Alphabets, Encoding, Language Recognition

Daniel Zeman, Rudolf Rosa

February 28, 2020




Charles University
Faculty of Mathematics and Physics
Institute of Formal and Applied Linguistics

Language Recognition

- Very useful:
  - You crawl the web to get data
  - You get various languages
  - You need to know what they are
- Document level
- Paragraph or sentence level
- Intra-sentence level, code switching

How to Recognize the Language?

- Dictionary
  - Good if you have it
  - Large data needed to obtain it
  - Problems with coverage

How to Recognize the Language?

- Dictionary
  - Good if you have it
  - Large data needed to obtain it
  - Problems with coverage
- What if no dictionary is available?
  - Hint: this doesn't seem to be a European language:
    ירושלים של זהב

How to Recognize the Language?

- Dictionary
  - Good if you have it
  - Large data needed to obtain it
  - Problems with coverage
- What if no dictionary is available?
  - Hint: this doesn't seem to be a European language:
    ירושלים של זהב
- Specific letters within Latin
  - $\check{R} \Rightarrow$ Czech?
  - $\emptyset \Rightarrow$ Danish / Norwegian?

How to Recognize the Language?

- Dictionary
  - Good if you have it
  - Large data needed to obtain it
  - Problems with coverage
- What if no dictionary is available?
  - Hint: this doesn't seem to be a European language:
    ירושלים של זהב
- Specific letters within Latin
  - $\check{R} \Rightarrow$ Czech?
  - $\emptyset \Rightarrow$ Danish / Norwegian?
- But most letters are shared among multiple languages
- And this is German!
  Die Burg wurde vom böhmischen König Přemysl erobert.

Character Frequency

- Czech ... O E N A T ... X Ď Q Ö Ł
- Slovak ... O E A N R ... X Ď Ĺ Ŕ W
- Russian ... О А Е И Н ... Ц Щ Ф Э Ъ
- English ... E I S N T ... K X Q J Z
- German ... E N T S R ... Ö J Y X Q
- Spanish ... E A I O R ... É X Q Ú Ñ
- French ... E I S T N ... Î Ô W K Â

Character Frequency

- Relative frequency of letter X in "language" (training document) … $j_X \in \langle 0; 1 \rangle$
- Relative frequency of letter X in tested document … $t_X \in \langle 0; 1 \rangle$
- Vector of frequencies of all letters (typically 20–50 letters)
- Some measure of vector distance, e.g.

  $d = \sqrt{(j_A - t_A)^2 + \ldots + (j_Z - t_Z)^2}$
- Some measure of vector similarity, e.g.

  $p = 1 - \frac{|j_A - t_A| + \ldots + |j_Z - t_Z|}{2}$

Similarity of Character Vectors

- Larger than a threshold $\Rightarrow$ same language
- Or:
  - Train on "all" languages
  - Find the closest language

Similarity of Character Vectors

- Larger than a threshold $\Rightarrow$ same language
- Or:
  - Train on "all" languages
  - Find the closest language
- Universal Declaration of Human Rights (UDHR)
- Bible
- Watchtower
- Wikipedia

Character N-grams

1. *che, sch, der, ich, ter, ung*

Character N-grams

1. *che, sch, der, ich, ter, ung*
2. *ver, gen, ing, ste, aar, ijn*

Character N-grams

1. *che, sch, der, ich, ter, ung*
2. *ver, gen, ing, ste, aar, ijn*
3. *ing, ion, tio, ent, ati, ers*

Character N-grams

1. *che, sch, der, ich, ter, ung*
2. *ver, gen, ing, ste, aar, ijn*
3. *ing, ion, tio, ent, ati, ers*
4. *pre, ova, nov, vol, nie, ový*

Character N-grams

1. *che, sch, der, ich, ter, ung*
2. *ver, gen, ing, ste, aar, ijn*
3. *ing, ion, tio, ent, ati, ers*
4. *pre, ova, nov, vol, nie, ový*
5. *ova, ých, ost, ová, ick, ého*

Character N-grams

1. *che, sch, der, ich, ter, ung*
2. *ver, gen, ing, ste, aar, ijn*
3. *ing, ion, tio, ent, ati, ers*
4. *pre, ova, nov, vol, nie, ový*
5. *ova, ých, ost, ová, ick, ého*
6. *szt, tás, asz, ban, ala, sza*

Character N-grams

1. *che, sch, der, ich, ter, ung*
2. *ver, gen, ing, ste, aar, ijn*
3. *ing, ion, tio, ent, ati, ers*
4. *pre, ova, nov, vol, nie, ový*
5. *ova, ých, ost, ová, ick, ého*
6. *szt, tás, asz, ban, ala, sza*
7. *nie, dzi, ego, kie, rze, ych*

