Tokenization and Word Segmentation

Daniel Zeman, Rudolf Rosa

March 4, 2019
**Tokenization and Word Segmentation**

- **IMPORTANT** because:
  - Training tokenization $\neq$ test tokenization
  - $\Rightarrow$ accuracy goes down

- Not always trivial
- May interact with morphology
- May include normalization (character-level)
“María, I love you!” Juan exclaimed.

«¡María, te amo!», exclamó Juan.

X PRON X VERB X

« ¡María , te amo ! » ,

PUNCT PUNCT PROPN PUNCT PRON VERB PUNCT PUNCT

- Classic tokenization:
  - Separate punctuation from words
  - Recognize certain clusters of symbols like “...”
  - Perhaps keep together things like user@mail.x.edu
Using Unicode Character Categories

- https://perldoc.perl.org/perlunicode.html
  - $text =~ s/(\pP)/ $1 /g;
  - $text =~ s/^\s+//;
  - $text =~ s/\s+$//;
  - @tokens = split(\s+/, $text);
  - Optionally recombine email addresses, URLs etc.

Some problems
- haven’t (English; should be haven’t)
- instal·lació (Catalan; should be 1 token)
- single quote (punctuation) misspelled as acute accent (modifier letter)
- writing systems without spaces
Using Unicode Character Categories

- [https://perldoc.perl.org/perlunicode.html](https://perldoc.perl.org/perlunicode.html)
  - $text =~ s/\pP/ $1 /g;
  - $text =~ s/^s+///;
  - $text =~ s/s+$/;  
  - @tokens = split(/s+/, $text);
  - Optionally recombine email addresses, URLs etc.

Some problems
- haven’t (English; should be have n’t)
- instal · lació (Catalan; should be 1 token)
- single quote (punctuation) misspelled as acute accent (modifier letter)
- writing systems without spaces
Normalization

- Often part of tokenization

- Decimal comma to decimal point; separator of thousands
Normalization

- Often part of tokenization

- Decimal comma to decimal point; separator of thousands

- Unicode directed quotes and long hyphens to undirected ASCII
  - Sometimes mistaken for ACUTE ACCENT, PRIME (math) etc.
Normalization

- Often part of tokenization

- Decimal comma to decimal point; separator of thousands

- Unicode directed quotes and long hyphens to undirected ASCII
  - Sometimes mistaken for ACUTE ACCENT, PRIME (math) etc.

- \TeX\-like ASCII directed quotes ` ` and ' ' and hyphens -- and ---
Normalization

- Often part of tokenization

- Decimal comma to decimal point; separator of thousands

- Unicode directed quotes and long hyphens to undirected ASCII
  - Sometimes mistaken for ACUTE ACCENT, PRIME (math) etc.

- \texttt{TEX}-like ASCII directed quotes `` and ’’ and hyphens -- and ---

- English/ASCII punctuation in foreign writing systems
  - 「你看過《三國演義》嗎？」他問我。
  - “你看過‘三國演義’嗎?”他問我.
Normalization

- Often part of tokenization

- Decimal comma to decimal point; separator of thousands
- Unicode directed quotes and long hyphens to undirected ASCII
  - Sometimes mistaken for ACUTE ACCENT, PRIME (math) etc.

- TE\(\text{X}\)-like ASCII directed quotes ``,`` and `'` and hyphens -- and ---

- English/ASCII punctuation in foreign writing systems
  - 「你看過《三國演義》嗎？」他問我。
  - “你看過‘三國演義’嗎?”他問我。

- European/ASCII digits in Arabic, Devanagari etc.
  - ٠ ١ ٢ ٣ ٤ ٥ ٦ ٧ ٨ ٩ (Western Arabic/European)
  - ٠ ١ ٢ ٣ ٤ ٥ ٦ ٧ ٨ ٩ (Eastern Arabic)
  - ० १ २ ३ ४ ५ ६ ७ ८ ९ (Devanagari)
Let’s go to the sea.

Vámonos al mar. Vamos nos a el mar.

