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

What if no dictionary is available?
- Hint: this doesn't seem to be a European language: ירושלים של זהב
- Specific letters within Latin: Ř, Ø
- Czech?
- Danish / Norwegian?
- But most letters are shared among multiple languages
- And this is German! Die Burg wurde vom böhmischen König Přemysl erobert.
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

- But most letters are shared among multiple languages
- And this is German!
  Die Burg wurde vom böhmischen König Přemysl erobert.
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: ירושלם של זהב

- Specific letters within Latin
  - Ř ➔ Czech?
  - Ø ➔ 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
  - Ř ⇒ Czech?
  - Ø ⇒ 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 Ľ Ř W
- Russian ... O A E И Н ... Ц Щ Ф Э Ь
- English ... E I S N T ... K X Q J Z
- German ... E N T S R ... Ô 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
<table>
<thead>
<tr>
<th>N-gram</th>
<th>Languages</th>
</tr>
</thead>
<tbody>
<tr>
<td>che, sch, der, ich, ter, ung</td>
<td>de (German)</td>
</tr>
<tr>
<td>ver, gen, ing, ste, aar, ijn</td>
<td>nl (Dutch)</td>
</tr>
<tr>
<td>ing, ion, tio, ent, ati, ers</td>
<td>en (English)</td>
</tr>
<tr>
<td>pre, ova, nov, vol, nie, ový</td>
<td>sk (Slovak)</td>
</tr>
<tr>
<td>ova, ých, ost, ová, ick, ého</td>
<td>cs (Czech)</td>
</tr>
<tr>
<td>szt, tás, asz, ban, ala, sza</td>
<td>hu (Hungarian)</td>
</tr>
<tr>
<td>nie, dzi, ego, kie, rze, ych</td>
<td>pl (Polish)</td>
</tr>
<tr>
<td>ent, tat, ato, est, ion, zio</td>
<td>it (Italian)</td>
</tr>
<tr>
<td>ado, nte, dad, ent, art, cto</td>
<td>es (Spanish)</td>
</tr>
<tr>
<td>ion, ent, tio, ale, eme, les</td>
<td>fr (French)</td>
</tr>
<tr>
<td>али, вал, ост, про, при</td>
<td>ru (Russian)</td>
</tr>
<tr>
<td>ent, ute, acu, cut, aci</td>
<td>(es) á</td>
</tr>
</tbody>
</table>
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, лън, 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, nî], [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, dzí, 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, ьлын, проб

- 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, nî], [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]
<table>
<thead>
<tr>
<th></th>
<th>sch, che, ung, ten, ich, gen</th>
<th></th>
<th>en], sch, er], che, ung, ten</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>ing, ver, nde, gen, oor, ijn</td>
<td></td>
<td>en], ing, ver, er], nde, de]</td>
</tr>
<tr>
<td>3</td>
<td>ing, ion, tio, ent, ati, ter</td>
<td></td>
<td>ing, ed], ng], es], ion, er]</td>
</tr>
<tr>
<td>4</td>
<td>ova, val, ali, pre, ala, nie</td>
<td></td>
<td>[pr, la], [po, ova, [ne, ch]</td>
</tr>
<tr>
<td>5</td>
<td>ova, ých, ost, ová, ick, ého</td>
<td></td>
<td>ch], [ne, ova, ní], [po, ou]</td>
</tr>
<tr>
<td>6</td>
<td>sze, meg, nak, ban, szt, nek</td>
<td></td>
<td>[sz, ek], ak], en], sze, an]</td>
</tr>
<tr>
<td>7</td>
<td>nie, rze, owa, dzi, prz, rzy</td>
<td></td>
<td>ie], nie, [po, [pr, rze, owa</td>
</tr>
<tr>
<td>8</td>
<td>ent, ion, nte, ato, zio, con</td>
<td></td>
<td>to], ent, re], te], ti], [co</td>
</tr>
<tr>
<td>9</td>
<td>ent, ado, nte, con, ica, ada</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>ent, ion, ant, tio, que, ati</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>ова, ост, енн, ого, льн, про</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
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. ova, 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

- 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 | ova, 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
- **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*
  - ⇒ 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*
  - ⇒ 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
• 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

http://www.centrum.cz/
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
Training and test data must use same encoding (ideally UTF8)
Do we know the encoding? \(\Rightarrow\) convert!
- Unix command `file` can sometimes guess 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!
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 cp11250)