Character N-grams

1. *che, sch, der, ich, ter, ung*
2. *ver, gen, ing, ste, aar, ijn*
3. *ing, ion, tio, ent, ati, ers*
4. *pre, ova, nov, vol, nie, ový*
5. *ova, ých, ost, ová, ick, ého*
6. *szt, tás, asz, ban, ala, sza*
7. *nie, dzi, ego, kie, rze, ych*
8. *ent, tat, ato, est, ion, zio*

Character N-grams

1. *che, sch, der, ich, ter, ung*
2. *ver, gen, ing, ste, aar, ijn*
3. *ing, ion, tio, ent, ati, ers*
4. *pre, ova, nov, vol, nie, ový*
5. *ova, ých, ost, ová, ick, ého*
6. *szt, tás, asz, ban, ala, sza*
7. *nie, dzi, ego, kie, rze, ych*
8. *ent, tat, ato, est, ion, zio*
9. *ado, nte, dad, ent, art, cto*

Character N-grams

1. *che, sch, der, ich, ter, ung*
2. *ver, gen, ing, ste, aar, ijn*
3. *ing, ion, tio, ent, ati, ers*
4. *pre, ova, nov, vol, nie, ový*
5. *ova, ých, ost, ová, ick, ého*
6. *szt, tás, asz, ban, ala, sza*
7. *nie, dzi, ego, kie, rze, ych*
8. *ent, tat, ato, est, ion, zio*
9. *ado, nte, dad, ent, art, cto*
10. *ion, ent, tio, ale, eme, les*

Character N-grams

1. *che, sch, der, ich, ter, ung*
2. *ver, gen, ing, ste, aar, ijn*
3. *ing, ion, tio, ent, ati, ers*
4. *pre, ova, nov, vol, nie, ový*
5. *ova, ých, ost, ová, ick, ého*
6. *szt, tás, asz, ban, ala, sza*
7. *nie, dzi, ego, kie, rze, ych*
8. *ent, tat, ato, est, ion, zio*
9. *ado, nte, dad, ent, art, cto*
10. *ion, ent, tio, ale, eme, les*
11. *али, вал, ост, про, при*

Character N-grams

1. *che, sch, der, ich, ter, ung*
2. *ver, gen, ing, ste, aar, ijn*
3. *ing, ion, tio, ent, ati, ers*
4. *pre, ova, nov, vol, nie, ový*
5. *ova, ých, ost, ová, ick, ého*
6. *szt, tás, asz, ban, ala, sza*
7. *nie, dzi, ego, kie, rze, ych*
8. *ent, tat, ato, est, ion, zio*
9. *ado, nte, dad, ent, art, cto*
10. *ion, ent, tio, ale, eme, les*
11. *али, вал, ост, про, при*
12. *ent, ute, acu, cut, aci*

Character N-grams

1. *che, sch, der, ich, ter, ung*
2. *ver, gen, ing, ste, aar, ijn*
3. *ing, ion, tio, ent, ati, ers*
4. *pre, ova, nov, vol, nie, ový*
5. *ova, ých, ost, ová, ick, ého*
6. *szt, tás, asz, ban, ala, sza*
7. *nie, dzi, ego, kie, rze, ych*
8. *ent, tat, ato, est, ion, zio*
9. *ado, nte, dad, ent, art, cto*
10. *ion, ent, tio, ale, eme, les*
11. *али, вал, ост, про, при*
12. *ent, ute, acu, cut, aci*

- de (German)
- nl (Dutch)
- en (English)
- sk (Slovak)
- cs (Czech)
- hu (Hungarian)
- pl (Polish)
- it (Italian)
- es (Spanish)
- fr (French)
- ru (Russian)
- (es) á

Trigrams with Word Boundaries

1. *sch, che, ung, ten, ich, gen*
2. *ing, ver, nde, gen, oor, ijn*
3. *ing, ion, tio, ent, ati, ter*
4. *ova, val, ali, pre, ala, nie*
5. *ova, ých, ost, ová, ick, ého*
6. *sze, meg, nak, ban, szt, nek*
7. *nie, rze, owa, dzi, prz, rzy*
8. *ent, ion, nte, ato, zio, con*
9. *ent, ado, nte, con, ica, ada*
10. *ent, ion, ant, tio, que, ati*
11. *ова, ост, енн, ого, льн, про*

