Combining Structural Information and the Use of Priors in Mixed Named-Page and Homepage Finding

Paul Ogilvie and Jamie Callan
Language Technologies Institute
School of Computer Science
Carnegie Mellon University
{pto, callan}@cs.cmu.edu

Abstract

This paper presents Carnegie Mellon University's experiments on the mixed named-page and homepage finding task of the TREC 12 Web Track. Our results were strong; we achieved the success using language models estimated from combining information from document text, in-link text, and information present in the structure of the documents. We also present experiments using expectations about posterior distributions to create class-based prior probabilities. We find that priors do provide a large gain for our official runs, but we do further experiments that show the priors do not always help. Some preliminary analysis shows that the prior probabilities are not providing the desired posterior distributions. In cases where applying the priors harm performance, the observed posterior distributions in the rankings are far off of the desired posterior distributions.

1. Introduction

Documents found on the Internet are rich in structure, and this provides many information sources useful for retrieval. In particular, structural information has been found useful for known-item searches such as homepage finding and named-page finding [Craswell 2001][Kraaij 2002][Ogilvie 2003][Zhang 2002]. A known-item retrieval system should attempt to leverage the structural information in a manner that is consistent with its model and provides an improvement in retrieval performance.

In this paper, we present experiments where we combine structural information using language modeling. We create several document representations for each document using the structural information present in HTML documents. From these document representations, we form language models. We combine the language models using a linear interpolation to form a new language model. This new language model is then used to estimate the probability that the document has generated the query.

It is also possible to leverage query independent information through the use of document priors. Document priors are probabilities or beliefs that the document is relevant to the query independent of any knowledge about the query. In previous homepage finding experiments, [Kraaij 2002] found a prior based on the type of the URL to be a very effective source of information. The URL types ROOT, SUBROOT, FILE, and PATH form four distinct document classes. We hypothesize that these classes will also be useful for a mixed homepage/named-page finding task, and present experiments using these priors. The prior probabilities are estimated from training data which gives us a desired posterior distribution. This desired posterior distribution defines ratios of the document classes what we would like to observe in the rankings. Using Bayes' rule, we can estimate the prior probabilities from training data and corpus statistics. A detailed derivation is provided Section 5.

In the next section, we describe the basic generative language model where we are using information from only one document representation. Section 3 describes combining information from several language models. Section 4 briefly discusses system specifics for the experiments. Section 5 provides a detailed derivation of the priors and posterior distribution we used for the mixed homepage/named-page finding track of the TREC 12 Web Track, and Section 6 describes our official runs and other experiments. Section 7 provides a discussion of the URL priors and their effectiveness in producing the desired posterior distributions over the rankings. We state conclusions in Section 8.

2. Generative Language Models

A unigram language model defines a multinomial probability distribution over all words in the vocabulary of the corpus. These probabilities are interpreted as word generation probabilities, and documents are ranked by their probability of generating the query. This generation probability is computed by taking the product over all query terms of the probability of the query term given the language model [Zhai 2001]:

$$\left(\mid D \right) = \prod_{=1}^{|Q|} \left(\mid D \right)$$