- `cp11250` ... `ova, ých, ost, ová, i ck, ého, pro`
- `cp1852` ... `ova, ěch, ost, ov , i ck, ,ho, pro`
- `iso-8859-2` ... `ova, ých, ost, ová, i ck, ého, pro`
- `utf-8` ... `nĂ-, ovĂ, ova, ŽMe, vĂ˘, nĂ©`
- `ascii` ... `ova, ych, van, ost, i ck, ove, eho`
Originally (1991) sixteen-bit code to replace various 8bit codepages

- ASCII (128 chars) → 8bit codepages (256 chars) → Unicode (65,536 chars)
- Fixed-with encoding (2 bytes) = UTF-16 \textit{Unicode Transformation Format}
Originally (1991) sixteen-bit code to replace various 8bit codepages
- ASCII (128 chars) → 8bit codepages (256 chars) → 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) → 8bit codepages (256 chars) → 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 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 ⇒ no more bytes

- 128 – 2048: two bytes
  - First byte: N highest bits set
  - N bytes total
  - Actual code starts after first zero

- 2048 – 65535: three bytes
  - Starts with 110 ...
  - Starts with 1110 ...
  - Starts with 10 ...
  - Non-first byte!

We can recognize the first byte

Not every byte sequence is valid UTF-8!
UTF-8

- 0 – 127: one byte
  - 8th bit empty ⇒ no more bytes
- 128 – 2048: two bytes
- 2048 – 65535: three bytes
  - First byte: N highest bits set ⇒ 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 ⇒ no more bytes
- 128 – 2048: two bytes
- 2048 – 65535: three bytes
  - First byte: $N$ highest bits set ⇒ $N$ 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)
“Č” is character 268 (hex 010C, bin 1 0000 1100)

127 < 268 < 2048 ⇒ 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
  - pad with two zeroes from the left
  - Resulting code: 110 00100 10 001100 = hex C4 8C
UTF-8 Example

- “Č” is character 268 (hex 010C, bin 1 0000 1100)
- 127 < 268 < 2048 ⇒ 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
  - ⇒ 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 ض ض ヒヒ 読

Alphabets, Encoding, Language Recognition
Unicode Character Types

- Letters
  - A a B b C c Щ щ Ω ω ﺿ ﺝ
- Mark
  - ﺞ ﺞ

Alphabets, Encoding, Language Recognition
Unicode Character Types

- Letters
  - A a B b C c Щ щ ω ω ض ض သ သ た
- Mark
  - ꞗ ꞗ
- Number
  - ٣ ٢ ١ ٢ ٣ ٢ ١ ٢ ٣
- Punctuation
- Symbol
- Separator
- Other
Unicode Character Types

- **Letters**
  - A a B b C c Щ щ Ω ω ﭨ ﭨ ﭧ ﭩ
- **Mark**
  - q
- **Number**
  - ٣٢١۰٣٢٢٣
- **Punctuation**
  - , ; : . ! ؟ ؟ ؍
Unicode Character Types

- **Letters**
  - A a B b C c Щ щ Ω ω ض ض اما

- **Mark**
  - ّ ّ ّ ّ ّ ّ

- **Number**
  - ٣ ₂ ١ ٢ ٣ ٢ ١ ٢ ٣

- **Punctuation**
  - , ; : . ! ؟ ؟ ؟ ؟

- **Symbol**
  - $ ¥ € ± ⇎ ∃ ☏ ☝ ☯
Unicode Character Types

- Letters
  - A a B b C c ¶ щ Ω ω ﭻ 読
- Mark
  - ﺑ ﺓ ﺔ ﺖ
- Number
  - ١ ٢ ٣ ﻒ ﺗ ﻒ ﻒ
- Punctuation
  - ، ; : . ! ؟ ؛ ؟ ﻻ
- Symbol
  - $ ¥ € ± ⇎ ∃ ☏ ☝ ☯
- Separator
  - spaces of various sizes
Unicode Character Types

