Chinese Word Segmentation

Daniel Zeman

October 30, 2020

LANGTECH

EUROPEAN UNION
European Structural and Investment Fund
Operational Programme Research,
Development and Education

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

What Is a Word?

- Phonological word: e.g. English words could be defined according to stress
  - Sproat 1992:
    - *SPARK plug*
    - *electrical ENGINEER*
  - Similarly cs: *v domě* “in the house” is one phonological word

What Is a Word?

- Phonological word: e.g. English words could be defined according to stress
  - Sproat 1992:
    - *SPARK plug*
    - *electrical ENGINEER*
  - Similarly `cs:` *v domě* "in the house" is one phonological word
- Orthographic word: roughly between two whitespaces (plus some rules for punctuation)
  - Differences between languages:
    - `en:` *life insurance company employee*
    - `de:` *Lebensversicherungsgesellschaftsangestellter*

What Is a Word?

- Phonological word: e.g. English words could be defined according to stress
  - Sproat 1992:
    - *SPARK plug*
    - *electrical ENGINEER*
  - Similarly `cs:` *v domě* “in the house” is one phonological word
- Orthographic word: roughly between two whitespaces (plus some rules for punctuation)
  - Differences between languages:
    - `en:` *life insurance company employee*
    - `de:` *Lebensversicherungsgesellschaftsangestellter*
  - Chinese (and other languages) does not mark words orthographically
  - Nevertheless, it is desirable to be able to define a word in NLP

Words in Chinese

- No orthographic word boundary
  - 这个多少钱？ ... simplified Chinese characters
  - *zhè ge duō shǎo qián ?* ... transcription
  - *this piece much little money ?* ... literal character-based

Words in Chinese

- No orthographic word boundary
  - 这个多少钱？ ... simplified Chinese characters
  - *zhè ge duō shǎo qián ?* ... transcription
  - *this piece much little money ?* ... literal character-based
  - *Zhège duōshǎo qián?* ... proposed segmentation
  - *This how-much money?* ... literal word-based

Words in Chinese

- No orthographic word boundary
  - 这个多少钱？ ... simplified Chinese characters
  - *zhè ge duō shǎo qián ?* ... transcription
  - *this piece much little money ?* ... literal character-based
  - *Zhège duōshǎo qián?* ... proposed segmentation
  - *This how-much money?* ... literal word-based
  - "How much is this?" ... translation

Words in Chinese

- No orthographic word boundary
  - 这个多少钱？ ... simplified Chinese characters
  - *zhè ge duō shǎo qián ?* ... transcription
  - *this piece much little money ?* ... literal character-based
  - *Zhège duōshǎo qián?* ... proposed segmentation
  - *This how-much money?* ... literal word-based
  - “How much is this?” ... translation
- Character = syllable ˜ morpheme
  - Thousands of characters mapped on about 400 possible syllables ⇒ gigantic homonymy!
  - Typical new words are compounds
  - Phonetically approximated loanwords

Do We Need Words?

- For NLP it is desirable to have words
  - Dictionaries
  - Most NLP applications assume there are words as units of the text
    - Indexing, language modeling, translation modeling
  - Meaning of sequence of characters cannot be always predicted from the meaning of the parts

Do We Need Words?

- For NLP it is desirable to have words
  - Dictionaries
  - Most NLP applications assume there are words as units of the text
    - Indexing, language modeling, translation modeling
  - Meaning of sequence of characters cannot be always predicted from the meaning of the parts
- Roosevelt Road (Taipei) = 罗斯福路 = *luō sī fú lù*
  - 罗 *luō* = net for catching birds
  - 斯 *sī* = this, thus, such
  - 福 *fú* = happiness, good fortune, blessing
  - 路 *lù* = road, path, way

Do We Need Words?