VERB? X NOUN PUNCT VERB PRON ADP DET NOUN PUNCT

- **Syntactic word** vs. orthographic word
- **Multi-word tokens**
- Two-level scheme:
  - Tokenization (low level, punctuation, concatenative)
  - Word segmentation (higher level, not necessarily concatenative)
Lexicalist hypothesis:
- Words (not morphemes) are the basic units in syntax
- Words enter in dependency relations
- Words are forms of lemmas and have morphological features

Orthographic vs. syntactic word
- Syntactically autonomous part of orthographic word
- Contractions \((al = a + el)\)
- Clitics \((vámonos = vamos + nos)\)
  - ¿A qué hora nos vamos mañana?
  - Nos despertamos a las cinco.
    “We wake up at five.”
  - Nuestro guía nos despierta a las cinco.
    “Our guide wakes us up at five.”
He abdicated in favour of his son Baudouin.
Segmentation as Part of Morphological Analysis

- **Arabic**
  - Enter "لابنه" (labnh)

- **Sanskrit**
  - Sanskrit Reader Companion: http://sanskrit.inria.fr/DICO/reader.fr.html
  - Select Input convention = Devanagari
  - Enter “सकलार्थशास्त्रसारं जगति समालोक्य विष्णुशर्मदम्” (sakalārthaśāstrasāraṁ jagati samālokya viṣṇuśarmedam)

- **German compound splitting (unsupervised)**
We are now in Valencia.

現在我們在瓦倫西亞。

Xiàn zài wǒmen zài Wǎlúnxīyǎ.

Now we in Valencia.

ADV PRON ADP PROPN PUNCT
I went to the beauty salon of Kyōdō [Beyond-R.]
I went to the beauty salon of Kyōdō [Beyond-R.]
I went to the beauty salon of Kyōdō [Beyond-R.]
All the concrete country roads are the result of...

- Spaces delimit monosyllabic morphemes, not words.
- Multiple syllables without space occur in loanwords (bê tfông).
- Spaces are allowed to occur word-internally in Vietnamese UD.
Il touche environ 100 000 sesterces par an.
One syntactic word spans several orthographic words?

# text = Bin nach wie vor sehr zufrieden.
# text_en = I am still very satisfied.

1  Bin  sein  AUX  ...  6  cop  __ __
2  nach nach  ADP  ...  6  obl  __ __
3  wie  wie  ADV  ...  2  fixed  __ __
4  vor  vor  ADP  ...  2  fixed  __ __
5  sehr  sehr  ADV  ...  6  advmod  __ __
6  zufrieden  zufrieden  ADJ  ...  0  root  __  SpaceAfter=No
7  .  .  PUNCT  ...  6  obl  __ __
Fixed Expressions

One syntactic word spans several orthographic words?

*I am still very satisfied.*
Some corpora use the underscore character to glue MWEs together.

*I am still very satisfied.*

```
Bin nach_wie_vor sehr zufrieden .
Am after_like_before very satisfied .
AUX ADV ADV ADJ PUNCT
```
Some corpora use the underscore character to glue MWEs together.

- Durante la presentación del libro “La_prosperidad_por_medio_de_la_investigación._La_investigación_básica_en_EEUU”, editado por la Comunidad_de_Madrid, el secretario general de la Confederación_Empresarial_de_Madrid-CEOE (CEIM), Alejandro_Couceiro, abogó por la formación de los investigadores en temas de innovación tecnológica.

- Lemmas?
- Tags?
Word Segmentation Summary

- When to split?
  - Only part of the token involved in a relation to something outside the token? Split!