- Letters
  - A a B b C c Щ щ ω ω ض ﻭ ﺭ ﻧ ﻣ
- Mark
  - ٍّ
- Number
  - ١ ٢ ٣ ٤ ٥ ٦ ٧ ٨ ٩ ٠ ﻤ ﻣ ﻚ ﻝ ﻨ ﻚ ﻝ ﻨ ﻚ ﻝ ﻨ ﻚ ﻝ ﻨ ﻚ ﻝ ﻨ ﻚ ﻝ ﻨ ﻚ ﻝ ﻨ ﻚ ﻝ ﻨ ﻚ ﻝ ﻨ ﻚ ﻝ ﻨ ﻚ ﻝ ﻨ ﻚ ﻝ ﻨ ﻚ ﻝ ﻨ ﻚ ﻝ ﻨ ﻚ ﻝ ﻨ ﻚ ﻝ ﻨ ﻚ ﻝ ﻨ ﻚ ﻝ ﻨ ﻚ ﻝ ﻨ ﻚ ﻝ ﻨ ﻚ ﻝ ﻨ ﻚ ﻝ ﻨ ﻚ ﻝ ﻨ ﻚ ﻝ ﻨ ﻚ ﻝ ﻨ ﻚ ﻝ ﻨ ﻚ ﻝ ﻨ ﻚ ﻝ ﻨ ﻚ ﻝ ﻨ ﻚ ﻝ ﻨ ﻚ ﻝ ﻨ ﻚ ﻝ ﻨ ﻚ ﻝ ﻨ ﻚ ﻝ ﻨ ﻚ ﻝ ﻨ ﻚ ﻝ ﻨ ﻚ ﻝ ﻨ ﻚ ﻝ ﻨ ﻚ ﻝ ﻨ ﻚ ﻝ ﻨ ﻚ ﻝ ﻨ ﻚ ﻝ ﻨ ﻚ ﻝ ﻨ ﻚ ﻝ ﻨ ﻚ ﻝ ﻨ ﻚ ﻝ ﻨ ﻚ ﻝ ﻨ ﻚ ﻝ ﻨ ﻚ ﻝ ﻨ ﻚ ﻝ ﻨ ﻚ ﻝ ﻨ ﻚ ﻝ ﻨ ﻚ ﻝ ﻨ ﻚ ﻝ ﻨ ﻚ ﻝ ﻨ ﻚ ﻝ ﻨ ﻚ ﻝ ﻨ ﻚ ﻝ ﻨ ﻚ ﻝ ﻨ ﻚ ﻝ ﻨ ﻚ ﻝ ﻨ ﻚ ﻝ 
- Punctuation
  - , ; : . ! ? ¡ ¿ ؟ ﻪ ﻪ ﻪ ﻪ ﻪ ﻪ 
- Symbol
  - $ ¥ € ± ⇎ ∃ ☏ ☝ ☯
- Separator
  - spaces of various sizes
- Other
  - ZERO WIDTH NON-JOINER
  - felestiniha “pencils” فلسطينیها vs. فلسطينیها
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!

- Rossiya / 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–ǃUi!
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

<table>
<thead>
<tr>
<th>Subtype</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Combining sequence</td>
<td>ζ ←→ C + ¨3</td>
</tr>
<tr>
<td>Ordering of combining marks</td>
<td>q + ¨ + ¨ ←→ q + ¨ + ¨</td>
</tr>
<tr>
<td>Hangul &amp; conjoining jamo</td>
<td>가 ←→ ㄱ + ㅏ</td>
</tr>
<tr>
<td>Singleton equivalence</td>
<td>Ω ←→ Ω</td>
</tr>
</tbody>
</table>
### Unicode Normalization

**Figure 2. Examples of Compatibility Equivalence**

<table>
<thead>
<tr>
<th>Subtype</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Font variants</td>
<td>ולה → ה</td>
</tr>
<tr>
<td></td>
<td>חח → ח</td>
</tr>
<tr>
<td>Linebreaking differences</td>
<td>[NBSP] → [SPACE]</td>
</tr>
<tr>
<td>Positional variant forms</td>
<td>ε → ε</td>
</tr>
<tr>
<td></td>
<td>ξ → ξ</td>
</tr>
<tr>
<td></td>
<td>ω → ω</td>
</tr>
<tr>
<td>Circed variants</td>
<td>① → 1</td>
</tr>
<tr>
<td>Width variants</td>
<td>カ → カ</td>
</tr>
<tr>
<td>Rotated variants</td>
<td></td>
</tr>
</tbody>
</table>
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/](http://unicode.org/reports/tr15/)