- For NLP it is desirable to have words
  - Dictionaries
  - Most NLP applications assume there are words as units of the text
    - Indexing, language modeling, translation modeling
  - Meaning of sequence of characters cannot be always predicted from the meaning of the parts
- There are dozens of other characters pronounced *fu*
  - 罗 *luō* = net for catching birds
  - 斯 *sī* = this, thus, such
  - 蝮 *fù* = venomous snake, viper
  - 路 *lù* = road, path, way

Do We Need Words?

- For NLP it is desirable to have words
  - Dictionaries
  - Most NLP applications assume there are words as units of the text
    - Indexing, language modeling, translation modeling
  - Meaning of sequence of characters cannot be always predicted from the meaning of the parts
- There are dozens of other characters pronounced *fu*
  - 罗 *luō* = net for catching birds
  - 斯 *sī* = this, thus, such
  - 腐 *fǔ* = rot, decay, spoil, rotten
  - 路 *lù* = road, path, way

What Is a Word in Chinese?

- Which sequences should be chosen to become words?
  - 火车站 = *huǒ chē zhàn* = fire vehicle stand = (railroad) station
    - Three, two, or one word?
  - 电话 = *diàn huà* = electric speech = telephone
  - 电脑 = *diàn nǎo* = electric brain = computer
    - Two words or one?
  - 北海 = *běi hǎi* = North Sea:
    - Two words as in English *(North Sea)* and Czech *(Severní moře)*?
    - One word as in German *(Nordsee)* and Dutch *(Noordzee)*?

What Is a Word in Chinese?

- A more peculiar case: verb-object constructions
- Transitive verbs have a "default object"
- 吃饭 = *chīfàn* = "to eat" (lit. "eat cooked rice")
- Used in sentences like "He likes eating" or "She is going to eat"
  - 我们吃饭吧 = *wǒmen chīfàn ba* = lit. "we eat(-cooked-rice) suggest" = "Let's eat"
- If there is a real object it replaces the default
  - 今天晚上吃中国菜 = *jīntiān wǎnshang chī zhōngguó cài* = lit. "today evening eat China dish" = "We are going to eat Chinese tonight"

- The confusion: Is *chīfàn* a morphological form of the verb *chī* or is it two words, *chī* and *fàn*?

Standards

- There have been various attempts to standardize words in Chinese
- GB/T 13715-92 (*guóbiāo*, 国标, “National Standard”)
  - Mainland China (PRC)
- Popular corpora, such as
  - Academia Sinica Treebank (Taiwan)
  - Penn Chinese Treebank (University of Pennsylvania)
  - City University Corpus (Hong Kong)

Words in Japanese

I went to a beauty salon in Kyōdō [, Beyond-R.]

経堂	の	美容室	に	行っ	て	き	まし	た
Kyōdō	no	miyōshitsu	ni	it	te	ki	mashi	ta
経堂	の	美容室	に	行く	て	来る	ます	た
Kyōdō	of	beauty-salon	to	go	CONV	come	will	PAST
PROPN	**ADP**	**NOUN**	**ADP**	**VERB**	**SCONJ**	**AUX**	**AUX**	**AUX**

経堂	の	美容室	に	行って	きました
Kyōdō	no	miyōshitsu	ni	itte	kimashita
経堂	の	美容室	に	行く	来る
Kyōdō	of	beauty-salon	to	going	come
PROPN	**ADP**	**NOUN**	**ADP**	**VERB**	**VERB**

経堂の	美容室に	行って	きました
Kyōdōno	miyōshitsuni	itte	kimashita
経堂	美容室	行く	来る
of-Kyōdō	to-beauty-salon	going	come
PROPN	**NOUN**	**VERB**	**VERB**

Methods

- Supervised
  - Vocabulary (manually created or learned from corpus
    - Main problem: OOV (out-of-vocabulary words), i.e. those that do not occur in training data but occur in test data
  - Greedy approach: left-to-right, longest match
    - Backtracking or out-of-vocabulary characters
    - Possible ambiguous readings
  - Viterbi algorithm
    - Score possible segmentation paths using N-gram language model
    - Search for the path with the highest score
- Unsupervised
  - Mutual co-occurrence of characters
  - Explore language regularities similarly to the unsupervised morphemic segmentation

