a rich source of information when troubleshooting, and it is often possible to resolve technical queries/problems via web-archived data. The search facilities provided by forums and web search engines tend to be over-simplistic, however, and there is a desperate need for more sophisticated search (Xi et al., 2004; Seo et al., 2009), including: favouring threads which have led to a successful resolution; reflecting the degree of clarity/reproducibility of the proposed solution in a given thread; representing threads via their threaded rather than simple chronological structure; the ability to highlight key aspects of the thread, in terms of the problem description and solution which led to a successful resolution; and ideally, the ability to represent the problem and solution in normalised form via information extraction.

This paper provides a brief outline of an attempt to achieve these and other goals in the context of Linux web user forum data, in the form of the ILIAD (Intelligent Linux Information Access by Data Mining) project. Linux users and developers rely particularly heavily on web user forums and mailing lists, due to the nature of the community, which is highly decentralised — with massive proliferation of packages and distributions — and notoriously bad at maintaining up-to-date documentation at a level suitable for newbie and even intermediate users.

2 Project Outline

Our proposed solution is as follows: (1) crawl data from a variety of web user forums; (2) analyse each thread, to identify named entities and generate metadata; (3) analyse post-level linkages; (4) predict user-level features which are expected to impinge on the quality of search results; and finally (5) draw together the features from (1) to (4) to enhance the quality of a traditional ranked IR approach. We briefly review each step below. Given space limitations, we focus on outlining our interpretation of the task in this paper. For further details and results, the reader is referred to the key papers cited herein.

2.1 Crawling

The first step is to crawl data from a variety of forums and mailing lists, for which we have developed open-source scraping software in the form of SITESCRAPER.[1] SITESCRAPER is designed such that the user simply copies relevant content from a browser-rendered version of a given set of pages, which it interprets as a structured record, and translates into a generalised XPATH query.

2.2 Thread-level analysis

Next, we perform named entity recognition (NER) over each thread to identify entities such as package and distribution names, version numbers and snippets of code; as part of this, we perform version

[1] `http://sitescraper.googlecode.com/`

Proceedings of the NAACL HLT 2010 Workshop on Computational Linguistics in a World of Social Media, pages 15–16,
Los Angeles, California, June 2010. ©2010 Association for Computational Linguistics

anchoring, in identifying what entity each version number relates to.

To generate thread-level metadata, we classify each thread for the following three features, based on an ordinal scale of 1–5 (Baldwin et al., 2007):

Complete: Is the problem description complete?

Solved: Is a solution provided in the thread?

Task Oriented: Is the thread about a specific problem?

We additionally automatically classify the nature of the thread content, in terms of, e.g., whether it contains documentation or installation details, or relates to software, hardware or programming.

Our experiments on thread-level classification are based on a set of 250 annotated threads from LinuxQuestions and other forums, as well as a dataset from CNET.

2.3 Post-level analysis

We automatically analyse the post-to-post discourse structure of each thread, in terms of which (preceding) post(s) each post relates to, and how, building off the work of Rosé et al. (1995) and Wolf and Gibson (2005). For example, a given post may refute the solution proposed in an earlier post, and also propose a novel solution in response to the initiating post.

Separately, we are developing techniques for identifying whether a new post to a given forum is sufficiently similar to other (ideally resolved) threads that the author should be prompted to first check the existing threads for redundancy before a new thread is initiated.

Our experiments on post-level analysis are, once again, based on data from LinuxQuestions and CNET.

2.4 User-level analysis

We are also experimenting with profiling users variously, based on a 5-point ordinal scale across a range of user characteristics. Our experiments are based on data from LinuxQuestions (Lui, 2009).

2.5 IR ranking

The various features are interfaced with an ad hoc information retrieval (IR) system via a learning-to-rank approach (Cao et al., 2007). In order to carry out IR evaluation, we have developed a set of queries and relevance judgements over a large-scale set of forum data.

Our experiments to date have been based on combination over three IR engines (LUCENE, ZETTAIR and LEMUR), and involved thread-level metadata only, but we have achieved encouraging results, suggesting that thread-level metadata can enhance IR effectiveness.

3 Conclusions

This paper provides an outline of the ILIAD project, focusing on the tasks of crawling, thread-level analysis, post-level analysis, user-level analysis and IR reranking. We have designed a series of class sets for the component tasks, and carried out experimentation over a range of data sources, achieving encouraging results.

Acknowledgements

NICTA is funded by the Australian Government as represented by the Department of Broadband, Communications and the Digital Economy and the Australian Research Council through the ICT Centre of Excellence program.

References

T Baldwin, D Martinez, and RB Penman. 2007. Automatic thread classification for Linux user forum information access. In *Proc of ADCS 2007*.

Z Cao, T Qin, TY Liu, MF Tsai, and H Li. 2007. Learning to rank: from pairwise approach to listwise approach. In *Proc of ICML 2007*.

M Lui. 2009. Impact of user characteristics on online forum classification tasks. Honours thesis, University of Melbourne. `http://repository.unimelb.edu.au/10187/5745`.

CP Rosé, B Di Eugenio, LS Levin, and C Van Ess-Dykema. 1995. Discourse processing of dialogues with multiple threads. In *Proc of ACL 1995*.

J Seo, WB Croft, and DA Smith. 2009. Online community search using thread structure. In *Proc of CIKM 2009*.

F Wolf and E Gibson. 2005. Representing discourse coherence: A corpus-based study. *Comp Ling*, 31(2).

W Xi, J Lind, and E Brill. 2004. Learning effective ranking functions for newsgroup search. In *Proc of SIGIR 2004*.