

INVERTED INDEXING OF LINEAR CLASSIFIERS

The purpose of this webpage is to discuss the data preparation, and classifier training for using inverted indexing of linear classifiers. The discussion focuses specifically on using models produced by BBR (sparse logistic regression software). These instructions may overlap somewhat with suggestions given elsewhere in the AFS documentation.

The software required, in addition to standard Unix/Linux utilities is:

- [Lemur Toolkit](#),
- [BBR](#) Software

The full process consists of the following steps:

1. Corpus Preparation
2. Construction of Lemur Index for the corpus
3. Training of BBR classifiers separately for each topic and extraction of model vectors

Each of these steps is discussed in some detail below.

1. Corpus Preparation

Complete data for the corpus consists of the following:

- Files of documents (train.docs, all.docs) in the format:


```
< DOCNO> doc_id
< TEXT>
text
< /TEXT>
< /DOC>
```

 spaces were inserted to make the markup visible
- list of topics in the sgml format (< DOC 1541>) and in the plain format (just one id per line)
- judgement file, in the form: "topic-id doc-id relevance". Relevance is 0 (nonrelevant) or 1 (relevant). Usually only relevant entries are kept and missing entries imply nonrelevance.

2. Construction of Lemur Index for the corpus

Lemur Index can be constructed by using the following calls to Lemur programs:

```
# parse document files
/usr/local/lemur/bin/ParseToFile parse_stem_train_param train.docs
/usr/local/lemur/bin/ParseToFile parse_stem_all_param all.docs

# build indices
/usr/local/lemur/bin/BuildBasicIndex build_train_param > log-build-train
/usr/local/lemur/bin/BuildBasicIndex build_all_param > log-build-all
```

Examples of the parameter files (parse_stem_train_param, parse_stem_all_param, build_train_param, build_all_param) are available [online](#). Once the Indices are constructed, the short path names in the *.bsc files should be changed to the full paths.

3. Training of BBR classifiers and extraction model vectors

The vector representation of the training data can be obtained with Index2SVM utility included in the AFS.

The training stage consists of applying BBRclassify to the training examples with appropriate labels. This involves extracting labels for a particular topic, combining them with the actual training vectors and extracting the model vector. You may be interested in a [sample script](#) for doing this.

In particular, an awk script is used to extract sparse vector representation of the model from the model file, with intercept represented as a feature with id=-1:

```
awk -v t=$topic '/feat/{for(i=1;i<=NF;i++){a[i]=$i;}}
/beta/{printf("%s",t); for(i=2;i < NF;i++){if($i)!=0{printf(" %d:%s",a[i],$i);}}printf(" -1:%s\n",$(NF));}'
model-$topic > m-$topic
```

Once the training is complete, the sparse model files are concatenated and sorted by their id:

```
cat m-* | sort -n +0 > all-models.dat
```