Example: Nokia Beijing System at the SIGHAN Bakeoff 2007

- List of recurring OOV strings created before the main segmentation process
  - Only OOV strings of 2 or 3 characters are considered as possible words
  - Heuristics: 的 (*de*, possessive particle) is a high-frequency single-character word. Don't consider repeating strings that contain it
- All possible paths of segmentation are considered
- Every candidate word is categorized into certain type
  - Dictionary (lexicon acquired from training data)
  - Factoid (Latin letters, Arabic numbers)
  - Named entities: names of persons and locations. Organizations left for postprocessing.
  - Recurring OOV
  - Other

Example: Nokia Beijing System at the SIGHAN Bakeoff 2007

- The paths are scored (tag-based N-gram language model)
  - Category transition probability $P(T_j|T_{j-1})$
  - Word emission probability $P(W_i|T_j)$
  - Product $\Rightarrow$ probability of the path
- OOV words detected during postprocessing based on character information
  - Merge two single characters to a new word
  - Combine parts of organization names

- Jiang Li, Rile Hu, Guohua Zhang, Yuezhong Tang, Zhanjiang Song, Xia Wang: NOKIA Research Center Beijing Chinese Word Segmentation System for the SIGHAN Bakeoff 2007. In: Proceedings of the Sixth SIGHAN Workshop on Chinese Language Processing, pp. 86–89, Hyderabad, India, 2008.
  `http://aclweb.org/anthology-new/I/I08/I08-4012.pdf`

Unsupervised Segmentation

- Cache-based model
  - The more times a word is observed, the more likely it is to be proclaimed a word the next time
  - Probability distribution covers infinite number of words
  - Yet shorter words are preferred

- Kevin Knight: Bayesian Inference with Tears. Marina del Rey, CA, USA, September 2009 `http://www.isi.edu/natural-language/people/bayes-with-tears.pdf`

Word Generation Model

1. Word $=$ empty.
2. Pick a Chinese character from the uniform distribution $(1/C)$.
3. Add the chosen character to the end of Word.
4. With probability 0.5, go to 2.
   With probability 0.5, quit and output Word.

5. E.g. if there are only three characters *a*, *b*, *c*, all two-character words get the same probability $P_0 = 1/3 \times 1/2 \times 1/3 \times 1/2 = 1/36 = 0.028$

Text Generation Model

1. $H = 0$. (H is the length of the history, the number of decisions taken so far.)
2. With probability $\alpha/(\alpha + H)$, generate a Chinese word according to the base distribution P_0.
   With probability $H/(\alpha + H)$, generate a Chinese word using the cache of words generated so far.
3. Write down the word just chosen.
4. With probability 0.99, $H = H + 1$; go to 2.
   With probability 0.01, quit.

5. Prior parameters: $P(\text{quit}) = 0.01$; α *(concentration parameter)*. Let's pick $\alpha = 1$.

Probability of a Word from Cache

$$P(w) = \frac{H}{\alpha + H} \times \frac{cacheCount(w)}{H}$$

$$P(w) = \frac{cacheCount(w)}{\alpha + H}$$

Probability of a Word Sequence $w_1 \dots w_n$

$$\prod_{i=1}^{n} \frac{\alpha \times P_0(w_i) + cacheCount(w_i)}{\alpha + i - 1} \times 0.99^{n-1} \times 0.01$$

Example

- We observe character sequence ab

Example

- We observe character sequence ab
- First possible derivation: one word ab
  - Probability of the derivation:

$$
\begin{aligned}
P &= \frac{\alpha \times P_0(ab) + 0}{\alpha + 0} \times 0.01 = P_0(ab) \times 0.01 \\
&= (1/3 \times 1/2 \times 1/3 \times 1/2) \times 0.01 = \underline{0.00028}
\end{aligned}
$$