- Hard time finding POS tag? Split!
- Hard time finding dependency relation? Don’t split!
- Or not hard time but the relation would be compound, flat, fixed or goeswith.
- Border case? Keep orthographic words (if they exist).
- Words with spaces
- Vietnamese writing system
- Very restricted set of exceptions (numbers)
- Special relations elsewhere (fixed, compound)
Word Segmentation Summary

- When to split?
  - Only part of the token involved in a relation to something outside the token? Split!
  - Hard time finding POS tag? Split!
  - Border case? Keep orthographic words (if they exist).
  - Words with spaces
  - Vietnamese writing system
  - Very restricted set of exceptions (numbers)
  - Special relations elsewhere (fixed, compound, flat, goeswith)
Word Segmentation Summary

- When to split?
  - Only part of the token involved in a relation to something outside the token? Split!
  - Hard time finding POS tag? Split!
  - Hard time finding dependency relation? Don’t split!
    - Or not hard time but the relation would be compound, flat, fixed or goeswith.
When to split?

- Only part of the token involved in a relation to something outside the token? Split!
- Hard time finding POS tag? Split!
- Hard time finding dependency relation? Don’t split!
  - Or not hard time but the relation would be compound, flat, fixed or goeswith.
- Border case? Keep orthographic words (if they exist).
When to split?
- Only part of the token involved in a relation to something outside the token? Split!
- Hard time finding POS tag? Split!
- Hard time finding dependency relation? Don’t split!
  - Or not hard time but the relation would be compound, flat, fixed or goeswith.
- Border case? Keep orthographic words (if they exist).

Words with spaces
- Vietnamese writing system
- Very restricted set of exceptions (numbers)
- Special relations elsewhere (fixed, compound)
# text = Vámonos al mar.
# text_en = Let’s go to the sea.

<table>
<thead>
<tr>
<th>ID</th>
<th>FORM</th>
<th>LEMMA</th>
<th>UPOS</th>
<th>HEAD</th>
<th>MISC</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-2</td>
<td>Vámonos</td>
<td>_</td>
<td>_</td>
<td>_</td>
<td>_</td>
</tr>
<tr>
<td>1</td>
<td>Vamos</td>
<td>ir</td>
<td>VERB</td>
<td>0</td>
<td>root</td>
</tr>
<tr>
<td>2</td>
<td>nos</td>
<td>nosotros</td>
<td>PRON</td>
<td>1</td>
<td>obj</td>
</tr>
<tr>
<td>3-4</td>
<td>al</td>
<td>_</td>
<td>_</td>
<td>_</td>
<td>_</td>
</tr>
<tr>
<td>3</td>
<td>a</td>
<td>a</td>
<td>ADP</td>
<td>5</td>
<td>case</td>
</tr>
<tr>
<td>4</td>
<td>el</td>
<td>el</td>
<td>DET</td>
<td>5</td>
<td>det</td>
</tr>
<tr>
<td>5</td>
<td>mar</td>
<td>mar</td>
<td>NOUN</td>
<td>1</td>
<td>obl</td>
</tr>
<tr>
<td>6</td>
<td>.</td>
<td>.</td>
<td>PUNCT</td>
<td>1</td>
<td>punct</td>
</tr>
</tbody>
</table>
Recoverability: CoNLL-U Format

```plaintext
# text = Vámonos al mar.
# text_en = Let’s go to the sea.

<table>
<thead>
<tr>
<th>ID</th>
<th>FORM</th>
<th>LEMMA</th>
<th>UPOS</th>
<th>HEAD</th>
<th>_</th>
<th>MISC</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-2</td>
<td>Vámonos</td>
<td>_</td>
<td>_</td>
<td>_</td>
<td></td>
<td>_</td>
</tr>
<tr>
<td>1</td>
<td>Vamos</td>
<td>ir</td>
<td>VERB</td>
<td>0</td>
<td>root</td>
<td>_</td>
</tr>
<tr>
<td>2</td>
<td>nos</td>
<td>nosotros</td>
<td>PRON</td>
<td>1</td>
<td>obj</td>
<td>_</td>
</tr>
<tr>
<td>3-4</td>
<td>al</td>
<td>_</td>
<td>_</td>
<td>_</td>
<td></td>
<td>_</td>
</tr>
<tr>
<td>3</td>
<td>a</td>
<td>a</td>
<td>ADP</td>
<td>5</td>
<td>case</td>
<td>_</td>
</tr>
<tr>
<td>4</td>
<td>el</td>
<td>el</td>
<td>DET</td>
<td>5</td>
<td>det</td>
<td>_</td>
</tr>
<tr>
<td>5-6</td>
<td>mar.</td>
<td>_</td>
<td>_</td>
<td>_</td>
<td></td>
<td>_</td>
</tr>
<tr>
<td>5</td>
<td>mar</td>
<td>mar</td>
<td>NOUN</td>
<td>1</td>
<td>obl</td>
<td>_</td>
</tr>
<tr>
<td>6</td>
<td>.</td>
<td>.</td>
<td>PUNCT</td>
<td>1</td>
<td>punct</td>
<td>_</td>
</tr>
</tbody>
</table>
```