- *en], sch, er], che, ung, ten*

Trigrams with Word Boundaries

1. *sch, che, ung, ten, ich, gen*
2. *ing, ver, nde, gen, oor, ijn*
3. *ing, ion, tio, ent, ati, ter*
4. *ova, val, ali, pre, ala, nie*
5. *ova, ých, ost, ová, ick, ého*
6. *sze, meg, nak, ban, szt, nek*
7. *nie, rze, owa, dzi, prz, rzy*
8. *ent, ion, nte, ato, zio, con*
9. *ent, ado, nte, con, ica, ada*
10. *ent, ion, ant, tio, que, ati*
11. *ова, ост, енн, ого, льн, про*

- *en], sch, er], che, ung, ten*
- *en], ing, ver, er], nde, de]*

Trigrams with Word Boundaries

1. *sch, che, ung, ten, ich, gen*
2. *ing, ver, nde, gen, oor, ijn*
3. *ing, ion, tio, ent, ati, ter*
4. *ova, val, ali, pre, ala, nie*
5. *ova, ých, ost, ová, ick, ého*
6. *sze, meg, nak, ban, szt, nek*
7. *nie, rze, owa, dzi, prz, rzy*
8. *ent, ion, nte, ato, zio, con*
9. *ent, ado, nte, con, ica, ada*
10. *ent, ion, ant, tio, que, ati*
11. *ова, ост, енн, ого, льн, про*

- *en], sch, er], che, ung, ten*
- *en], ing, ver, er], nde, de]*
- *ing, ed], ng], es], ion, er]*

Trigrams with Word Boundaries

1. *sch, che, ung, ten, ich, gen*
2. *ing, ver, nde, gen, oor, ijn*
3. *ing, ion, tio, ent, ati, ter*
4. *ova, val, ali, pre, ala, nie*
5. *ova, ých, ost, ová, ick, ého*
6. *sze, meg, nak, ban, szt, nek*
7. *nie, rze, owa, dzi, prz, rzy*
8. *ent, ion, nte, ato, zio, con*
9. *ent, ado, nte, con, ica, ada*
10. *ent, ion, ant, tio, que, ati*
11. *ова, ост, енн, ого, льн, про*

- *en], sch, er], che, ung, ten*
- *en], ing, ver, er], nde, de]*
- *ing, ed], ng], es], ion, er]*
- *[pr, la], [po, ova, [ne, ch]*

Trigrams with Word Boundaries

1. *sch, che, ung, ten, ich, gen*
2. *ing, ver, nde, gen, oor, ijn*
3. *ing, ion, tio, ent, ati, ter*
4. *ova, val, ali, pre, ala, nie*
5. *ova, ých, ost, ová, ick, ého*
6. *sze, meg, nak, ban, szt, nek*
7. *nie, rze, owa, dzi, prz, rzy*
8. *ent, ion, nte, ato, zio, con*
9. *ent, ado, nte, con, ica, ada*
10. *ent, ion, ant, tio, que, ati*
11. *ова, ост, енн, ого, льн, про*

- *en], sch, er], che, ung, ten*
- *en], ing, ver, er], nde, de]*
- *ing, ed], ng], es], ion, er]*
- *[pr, la], [po, ova, [ne, ch]*
- *ch], [ne, ova, ní], [po, ou]*

Trigrams with Word Boundaries

1. *sch, che, ung, ten, ich, gen*
2. *ing, ver, nde, gen, oor, ijn*
3. *ing, ion, tio, ent, ati, ter*
4. *ova, val, ali, pre, ala, nie*
5. *ova, ých, ost, ová, ick, ého*
6. *sze, meg, nak, ban, szt, nek*
7. *nie, rze, owa, dzi, prz, rzy*
8. *ent, ion, nte, ato, zio, con*
9. *ent, ado, nte, con, ica, ada*
10. *ent, ion, ant, tio, que, ati*
11. *ова, ост, енн, ого, льн, про*

- *en], sch, er], che, ung, ten*
- *en], ing, ver, er], nde, de]*
- *ing, ed], ng], es], ion, er]*
- *[pr, la], [po, ova, [ne, ch]*
- *ch], [ne, ova, ní], [po, ou]*
- *[sz, ek], ak], en], sze, an]*

Trigrams with Word Boundaries

1. *sch, che, ung, ten, ich, gen*
2. *ing, ver, nde, gen, oor, ijn*
3. *ing, ion, tio, ent, ati, ter*
4. *ova, val, ali, pre, ala, nie*
5. *ova, ých, ost, ová, ick, ého*
6. *sze, meg, nak, ban, szt, nek*
7. *nie, rze, owa, dzi, prz, rzy*
8. *ent, ion, nte, ato, zio, con*
9. *ent, ado, nte, con, ica, ada*
10. *ent, ion, ant, tio, que, ati*
11. *ова, ост, енн, ого, льн, про*