Example

- We observe character sequence ab
- First possible derivation: one word ab
  - Probability of the derivation:

$$P = \frac{\alpha \times P_0(ab) + 0}{\alpha + 0} \times 0.01 = P_0(ab) \times 0.01$$
$$= (1/3 \times 1/2 \times 1/3 \times 1/2) \times 0.01 = \underline{0.00028}$$

- Other possible derivation: two words $a\ b$
  - Probability of the derivation:

$$P = \frac{\alpha \times P_0(a) + 0}{\alpha + 0} \times 0.99 \times \frac{\alpha \times P_0(b) + 0}{\alpha + 1} \times 0.01$$
$$= (1/3 \times 1/2) \times 0.99 \times (1/3 \times 1/2)/2 \times 0.01 = \underline{0.00013}$$

Example

- We observe character sequence ab
- First possible derivation: one word ab
  - Probability of the derivation:

$$
\begin{aligned}
P &= \frac{\alpha \times P_0(ab) + 0}{\alpha + 0} \times 0.01 = P_0(ab) \times 0.01 \\
&= (1/3 \times 1/2 \times 1/3 \times 1/2) \times 0.01 = \underline{0.00028}
\end{aligned}
$$

- Other possible derivation: two words a b
  - Probability of the derivation:

$$
\begin{aligned}
P &= \frac{\alpha \times P_0(a) + 0}{\alpha + 0} \times 0.99 \times \frac{\alpha \times P_0(b) + 0}{\alpha + 1} \times 0.01 \\
&= (1/3 \times 1/2) \times 0.99 \times (1/3 \times 1/2)/2 \times 0.01 = \underline{0.00013}
\end{aligned}
$$

- The rest of the probability mass goes to character sequences other than ab

Example

- We observe character sequence aa
- First possible derivation: one word aa
  - Probability of the derivation:

$$P = \frac{\alpha \times P_0(aa) + 0}{\alpha + 0} \times 0.01 = P_0(aa) \times 0.01$$
$$= (1/3 \times 1/2 \times 1/3 \times 1/2) \times 0.01 = \underline{0.00028}$$

- The rest of the probability mass goes to character sequences other than aa

Example

- We observe character sequence aa
- First possible derivation: one word aa
  - Probability of the derivation:

$$
\begin{aligned}
P &= \frac{\alpha \times P_0(aa) + 0}{\alpha + 0} \times 0.01 = P_0(aa) \times 0.01 \\
&= (1/3 \times 1/2 \times 1/3 \times 1/2) \times 0.01 = \underline{0.00028}
\end{aligned}
$$

- Other possible derivation: two words $a\ a$
  - Probability of the derivation:

$$
\begin{aligned}
P &= \frac{\alpha \times P_0(a) + 0}{\alpha + 0} \times 0.99 \times \frac{\alpha \times P_0(a) + 1}{\alpha + 1} \times 0.01 \\
&= (1/3 \times 1/2) \times 0.99 \times (1/3 \times 1/2 + 1)/2 \times 0.01 = \underline{0.00096}
\end{aligned}
$$

- The rest of the probability mass goes to character sequences other than aa

Example

- We observe character sequence $abab$
- $abab$

$$P = \frac{\alpha \times P_0(abab) + 0}{\alpha + 0} \times 0.01 = P_0(abab) \times 0.01$$
$$= (1/3 \times 1/2 \times 1/3 \times 1/2 \times 1/3 \times 1/2 \times 1/3 \times 1/2) \times 0.01 = \underline{0.0000077}$$

Example

- We observe character sequence $abab$
- $abab$

$$P = \frac{\alpha \times P_0(abab) + 0}{\alpha + 0} \times 0.01 = P_0(abab) \times 0.01$$
$$= (1/3 \times 1/2 \times 1/3 \times 1/2 \times 1/3 \times 1/2 \times 1/3 \times 1/2) \times 0.01 = \underline{0.0000077}$$

