DUPDETECT - IMPLEMENTATION OF A NEAR-DUPLICATE DETECTION SYSTEM ON USER GENERATED FORUM CONTENT.

Task for Komplexpraktikum „Entwicklung mobiler und verteilter Systeme“

Winter Semester 2009/2010, Supervisors: Klemens Muthmann

INTRODUCTION

Everyone knows forums on the Internet where users discuss highly important and current topics about their favorite interests. Software developers for example talk about the most recent bugs and tricks to develop cutting edge programs, World of Warcraft players talk about their favorite class-race combinations or the newest update of the game and Linux users find advice for tailoring their system exactly to their needs.

Imagine an Ubuntu Linux user who wants to set up server to accept remote desktop connections from his laptop over the local LAN. Because of his unusual network configuration he runs into problems and several tries later gives up and searches the solution on the Internet. Unfortunately the official documentation seems not to apply for his scenario, so he searches the web forum of his Linux distribution and gets a huge list of 250 results about this and similar topics (see Figure 1). After scanning the first five threads he finds a “wannabe” solution and tries it. Unfortunately this solution does also not work for his scenario. So he continues scanning result after result, skipping the ones with non-promising titles. Besides a little additional information, he reads useless or redundant contributions in every thread he visits. After several hours of skimming through threads, editing several configuration files in different ways and sending command after command to his terminals he has a quite good overview of what is going wrong. Unfortunately he did not solve his problem but is tired of keeping on searching. So he creates a new forum thread stating what he already tried and hopes someone has a deeper insight into all this remote desktop stuff. Although this produces additional possibly unnecessary traffic and another duplicate of all the explanations already given, this action is absolutely comprehensible. However he might still have missed the already existing solution to his problem.
There are many forums for nearly all areas of interest on the web. On some of them software companies provide a platform to discuss their products while others are used by various hobbyists, to talk to like minded people. The well known Microsoft Developer Network for example is a big platform to discuss about topics like C#, Silverlight and Visual Studio. While all of them contain a huge set of information, they also face their users with problems similar to the one described above. It does not matter whether one is searching for the solution of an unexpected Java exception or for opinions of the new Paladin re-balancing in his favorite Massive Multiplayer Online Role Playing Game. The source of the problem is that data in web forums is mostly hidden in statically and shallow organized free written text.

Figure 1: Example Search on ubuntuforums.com

Although forum search engines are designed to provide access to this data, they only touch the tip of the iceberg. This is due to the fact that they are only keyword based. So it often happens that a forum user is not able to find an answer to his question even if this answer already exists. Therefore he creates a new discussion and thus produces a duplicate of a question already posed before.

Duplicates are a big problem in forums. They hide the really important questions, they raise the load on hardware, software and moderators and make it even harder to spot interesting answers. If it would be possible to reliably detect such duplicates a multitude of applications becomes possible. For example one could shorten long search result lists by showing groups of very similar threads as one result, one could capture the most discussed topics and one could get grasp of questions that are often asked but never answered.
IMPLEMENTATION

A System to detect near-duplicates might look similar to the architecture presented in Figure 2). The base data store is one or several forums. This data is extracted with a web crawler optimized for the forum structure and stored to a database. Since duplicate detection is not easy on unstructured content, an Unstructured Information Management (UIM) System needs to enrich the extracted data with structured features like “Part-of-Speach” (PoS) Tags (nouns, verbs, adjectives) or entities (products, persons, names, locations).

Duplicate detection is carried out based on the structured features. It runs in a three step process. The first step creates a candidate set to reduce the amount of necessary computation during further steps. The second step does the pairwise duplicate detection and finally the last step groups all duplicates. The groups are saved to a data store again. This data store serves applications such as search engines or data mining to present grouped search results or purge duplicate content in the forum.

SUBTASKS

- Learn to know necessary algorithms for the individual subsystems
- Requirements analysis for all the subsystems
- Modeling the detailed architecture of the subsystems
• Definition of interfaces between the subsystems
• Implement the subsystems
• Integrate the subsystems with each other and verify that the system works