- *en], sch, er], che, ung, ten*
- *en], ing, ver, er], nde, de]*
- *ing, ed], ng], es], ion, er]*
- *[pr, la], [po, ova, [ne, ch]*
- *ch], [ne, ova, ní], [po, ou]*
- *[sz, ek], ak], en], sze, an]*
- *ie], nie, [po, [pr, rze, owa*

Trigrams with Word Boundaries

1. *sch, che, ung, ten, ich, gen*
2. *ing, ver, nde, gen, oor, ijn*
3. *ing, ion, tio, ent, ati, ter*
4. *ova, val, ali, pre, ala, nie*
5. *ova, ých, ost, ová, ick, ého*
6. *sze, meg, nak, ban, szt, nek*
7. *nie, rze, owa, dzi, prz, rzy*
8. *ent, ion, nte, ato, zio, con*
9. *ent, ado, nte, con, ica, ada*
10. *ent, ion, ant, tio, que, ati*
11. *ова, ост, енн, ого, льн, про*

- *en], sch, er], che, ung, ten*
- *en], ing, ver, er], nde, de]*
- *ing, ed], ng], es], ion, er]*
- *[pr, la], [po, ova, [ne, ch]*
- *ch], [ne, ova, ní], [po, ou]*
- *[sz, ek], ak], en], sze, an]*
- *ie], nie, [po, [pr, rze, owa*
- *to], ent, re], te], ti], [co*

Trigrams with Word Boundaries

1. *sch, che, ung, ten, ich, gen*
2. *ing, ver, nde, gen, oor, ijn*
3. *ing, ion, tio, ent, ati, ter*
4. *ova, val, ali, pre, ala, nie*
5. *ova, ých, ost, ová, ick, ého*
6. *sze, meg, nak, ban, szt, nek*
7. *nie, rze, owa, dzi, prz, rzy*
8. *ent, ion, nte, ato, zio, con*
9. *ent, ado, nte, con, ica, ada*
10. *ent, ion, ant, tio, que, ati*
11. *ова, ост, енн, ого, льн, про*

- *en], sch, er], che, ung, ten*
- *en], ing, ver, er], nde, de]*
- *ing, ed], ng], es], ion, er]*
- *[pr, la], [po, ova, [ne, ch]*
- *ch], [ne, ova, ní], [po, ou]*
- *[sz, ek], ak], en], sze, an]*
- *ie], nie, [po, [pr, rze, owa*
- *to], ent, re], te], ti], [co*
- *os], as], es], ent, do], ado*

Trigrams with Word Boundaries

1. *sch, che, ung, ten, ich, gen*
2. *ing, ver, nde, gen, oor, ijn*
3. *ing, ion, tio, ent, ati, ter*
4. *ova, val, ali, pre, ala, nie*
5. *ova, ých, ost, ová, ick, ého*
6. *sze, meg, nak, ban, szt, nek*
7. *nie, rze, owa, dzi, prz, rzy*
8. *ent, ion, nte, ato, zio, con*
9. *ent, ado, nte, con, ica, ada*
10. *ent, ion, ant, tio, que, ati*
11. *ова, ост, енн, ого, льн, про*

- *en], sch, er], che, ung, ten*
- *en], ing, ver, er], nde, de]*
- *ing, ed], ng], es], ion, er]*
- *[pr, la], [po, ova, [ne, ch]*
- *ch], [ne, ova, ní], [po, ou]*
- *[sz, ek], ak], en], sze, an]*
- *ie], nie, [po, [pr, rze, owa*
- *to], ent, re], te], ti], [co*
- *os], as], es], ent, do], ado*
- *es], ent, nt], er], ion, on]*

Trigrams with Word Boundaries

1. *sch, che, ung, ten, ich, gen*
2. *ing, ver, nde, gen, oor, ijn*
3. *ing, ion, tio, ent, ati, ter*
4. *ova, val, ali, pre, ala, nie*
5. *ova, ých, ost, ová, ick, ého*
6. *sze, meg, nak, ban, szt, nek*
7. *nie, rze, owa, dzi, prz, rzy*
8. *ent, ion, nte, ato, zio, con*
9. *ent, ado, nte, con, ica, ada*
10. *ent, ion, ant, tio, que, ati*
11. *ова, ост, енн, ого, льн, про*