- $a - b - a - b$

$$P = \frac{\alpha \times P_0(a) + 0}{\alpha + 0} \times 0.99 \times \frac{\alpha \times P_0(b) + 0}{\alpha + 1} \times 0.99$$
$$\times \frac{\alpha \times P_0(a) + 1}{\alpha + 2} \times 0.99 \times \frac{\alpha \times P_0(b) + 1}{\alpha + 3} \times 0.01$$
$$= \frac{1/6}{1} \times 0.99 \times \frac{1/6}{2} \times 0.99 \times \frac{1/6 + 1}{3} \times 0.99 \times \frac{1/6 + 1}{4} \times 0.01 = \underline{0.0000153}$$

Example

- We observe character sequence $abab$
- $a - b - a - b$

$$P = \frac{\alpha \times P_0(a) + 0}{\alpha + 0} \times 0.99 \times \frac{\alpha \times P_0(b) + 0}{\alpha + 1} \times 0.99$$
$$\times \frac{\alpha \times P_0(a) + 1}{\alpha + 2} \times 0.99 \times \frac{\alpha \times P_0(b) + 1}{\alpha + 3} \times 0.01$$
$$= \frac{1/6}{1} \times 0.99 \times \frac{1/6}{2} \times 0.99 \times \frac{1/6 + 1}{3} \times 0.99 \times \frac{1/6 + 1}{4} \times 0.01 = \underline{0.0000153}$$

- $ab - ab$

$$P = \frac{\alpha \times P_0(ab) + 0}{\alpha + 0} \times 0.99 \times \frac{\alpha \times P_0(ab) + 1}{\alpha + 1} \times 0.01$$
$$= \frac{1/36}{1} \times 0.99 \times \frac{1/36 + 1}{2} \times 0.01 = \underline{0.0001413}$$

Example

- We observe character sequence $abab$
- $ab - ab$

$$P = \frac{\alpha \times P_0(ab) + 0}{\alpha + 0} \times 0.99 \times \frac{\alpha \times P_0(ab) + 1}{\alpha + 1} \times 0.01$$
$$= \frac{1/36}{1} \times 0.99 \times \frac{1/36 + 1}{2} \times 0.01 = \underline{0.0001413}$$

- $aba - b$

$$P = \frac{\alpha \times P_0(aba) + 0}{\alpha + 0} \times 0.99 \times \frac{\alpha \times P_0(b) + 0}{\alpha + 1} \times 0.01$$
$$= \frac{1/216}{1} \times 0.99 \times \frac{1/6}{2} \times 0.01 = \underline{0.0000038}$$

Example

- We observe character sequence $abab$
- $aba - b$

$$P = \frac{\alpha \times P_0(aba) + 0}{\alpha + 0} \times 0.99 \times \frac{\alpha \times P_0(b) + 0}{\alpha + 1} \times 0.01$$
$$= \frac{1/216}{1} \times 0.99 \times \frac{1/6}{2} \times 0.01 = \underline{0.0000038}$$

- $a - b - ab$

$$P = \frac{\alpha \times P_0(a) + 0}{\alpha + 0} \times 0.99 \times \frac{\alpha \times P_0(b) + 0}{\alpha + 1} \times 0.99 \times \frac{\alpha \times P_0(ab) + 0}{\alpha + 2} \times 0.01$$
$$= \frac{1/6}{1} \times 0.99 \times \frac{1/6}{2} \times 0.99 \times \frac{1/36}{3} \times 0.01 = \underline{0.0000013}$$

Search Problem

- We can compute probability of any derivation of a given character sequence
- Unfortunately, examining all possible derivations of a long sequence is not tractable

- So how do we find the highest-ranking derivation?

