**Building the Language Identifier code**

Either compile the source code using Visual Studio, or run the file make.bat to build the library and the executables using the C# compiler that comes standard with Windows.

**Obtaining Wikipedia training data**

Download either Wikipedia abstracts or full Wikipedia documents for the languages you want to train on. The URLs are of the pattern:

http://dumps.wikimedia.org/enwiki/20130503/enwiki-20130503-abstract.xml

http://dumps.wikimedia.org/enwiki/20130503/enwiki-20130503-pages-articles.xml.bz2

where “20130503” is the date on which the Wikipedia dump was written out, and “en” is a ISO 639 language code.

**Parsing the Wikipedia files**

Create a file listing all the Wikipedia XML files you downloaded, for example like this:

dir /b \*wiki\*.xml > wiki-files.txt

Then extract the abstract or content from the Wikipedia files. To extract the abstracts, run:

e:\>ParseWikipedia wiki-files.txt abstract

Done with ar. Wrote 154105 docs.

…

**Generating language profiles**

Create a file listing all the files that were written out by ParseWikipedia, for example like this:

e:\>dir /b \*\_parsed.txt > parsed-files.txt

Next, build the language profiles. To build language profiles containing all character {1,…,5}-grams without first converting text to lower-case (“no-case-folding”) and after cleaning out any Wikipedia-specific markup, run:

e:\>CompileDistro char 1 5 ncf -1 wiki parsed-files.txt langprofiles-char-1\_5-nfc-all.bin.gz

To build language profiles containing the top ten thousand word unigrams per language, after converting text to lower-case and after cleaning out any Wikipedia-specific markup, run:

e:\>CompileDistro word 1 1 tlc 10000 wiki parsed-files.txt langprofiles-word-1\_5-nfc-10k.bin.gz

**Using language profiles**

To get an idea on how to use the Language Identifier, take a look at the file TestlanguageIdentifier.cs. There are two key operations:

The static method LanguageIdentifier.New(langProfilesFile, liSpec, cap) creates and returns a new LanguageIdentifier object. The string langProfilesFile refers to a file produced by running CompileDistro, containing the profiles of all the languages that were listed in parsed-files.txt. The string liSpec declares what language identification method should be used. “Vector” indicates that the cosine-simililarity based approach should be used; “Likely” indicates that the conditional-likelihood-based approach should be used; and “Rank,sf” or “Rank,sr” indicate that the rank-based approach (using Spearman’s footrule or Spearman’s rho, respectively) should be used. Finally, the integer cap indicates how many grams should be saved per language; a value of -1 indicates that all grams should be used.

Any LanguageIdentifier object provides a method Identify(text), which takes a string text and returns a string indicating the ISO 639 language code of the language that text is predicted to be in.