- *en], sch, er], che, ung, ten*
- *en], ing, ver, er], nde, de]*
- *ing, ed], ng], es], ion, er]*
- *[pr, la], [po, ova, [ne, ch]*
- *ch], [ne, ova, ní], [po, ou]*
- *[sz, ek], ak], en], sze, an]*
- *ie], nie, [po, [pr, rze, owa*
- *to], ent, re], te], ti], [co*
- *os], as], es], ent, do], ado*
- *es], ent, nt], er], ion, on]*
- *[пр, [по, ся], ой], ова, ост*

Impact of Word Frequency

- Frequent words $\Rightarrow$ frequent trigrams
- Word frequency is another language feature

- Sparse data problem:
  - Frequent words characteristic of the text topic (rather than of the language)
  - UDHR is small, thus a problem
  - It hurts less with large training data

Example

- Universal Declaration of Human Rights
  - *právo, povinnost, poslání* ("right, responsibility, mission")
  - cs: *prá*, *ráv*, *ost*, *lán*, *nos*, *ání*
  - sk: *prá*, *ráv*, *ost*, *ani*, *nos*, *kto*
  - pl: *nie*, *pra*, *ani*, *raw*, *nia*, *wie*
  - ru: *рав, ств, пра, ени, ать, ове*
  - hr: *rav*, *pra*, *ima*, *nje*, *anj*, *vat*

Example

- Universal Declaration of Human Rights
  - *právo, povinnost, poslání* ("right, responsibility, mission")
  - cs: *prá*, *ráv*, *ost*, *lán*, *nos*, *ání*
  - sk: *prá*, *ráv*, *ost*, *ani*, *nos*, *kto*
  - pl: *nie*, *pra*, *ani*, *raw*, *nia*, *wie*
  - ru: *рав, ств, пра, ени, ать, ове*
  - hr: *rav*, *pra*, *ima*, *nje*, *anj*, *vat*
- Centrum.cz (a Czech web portal)
  - There is a menu with districts *Praha 1* ... *Praha 15*
  - $\Rightarrow$ top ten trigrams contain *Pra*, *rah*, *aha*!

Example

- Universal Declaration of Human Rights
  - *právo, povinnost, poslání* ("right, responsibility, mission")
  - cs: *prá*, *ráv*, *ost*, *lán*, *nos*, *ání*
  - sk: *prá*, *ráv*, *ost*, *ani*, *nos*, *kto*
  - pl: *nie*, *pra*, *ani*, *raw*, *nia*, *wie*
  - ru: *рав, ств, пра, ени, ать, ове*
  - hr: *rav*, *pra*, *ima*, *nje*, *anj*, *vat*
- Centrum.cz (a Czech web portal)
  - There is a menu with districts *Praha 1 ... Praha 15*
  - $\Rightarrow$ top ten trigrams contain *Pra*, *rah*, *aha*!
- Prague Dependency Treebank
  - *pro*, *ost*, *ých*, *ova*, *sta*, *ení*, *ter*, *pře*, *ého*, *kte*, *řed*, *sti*, *pod*, *ích*, *ick*, *nos*, *kon*, *ské*, *ist*, *ent*...

Word Frequency

- Prague Dependency Treebank (over 1,000,000 words)
  - *a, v, se, na, je, že, o, s, z, by, i, do, to, k, ve...*
  - = and, in, itself, on, is, that, about, with, from, would, and, to, it, to, in

Word Frequency

- Prague Dependency Treebank (over 1,000,000 words)
  - *a, v, se, na, je, že, o, s, z, by, i, do, to, k, ve...*
  - = and, in, itself, on, is, that, about, with, from, would, and, to, it, to, in
- Universal Declaration of Human Rights (Czech)
  - *a, právo, na, nebo, má, Článek, Každý, v...*
  - = and, right, on, or, has, Article, Every, in

Word Frequency

- Prague Dependency Treebank (over 1,000,000 words)
  - *a, v, se, na, je, že, o, s, z, by, i, do, to, k, ve...*
  - = and, in, itself, on, is, that, about, with, from, would, and, to, it, to, in
- Universal Declaration of Human Rights (Czech)
  - *a, právo, na, nebo, má, Článek, Každý, v...*
  - = and, right, on, or, has, Article, Every, in
- http://www.centrum.cz/
  - *Praha, čeština, a, nad, do, 1, Hledej, Kč, byty...*
  - = Prague, Czech, and, over, to, 1, Search, CZK, apartments

Neutralize Word Frequency

- Count every word type only once
- Prague Dependency Treebank (over 1,000,000 words)
  - before: *pro*, *ost*, *ých*, *ova*, *sta*, *ení*, *ter*, *pře*, *ého*
  - after: *ova*, *ých*, *ost*, *ová*, *ick*, *ého*, *pro*, *val*, *kov*

