**IR Project**

Title: Implement n-gram language model based retrieval and ranking within Galago. Length of n-grams must vary from 1 to N=5.

Description:

Language modeling based retrieval system is based on the query likelihood scoring method proposed by Ponte and Croft. For each document in the collection, a language model is estimated and the documents are ranked based on the likelihood of the query according to the estimated language model. In a language model, the query is assumed to be a sample of words drawn according to a model estimated based on a document.

In query likelihood model, a language model *Md is constructed* from each document *d. The ranking of documents is given by P* (*d*|*q*), where the probability of a document is interpreted as the likelihood that it is relevant to the query. Using Bayes rule:

*P*(*d*|*q*) = *P*(*q*|*d*)*P*(*d*)/*P*(*q*)

The Language Modeling approach attempts to model the query generation process and the documents are ranked by the probability that a query would be observed as a random sample from the document model. The docuements are ranked using a multinomial unigram language model where the documents are the classes and each document is treated in the estimation as a separate “language”. Using this



Where, Kq is the multinomial coefficient for the query *q*, given as:



The retrieval of documents based on language models can now be described as :

1. Estimating the language model Md for each document in the collection.
2. Estimate *P* (*q*|*Mdi*), the probability of generating the query according to each of these document models
3. Rank the documents according to these probabilities

The language model assumes that the user has a prototype document in mind and generates a query based on distinguishable words that are likely to appear in the document.

**Estimating the query generation probability:**

Using the unigram assumption (each query term is independent for each other), the probability of producing the query given the language model *Md* of document *d* using maximum likelihood estimation (MLE):



Where, *Md* is the language model of document *d*, tf*t*,*d* is the term frequency of term *t* in document *d*, and *Ld* is the number of tokens in document *d*.

**Variants of the Basic Language Modeling Approach**

If for some words that do not appear in the document, *P* (*t*|*Md*)ius estimated to be zero. This means that documents will only give a query non-zero probability if all of the query terms appear in the document. Therefore, to smooth probabilities in the document language models, we need to discount non-zero probabilities and give some probability mass to unseen words.

The basic language modeling approach (i.e., the query likelihood scoring method) can be instantiated in different ways by varying:

(1) Q*D* (e.g., multiple Bernoulli or multinomial)

(2) Estimation methods of Q*D*(e.g., different smoothing methods)

(3) The document prior *p*(*D*).

The original proposed language model proposed by Ponte and Croft used the multiple Bernoulli models that ignore query term frequencies. The multinomial model captures the term frequency in documents (as well as the query). In multiple Bernoulli model, the presence/absence of a term is assumed to be independent of that of other terms, whereas in multinomial model, every word occurrence is assumed to be independent, including the multiple occurrences of the *same* term.

Estimation of Q*D* is critical for the success of language models, and a particularly important issue is how to smooth the maximum likelihood estimate which assigns zero probability to unseen words. Different smoothing techniques have been proposed in literature.

**Dirichlet prior smoothing** works well especially for keyword queries as it adjusts the amount of smoothing according to the length of a document.

The Dirichlet prior smoothing method can be derived by using Bayesian estimation:



where *p*(*w|C*) is a background (collection) language model and *μ* is a smoothing parameter.

**Linear Interpolation smoothing**: Uses a mixture between a document-specific multinomial distribution

and a multinomial distribution estimated from the entire collection:



where 0 < *l* < 1 and *Mc* is a language model built from the entire document collection. This mixes the probability from the document with the general collection frequency of the word

**Types of language models**

To find the probabilities over sequences of terms, we can use the chain rule to decompose the probability of a sequence of events into the probability of each successive event conditioned on earlier events:

****

**Unigram models:** The simplest form of language model simply throws away all conditioning context, and estimates each term independently. Such a model is called a UNIGRAM LANGUAGE *unigram language model*:

****

**Bigram models:** In a bigram language model, the generation of a current word would be dependent on the previous word generated, thus it can potentially capture the dependency of adjacent words (e.g., phrases). Specifically, the query likelihood would be

****

****

Where, p(*qi|qi−*1*,D*) is the conditional probability of generating *qi* after we have just generated *qi−*1.

Such n-gram models capture dependency based on word positions. With increasing n, complexity of the models increases that makes it difficult to obtain accurate estimation of the model.

**Structured Document Retrieval and Combining Representations:**

Information retrieval systems generally work on bag of words representation of the documents and ignore the structural information of the document. A document may have intra-document structures (e.g., title vs. body) and inter-document structures (e.g., hyperlinks and topical relations), which can be used to improve the performance of the retrieval system.

In this project, we will divide each document is composed of several fields/context (such as title and body). The field structure of documents allows each field to have weights depending on the field and query-term. Each document d in the collection is composed of fields (f1, f2, … fn). The project will assign different weights to different parts of the document. The query generation process consists of two steps:

1. A part *fi* is selected from the structured document *D* according to a selection probability *p*(*fi|D*). It can be interpreted as the weight assigned to Di and can be set based on prior knowledge or estimated using training data.
2. A query is generated using the selected part *fi*. Thus, the query likelihood is given by:



**Replace Dj with fj in the equation above**

**Description of the approach:**

1. **Inverted Index:** We will use galago to implement the inverted index with the position listing. Using the inverted index, we can estimate the probability of a query term occurring in the document as well as the whole collection.
2. **To incorporate n-gram language models:** we will estimate the conditional probabilities for sequence of events conditioned on earlier events. This will form the basis for extending the model beyond the uni-gram model (where each term is independent of each other).
3. **Algorithm:** We will implement the query likelihood model for semi-structured documents using n-gram language models. For each document, a language model will be constructed for different fields and will be combined with different weights.
4. **Evaluation**: We will use user feedback to annotate the top ten retrieved results as relevant or irrelevant. Further we will evaluate the performance of our retrieval system in terms of precision. We will also check our system for keyword queries as well as verbose queries.