

Unicode in Natural Language Processing Applications

Tom Emerson
Sr. Computational Linguist

strategy • process • technology • results

www.basistech.com

Overview

- Why is Unicode important in NLP?
- Examples from Asian language MA
 - Not a big deal for European languages where there is often a single encoding (Russian excepted)
- Examples from Basis Technology's language analyzers

NLP Functionality

- **Tokenization**
 - Word, sentence, and paragraph
- **Lexical Analysis**
 - Finding word boundaries, compounds, names,...
- **Entity Extraction**
 - Person names, company names, dates, numbers, and most anything else.
- **Parsing**

Character Set Issues in NLP

- Often not even thought about
 - System designed for one language
 - System designed for one encoding of one character set of a language
- Scalability and portability issues
 - Porting to different operating systems
 - EUC-JP vs. CP932
 - Multiple encodings for different uses:
 - EUC-KR vs. JOHAB

Attractiveness of Unicode in NLP

- Characters have defined semantic properties which are language independent
 - Facilitates generic tokenisation of many languages/scripts
 - Word, sentence, and paragraph boundaries
 - Case-folding support for multiple languages
 - Canonical Decomposition allows for advanced “fuzzy” dictionary searching in a standardised way

Tokenization

- Complex cases need to be handled:
 - “don’t use tree@basistech.com, O.K.? Good.”
- The semantic information assigned to characters facilitates generic tokenization
 - UTR #14: Line Breaking Properties
 - UTR #18: Unicode Regular Express Guidelines
 - Word boundary detection
 - 10 character classes
 - 10 states
 - Works well for space-separated languages.

Tokenization

- What about Asian text?
 - Script boundaries are useful
 - Katakana, hiragana, kanji, and roomaji can be trivially detected. Not perfect, but is a first approximation.
 - Hanzi and non-hanzi divisions also easily detected.
 - Punctuation easily found
 - Numbers are easily found
 - But: mixed latin/han numerals are common and need to be handled.
 - And: you will find characters you don't expect being used in numbers.

Tokenization/Analysis

- Korean Processing Requires Decomposition
 - Individual eumjeol provide the top-level character analysis unit
 - However, when looking at inflectional endings, you really need to work with individual Jamo.
 - Requires use of two character encodings unless you're using Unicode: KSC 5601 and Johab.
 - In Unicode, one encoding with canonical decomposition from precomposed Han'gul to Jamo.

Advantages in MA Design

- Asian Languages
 - Our approach to handling Japanese and Chinese is similar and can be shared between implementations.
 - The use of Unicode means encoding issues are irrelevant (to us):
 - EUC-JP? CP932? ISO 2022-JP? ISO 2022-JP2?
 - EUC-CN? EUC-CN-EXT? Big Five? CP936? HKSCS? GB 18030?
 - Whatever.
 - Common analyzer engine and common dictionary formats which are language neutral at their core.

Advantages in MA Design

- Canonical decomposition make variant handling quite clean:
 - schön
 - schoïn
 - schoen
 - schon
- This makes it quite possible to easily handle these cases in the dictionary.

API Design

- A single character representation means that you can implement multiple languages through a single API, reducing special casing in customer code.
- Basis Technology has a single API for:
 - Chinese
 - Japanese
 - Korean
 - German
 - Arabic
 - “Generic”

Complications

- Unicode is large
 - Requires careful implementation of some commonly used algorithms in NLP, e.g. finite-state automata.
 - Active area of research in industry and academia
 - Multiple ways of handling these cases
- False Security
 - Unicode is not a panacea for all problems in NLP text processing.

Test Data

- XML provides a very convenient format for storing gold-standard test data.
- “Dagobah” DTD is used at Basis Tech for test data
 - Associates input with expected analyzer output
 - Specification of analyzer options within the data file
 - Allows entire operation of an analyzer to be driven by data for testing purposes.

Conclusion

- Unicode offers many advantages in the engineering and testing of NLP applications.
- Requires careful implementation in some cases
- Allows a lot of code reuse, which is a Good Thing.

Questions?