Neutralize Word Frequency

- Count every word type only once
- Prague Dependency Treebank (over 1,000,000 words)
  - before: *pro*, *ost*, *ých*, *ova*, *sta*, *ení*, *ter*, *pře*, *ého*
  - after: *ova*, *ých*, *ost*, *ová*, *ick*, *ého*, *pro*, *val*, *kov*
- Universal Declaration of Human Rights (Czech)
  - before: *prá*, *ráv*, *ost*, *lán*, *nos*, *ání*, *neb*, *ávo*, *ažd*
  - after: *ost*, *ání*, *nos*, *ení*, *ého*, *ých*, *ván*, *ová*, *roz*

Neutralize Word Frequency

- Count every word type only once
- Prague Dependency Treebank (over 1,000,000 words)
  - before: *pro*, *ost*, *ých*, *ova*, *sta*, *ení*, *ter*, *pře*, *ého*
  - after: *ova*, *ých*, *ost*, *ová*, *ick*, *ého*, *pro*, *val*, *kov*
- Universal Declaration of Human Rights (Czech)
  - before: *prá*, *ráv*, *ost*, *lán*, *nos*, *ání*, *neb*, *ávo*, *ažd*
  - after: *ost*, *ání*, *nos*, *ení*, *ého*, *ých*, *ván*, *ová*, *roz*
- http://www.centrum.cz/
  - before: *tin*, *šti*, *Pra*, *aha*, *rah*, *sko*, *ost*, *ešt*, *češ*
  - after: *sko*, *ina*, *ost*, *lov*, *ský*, *str*, *ava*, *cho*, *rav*

Character Encoding

- Training and test data must use same encoding (ideally UTF8)
- Do we know the encoding? $\Rightarrow$ convert!
  - Unix command `file` can sometimes guess encoding

Character Encoding

- Training and test data must use same encoding (ideally UTF8)
- Do we know the encoding? $\Rightarrow$ convert!
  - Unix command `file` can sometimes guess encoding
- Not? $\Rightarrow$ Every encoding is a language!

Character Encoding

- Training and test data must use same encoding (ideally UTF8)
- Do we know the encoding? ⇒ convert!
  - Unix command `file` can sometimes guess encoding
- Not? ⇒ Every encoding is a language!

- Prague Dependency Treebank in different encodings (viewed through `cpl1250`)
  - `cpl1250` … *ova, ých, ost, ová, ick, ého, pro*
  - `cpl852` … *ova, ěch, ost, ov , ick, ,ho, pro*
  - `iso-8859-2` … *ova, ých, ost, ová, ick, ého, pro*
  - `utf-8` … *nĂ-, ovĂ, ova, Ĺ™e, vĂ ˇ, nĂ©*
  - `ascii` … *ova, ych, van, ost, ick, ove, eho*

Unicode

- Originally (1991) sixteen-bit code to replace various 8bit codepages
  - ASCII (128 chars) $\rightarrow$ 8bit codepages (256 chars) $\rightarrow$ Unicode (65,536 chars)
  - Fixed-with encoding (2 bytes) = UTF-16 *Unicode Transformation Format*

Unicode

- Originally (1991) sixteen-bit code to replace various 8bit codepages
  - ASCII (128 chars) $\rightarrow$ 8bit codepages (256 chars) $\rightarrow$ Unicode (65,536 chars)
  - Fixed-with encoding (2 bytes) = UTF-16 *Unicode Transformation Format*
- Since version 3.2 (2002) needs 32 bits
  - Actually used only 0 to 10FFFF (over 1 million chars)
  - UTF-32: always 4 bytes (although one is empty)
  - Examples of chars above FFFF:
    - 1D000–1D0F5 BYZANTINE MUSICAL SYMBOL PSILI – BYZANTINE MUSICAL SYMBOL GORGON NEO KATO
    - 1F030–1F093 DOMINO TILE HORIZONTAL BACK – DOMINO TILE VERTICAL-06-06

Unicode

- Originally (1991) sixteen-bit code to replace various 8bit codepages
  - ASCII (128 chars) $\rightarrow$ 8bit codepages (256 chars) $\rightarrow$ Unicode (65,536 chars)
  - Fixed-with encoding (2 bytes) = UTF-16 *Unicode Transformation Format*
- Since version 3.2 (2002) needs 32 bits
  - Actually used only 0 to 10FFFF (over 1 million chars)
  - UTF-32: always 4 bytes (although one is empty)
  - Examples of chars above FFFF:
    - 1D000–1D0F5 BYZANTINE MUSICAL SYMBOL PSILI – BYZANTINE MUSICAL SYMBOL GORGON NEO KATO
    - 1F030–1F093 DOMINO TILE HORIZONTAL BACK – DOMINO TILE VERTICAL-06-06
- Currently (2017) Unicode 10.0: 136,690 chars from 139 languages/scripts

UTF-8

- UTF-8 is a way of encoding Unicode
- Variable character width

- If possible, make sure that all your data is UTF-8 and all your software assumes UTF-8 everywhere!
  - (A bit fight in Windows command line, but can be enforced even there.)