Tokenization and Word Segmentation
Tokenization vs. Multi-word Tokens

- Parallelism among closely related languages
  - **ca:** informar-se sobre el patrimoni cultural
  - **es:** informarse sobre el patrimonio cultural
  - **en:** learn about cultural heritage

- **ca:** L’únic que veig és => L’ únic que veig és
- **en:** don’t => don’t

- No strict guidelines for tokenization (yet)
  - **UD English:** non-stop, post-war: single-word tokens
  - **UD Czech:** non-stop would be split to three tokens
  - Abbreviations: etc.
    - End of sentence...
- Punctuation involved? Low level!
  - Exceptions: Spanish-Catalan parallelism.
Tokenization vs. Multi-word Tokens Summary

- Punctuation involved? Low level!
  - Exceptions: Spanish-Catalan parallelism.

- Boundary between two letters? Typically high level.
  - Exceptions: Chinese, Japanese.
Tokenization vs. Multi-word Tokens Summary

- Punctuation involved? Low level!
  - Exceptions: Spanish-Catalan parallelism.

- Boundary between two letters? Typically high level.
  - Exceptions: Chinese, Japanese.

- Non-concatenative? High level!
Errors in Underlying Text

- We do not want to hide errors (learning robust parsers!)
  - But: reference corpora (linguistic research) may want to hide them.
Errors in Underlying Text

- We do not want to hide errors (learning robust parsers!)
  - But: reference corpora (linguistic research) may want to hide them.
- Possibilities:
  - Typo not involving word boundary
    - FORM = annotation; LEMMA = annotation; FEATS: Typo=Yes; MISC: Correct=annotation
  - Wrongly split word: annotation
  - Wrongly merged words: thecar
  - Fix tokenization (i.e. two lines); first line MISC: SpaceAfter=No | CorrectSpaceAfter=Yes
  - Sentence segmentation can be affected, too!
We do not want to hide errors (learning robust parsers!)
  - But: reference corpora (linguistic research) may want to hide them.

Possibilities:
- Typo not involving word boundary
  - FORM = anotation; LEMMA = annotation; FEATS: Typo=Yes; MISC: Correct=annotation

Wrongly split word:
Errors in Underlying Text

- We do not want to hide errors (learning robust parsers!)
  - But: reference corpora (linguistic research) may want to hide them.

- Possibilities:
  - Typo not involving word boundary
    - FORM = annotation; LEMMA = annotation; FEATS: Typo=Yes; MISC: Correct=annotation
  
  \[
  \text{goeswith} \\
  \text{ann} \quad \text{otation}
  \]
  \[
  \text{X} \quad \text{X}
  \]

- Wrongly split word:

- Wrongly merged words: thecar
  - Fix tokenization (i.e. two lines); first line MISC: SpaceAfter=No | CorrectSpaceAfter=Yes
  - Sentence segmentation can be affected, too!
Errors in Underlying Text

- Wrong morphology: *the cars is produced in Detroit*

- Example:
  - *cs: viděl moři* → *he saw the sea*
  - Should be *moře*
  - Would be Case=Acc (disambiguated from Case=Acc,Gen,Nom,Voc)
  - This form is Case=Dat,Loc (but which one?)
  - *cestoval k moři* → *he traveled to the sea*
  - *plavil se po moři* → *he sailed the sea*
Wrong morphology: the *cars is produced in Detroit*
- Not like normal typo (*the car iss produced...*)
Errors in Underlying Text

