Character Encoding Identification Algorithm

Need:

- Large number of languages across the world with incompatible encodings (No standard encoding)
- Different communication protocols with different tagging mechanisms for the charset for delivering the content. (HTML, HTTP etc.)
- International e-mails as a medium of communication and business between people of different native languages, encountering annoying messages due to the wrong encoding.

Existing Implementation:

The most generic way of indicating the encoding is through the file header.

Reference - https://www-archive.mozilla.org/projects/intl/universalcharsetdetection

Devised by the Mozilla Foundation in 2002 after the detection algorithm was specified in a paper (specifically G. Kikui in 1996). It is a composite method including all the three methods discussed below.

The initial three methods included:

1. Coding Scheme method:

This method is a proof by contradiction method often used with multi-byte encodings. It eliminates a certain encoding if an illegal byte sequence is encountered while verifying a certain encoding.

Good for 7-bit multi-byte encodings like ISO-2022-xx, HZShift_JIS and EUC-JP, but not for others like EUC-CN and EUC-KR. It is not very useful for single-byte encodings.

2. Character Distribution method:

A statistical method which exploits the facts that in any language, some characters and used more often than other characters. This can be used to detect encoding for languages a with large number of distinct characters i.e. Chinese, Japanese and Korean.

3. Two-Char Sequence Distribution Method:

For the languages using smaller number of characters, another method can be utilized by grouping two characters and finding the sequence distribution of this which is encoding and language dependent. Useful in detecting single-byte encodings (Russian) but not efficient with multi-byte encoding.

The composite method is summarized by:

a. Most web pages are still encoded in ASCII. This top-level control algorithm begins with an ASCII verifier. If all characters are ASCII, there is no need to launch other detectors except ISO-2022-xx and HZ ones.
b. ISO-2022-JP or KR and HZ detectors are launched only after encountering ESC or ~{, and they are abandoned immediately when a byte is met.
c. BOM is being searched to identify UCS2. We found that some web sites send 0x00 inside http stream, and using this byte for identifying UCS2 proved to be unreliable.
d. If any one of the active detectors received enough data and reaches a high level of confidence, the entire auto-detecting process will be terminated and that charset will be returned as the result.