UTF-8

- UTF-8 is a way of encoding Unicode
- Variable character width
- Frequent characters: 1 byte
  - ASCII: English letters, numbers, punctuation

- If possible, make sure that all your data is UTF-8 and all your software assumes UTF-8 everywhere!
  - (A bit fight in Windows command line, but can be enforced even there.)

UTF-8

- UTF-8 is a way of encoding Unicode
- Variable character width
- Frequent characters: 1 byte
  - ASCII: English letters, numbers, punctuation
- Less frequent: 2 bytes
  - Accented Latin characters (not all but most)

- If possible, make sure that all your data is UTF-8 and all your software assumes UTF-8 everywhere!
  - (A bit fight in Windows command line, but can be enforced even there.)

UTF-8

- UTF-8 is a way of encoding Unicode
- Variable character width
- Frequent characters: 1 byte
  - ASCII: English letters, numbers, punctuation
- Less frequent: 2 bytes
  - Accented Latin characters (not all but most)
- Even less frequent: 3 or 4 bytes
  - Chinese characters

- If possible, make sure that all your data is UTF-8 and all your software assumes UTF-8 everywhere!
  - (A bit fight in Windows command line, but can be enforced even there.)

UTF-8

- 0 – 127: one byte
  - 8th bit empty $\Rightarrow$ no more bytes

UTF-8

- 0 – 127: one byte
  - 8th bit empty $\Rightarrow$ no more bytes
- 128 – 2048: two bytes
- 2048 – 65535: three bytes
  - First byte: N highest bits set $\Rightarrow$ N bytes total
  - Actual code starts after first zero
  - starts with 0 ... 1 byte
  - starts with 110 ... 2 bytes
  - starts with 1110 ... 3 bytes
  - starts with 10 ... non-first byte!

UTF-8

- 0 – 127: one byte
  - 8th bit empty $\Rightarrow$ no more bytes
- 128 – 2048: two bytes
- 2048 – 65535: three bytes
  - First byte: N highest bits set $\Rightarrow$ N bytes total
  - Actual code starts after first zero
  - starts with 0 ... 1 byte
  - starts with 110 ... 2 bytes
  - starts with 1110 ... 3 bytes
  - starts with 10 ... non-first byte!
- We can recognize the first byte
- Not every byte sequence is valid UTF-8!

UTF-8 Example

- "Č" is character 268 (hex 010C, bin 1 0000 1100)

UTF-8 Example

- "Č" is character 268 (hex 010C, bin 1 0000 1100)
- $127 < 268 < 2048 \Rightarrow$ need 2 bytes
  - First starts 110 and has 5 content bits
  - Second starts 10 and has 6 content bits

UTF-8 Example

- "Č" is character 268 (hex 010C, bin 1 0000 1100)
- $127 < 268 < 2048 \Rightarrow$ need 2 bytes
  - First starts 110 and has 5 content bits
  - Second starts 10 and has 6 content bits
  - We need only 9 content bits
  - $\Rightarrow$ pad with two zeroes from the left
- Resulting code:
  - 110 00100 10 001100 = hex C4 8C

Unicode Character Types

- Letters
  - A a B b C c Щ щ Ω ω ט ض ண 読

Unicode Character Types

- Letters
  - A a B b C c Щ щ Ω ω ל ض ண 読
- Mark
  - q̣ ◌̣

Unicode Character Types

- Letters
  - A a B b C c Щ щ Ω ω ל ض ண் 読
- Mark
  - q̐ ̐
- Number
  - 1 2 3 ١ ٢ ٣ ൧ ൨ ൩

Unicode Character Types

- Letters
  - A a B b C c Щ щ Ω ω ל ض ண் 読
- Mark
  - q̐ ँ
- Number
  - 1 2 3 ١ ٢ ٣ ൧ ൨ ൩
- Punctuation
  - , ; : . ! ? ¡ ¿ ، ؟ ॥

Unicode Character Types

- Letters
  - A a B b C c Щ щ Ω ω ל ض ண 読
- Mark
  - q̣ ̆
- Number
  - 1 2 3 ١ ٢ ٣ ↻ ৩ ვ
- Punctuation
  - , ; : . ! ? ¡ ¿ ، ؟ ॥
- Symbol
  - $ ¥ € ± ⇎ ∃ ☏ ☝ ☯