- Wrong morphology: *the cars is produced in Detroit*
  - Not like normal typo (*the car iss produced...*)
  - Not obvious what is correct
    - *the car is*
    - *the cars are*
Errors in Underlying Text

- Wrong morphology: *the cars is produced in Detroit*
  - Not like normal typo (*the car iss produced...*)
  - Not obvious what is correct
  - *the car is*
  - *the cars are*

- Suggestion: select which word to fix, e.g. *cars to car*
- FORM = *cars*; FEATS: Number=Plur; MISC: Correct=car | CorrectNumber=Sing
Errors in Underlying Text

- Wrong morphology: *the cars is produced in Detroit*
  - Not like normal typo (*the car iss produced...*)
  - Not obvious what is correct
    - *the car is*
    - *the cars are*

- Suggestion: select which word to fix, e.g. *cars to car*
- FORM = *cars*; FEATS: Number=Plur; MISC: Correct=car | CorrectNumber=Sing
- cs: *viděl moři* “he saw the sea”
  - Should be *moře*
  - Would be Case=Acc (disambiguated from Case=Acc,Gen,Nom,Voc)
  - This form is Case=Dat,Loc (but which one?)
- *cestoval k moři* “he traveled to the sea” Case=Dat
- *plavil se po moři* “he sailed the sea” Case=Loc
Tokenization Alignment

- If you need to match two different tokenizations
- Use case: evaluation of end-to-end parsing systems
Tokenization Alignment

- If you need to match two different tokenizations
- Use case: evaluation of end-to-end parsing systems

- Normalization involved? Bad luck...
  - Normalization rules needed
  - Or: Longest common subsequence (LCS) algorithm
Tokenization Alignment

- If you need to match two different tokenizations
- Use case: evaluation of end-to-end parsing systems

- Normalization involved? Bad luck…
  - Normalization rules needed
  - Or: Longest common subsequence (LCS) algorithm

- Otherwise easy
  - Non-whitespace character offsets
Evaluation Metrics

- Align system-output tokens to gold tokens

*Al-Zaman*: American forces killed Shaikh Abdullah al-Ani, the preacher at the mosque in the town of Qaim, near the Syrian border.

**GOLD:**  
*Al-Zaman*: American forces killed Shaikh  
OFFSET: 0-1 2 3-7 8 9-16 17-22 23-28 29-34

- All characters except for whitespace match => easy align!

**SYSTEM:**  
*Al-Zaman*: American forces killed Shaikh  
OFFSET: 0-7 8 9-16 17-22 23-28 29-34
Evaluation Metrics

- Align system-output tokens to gold tokens

*Die Kosten sind definitiv auch im Rahmen.*

**GOLD:** Die Kosten sind definitiv auch im Rahmen.

**SPLIT:** Die Kosten sind definitiv auch in dem Rahmen.

**OFFSET:** 0-2 3-8 9-12 13-21 22-25 26-27 28-33 34

- Corresponding but not identical spans?
- Find longest common subsequence

**SYSTEM:** Kosten sind definitiv auch im Rahmen.

**SPLIT:** Kosten sind *de finitiv* auch im Rahmen.

**OFFSET:** 3-8 9-12 13-21 22-25 26-27 28-33 34
Evaluation Metrics

- Align system-output tokens to gold tokens

Die Kosten sind definitiv auch im Rahmen.

**GOLD:** Die Kosten sind definitiv auch **im** Rahmen.

**SPLIT:** Die Kosten sind definitiv auch **in dem** Rahmen.

**OFFSET:** 0-2 3-8 9-12 13-21 22-25 **26-27** 28-33 34

- Corresponding but not identical spans?
- Find longest common subsequence

**SYSTEM:** auch **im** Rahmen.

**SPLIT:** auch **in einem, dem alle zustimmen**, Rahmen.

**OFFSET:** 22-25 **26-27** 28-33 34