Focused Retrieval of University Course Descriptions from Highly Variable Sources

Thomas Effland - SUNY University at Buffalo

Motivating Question

How can we automatically retrieve semantically similar content (such as university course descriptions) from many disparate sources on the Web that do not reference each other when we only know the domain names and have limited computational and training resources?

Challenges

- Target content is typically very sparse on large sites, so brute force crawling is unreasonable.
- Organizational structure and content location vary highly for each site, thus canonical rule-based approaches are ineffective.
- Typical topical-locality assumptions made in focused web crawling do not hold when sites do not reference each other.
- Retrieving relevant content requires identifying and tunneling through irrelevant pages that do not reference each other.
- Gathering hand-labeled data is costly.

Webpage Representation

Each page is represented by a feature vector of the page content and a set of feature vectors for each link on the page.

Page Features

- TF-IDF of words and bigrams of segmented url
- TF-IDF of words and bigrams of the title
- Latent Semantic Analysis (LSA) of TF-IDF of words and bigrams of page body words

Url Features

- TF-IDF of words and bigrams of segmented url
- TF-IDF of words and bigrams of the link anchor text

Algorithm & System Design

Defining a Relevance Metric

To obtain a measure of how close an irrelevant page is to a target content page, we define the label R of a page as the normalized link distance from the page to the target. This relevance metric helps address variable structure of sites.

Training Stage

1) User marks sample traversal paths using a simple Chrome extension called MarkIt.
2) Labeled and unlabeled data are collected from paths and training data is extracted.
3) Two Random Forest Regressors [5] are fit to training data.
4) Regressors are used to generate more training data by ranking pages and labeling highly ranked pages or asking user for input on middle ranked pages. This semi-supervised approach combines self-training [6] with active learning [7] and saves considerable time in generating large training set from significantly less intervention.

This yields flexible regressors from only a small number of sample traversals from a few sites.

Deployment Stage

1) Top level url is input in queue. Queue pops top url.
2) Features for each url on page are extracted and urls are ranked by relevance prediction, then pushed into priority queue.
3) Features for page content are extracted and relevance is predicted. If relevance prediction for the page is high and agrees within threshold amount with initial prediction from its url, the page is classified as a target page. The user may review the proposed retrieval and classify the prediction as correct or not and the page is added to the training data. This is an example of active learning.
4) The next page from the queue is popped and the cycle continues until user-specified endpoint.

Results

Here we show how the absolute regression error for the page and url regressors improves by using self-training. Regressors for each school were trained on data from the other schools. The thick green lines show the average among all five schools.

The best generalization scores are labeled with stars and are 0.048 and 0.052 for the page and url regressors respectively.

This graph shows how we are able to automatically train incrementally more accurate general regressors, starting with less than 100 initially labeled pages.

Conclusions & Impact

- General Information Retrieval framework in problem domain where seed pages are irrelevant and no topical locality assumption.
- Novel reformulation of page relevance as normalized link-distance.
- Novel combination of self-training and active-learning for focused crawling on little training data.
- Novel focused-crawling architecture.

References