Unicode Character Types

- Letters
  - A a B b C c Щ щ Ω ω ל ض ண 読
- Mark
  - q̐ ँ
- Number
  - 1 2 3 ١ ٢ ٣ ൧ ൨ ൩
- Punctuation
  - , ; : . ! ? ¡ ¿ ، ؟ ॥
- Symbol
  - $ ¥ € ± ⇎ ∃ ☏ ☝ ☯
- Separator
  - spaces of various sizes

Unicode Character Types

- Letters
  - A a B b C c Щ щ Ω ω ל ض ഞ 読
- Mark
  - q̊ ̊
- Number
  - 1 2 3 ١ ٢ ٣ ↻ ৩ ვ
- Punctuation
  - , ; : . ! ? ¡ ¿ ، ؟ ॥
- Symbol
  - $ ¥ € ± ⇎ ∃ ☏ ☝ ☯
- Separator
  - spaces of various sizes
- Other
  - ZERO WIDTH NON-JOINER
    felestiniha “pencils” ها فلسطيني vs. فلسطينيها

Unicode in Regular Expressions

- Perl (see `https://perldoc.perl.org/perlunicode.html`)
  - `$text =~ m/^[\p{Ll}\p{Lm}\p{Lo}\p{M}]+$/;`
  - `\p{Cyrillic}`
- Python
  - Not available by default (?)
  - `import regex as re`
- Perl: Unicode character names
  - `use charnames ();`
  - `print charnames::viacode(ord($char));`
    - LATIN SMALL LETTER N PRECEDED BY APOSTROPHE
- Python: `import unicodedata`
  - `unicodedata.name('č')`
  - `unicodedata.normalize('NFC', 'č')`

What You See May Not Be What You Have!

- Россия / Rossija

What You See May Not Be What You Have!

- Россия / Rossija
  - CYRILLIC CAPITAL LETTER ER
  - LATIN SMALL LETTER O
  - LATIN SMALL LETTER C
  - CYRILLIC SMALL LETTER ES
  - CYRILLIC SMALL LETTER I
  - CYRILLIC SMALL LETTER YA

What You See May Not Be What You Have!

- Россия / Rossija
  - CYRILLIC CAPITAL LETTER ER
  - LATIN SMALL LETTER O
  - LATIN SMALL LETTER C
  - CYRILLIC SMALL LETTER ES
  - CYRILLIC SMALL LETTER I
  - CYRILLIC SMALL LETTER YA
- I don't speak Taa–!Ui!

What You See May Not Be What You Have!

- Россия / Rossija
  - CYRILLIC CAPITAL LETTER ER
  - LATIN SMALL LETTER O
  - LATIN SMALL LETTER C
  - CYRILLIC SMALL LETTER ES
  - CYRILLIC SMALL LETTER I
  - CYRILLIC SMALL LETTER YA

- I don't speak Taa–!Ui!
  - ! 451 01C3 L LATIN LETTER RETROFLEX CLICK
  - U 85 0055 L LATIN CAPITAL LETTER U
  - i 105 0069 L LATIN SMALL LETTER I
  - ! 33 0021 P EXCLAMATION MARK

Unicode Normalization

Figure 1. Examples of Canonical Equivalence

Subtype	Examples
Combining sequence	Ç ⟷ C+◌̧
Ordering of combining marks	q+◌̇+◌̣ ⟷ q+◌̣+◌̇
Hangul & conjoining jamo	가 ⟷ ㄱ + ㅏ
Singleton equivalence	Ω ⟷ Ω

Unicode Normalization

Figure 2. Examples of Compatibility Equivalence

Subtype	Examples
Font variants	ℌ → H
	ℍ → H
Linebreaking differences	[NBSP] → [SPACE]
Positional variant forms	ε → ε
	ϵ → ε
	⋹ → ε
	𝛆 → ε
Circled variants	① → 1
Width variants	ｶ → 力
Rotated variants	ɾ

Unicode Normalization Forms

- NFD: Canonical Decomposition
- NFC: Canonical Decomposition, followed by Canonical Composition

- NFKD: Compatibility Decomposition
- NFKC: Compatibility Decomposition, followed by Canonical Composition

- `http://unicode.org/reports/tr15/`

Unicode Normalization in Perl and Python

- Perl

```
use Unicode::Normalize;
$normalized = NFC($text);
```

- Python

```
import unicodedata
normalized = unicodedata.normalize('NFC', text)
```

- Warning: Different Perl/Python version ⇒ different version of the Unicode database ⇒ slightly different results!