Search Problem

- We can compute probability of any derivation of a given character sequence
- Unfortunately, examining all possible derivations of a long sequence is not tractable

- So how do we find the highest-ranking derivation?
  - Enumerating $\rightarrow$ sampling
  - All derivations $\rightarrow$ some derivations
    - Selected randomly but in proportion to their probabilities

Gibbs Sampling

1. Start with some initial sample (e.g., a random segmentation of the sequence)

Gibbs Sampling

1. Start with some initial sample (e.g., a random segmentation of the sequence)
2. Make a small change to the sample by weighted coin flip
   - Select i-th position and decide whether there should be a word boundary. 'No change' is also a valid outcome of the coin flip
   - The probability with which we make the change should be proportional to the entire $P(\text{derivation})$ with the change
     - Before the coin flip, incrementally compute probabilities of both derivations with and without word boundary at position i. Then bias the coin
   - We are more likely to make a change that leads to a better segmentation. Reaching the global optimum is not guaranteed but we are likely to be headed in its general direction

Gibbs Sampling

1. Start with some initial sample (e.g., a random segmentation of the sequence)
2. Make a small change to the sample by weighted coin flip
    - Select i-th position and decide whether there should be a word boundary. 'No change' is also a valid outcome of the coin flip
    - The probability with which we make the change should be proportional to the entire $P(\text{derivation})$ with the change
        - Before the coin flip, incrementally compute probabilities of both derivations with and without word boundary at position i. Then bias the coin
    - We are more likely to make a change that leads to a better segmentation. Reaching the global optimum is not guaranteed but we are likely to be headed in its general direction
3. Collect whole counts off the new sample (which might be the same as the old sample if the segmentation didn't change)

Gibbs Sampling

1. Start with some initial sample (e.g., a random segmentation of the sequence)
2. Make a small change to the sample by weighted coin flip
   - Select i-th position and decide whether there should be a word boundary. 'No change' is also a valid outcome of the coin flip
   - The probability with which we make the change should be proportional to the entire $P(\text{derivation})$ with the change
     - Before the coin flip, incrementally compute probabilities of both derivations with and without word boundary at position i. Then bias the coin
   - We are more likely to make a change that leads to a better segmentation. Reaching the global optimum is not guaranteed but we are likely to be headed in its general direction
3. Collect whole counts off the new sample (which might be the same as the old sample if the segmentation didn't change)
4. Until tired, go to 2. (Next time, change $(i+1)$-th position.)

Other Languages without Word Boundaries

- Japanese
  - 語の厳密な定義は各言語によるが、一般に以下の性質がある。
- Korean written in *hanja* (Chinese characters).
  In *hangul* (Korean script), spaces are used:
  - 다른 낱말이나 낱말의 일부와 합쳐진 낱말은 혼성어를 형성한다.
- Vietnamese uses Latin script but spaces delimit monosyllabic morphemes, not words
  - *Từ là đơn vị nhỏ nhất, cấu tạo ổn định, mang nghĩa hoàn chỉnh ...*
- Not to be confused with polysynthetic languages (Siberia, Americas)
  - They intricately compose words of many lexical morphemes that are not easily told apart
  - That's why linguists decided not to separate them orthographically
  - One long word may cover a whole sentence
  - Nevertheless, words usually are separated. They are just long

Word Boundaries Are Modern Development

Greek manuscript from the 4th century:

ποιη εν τη βασιλει-
α αυτου· και ουτως
πασαι αι γυναικες
περιθησουσιν τι-
μην τοις ανδρασι
εαυτων απο πτω-
χου εως πλουσιου·
και ηρεσεν ο λο-
γος τω βασιλει· και
τοις αρχουσιν· και
εποιησεν ο βασι-
λευς καθ' α ελαλη-
σεν ο μαμουχεος·

Available Segmenters

- CoNLL UD Shared Task 2018 included word segmentation, among other things
- Results:
  `http://universaldependencies.org/conll18/results-words.html#zh_gsd`
- Some of the systems are publicly available, e.g., the Uppsala segmenter:
  `https://github.com/UppsalaNLP/segmenter`

- UDPipe (`https://ufal.mff.cuni.cz/udpipe`) is available as a web service and includes Chinese segmentation. Select a Chinese model at `https://lindat.mff.cuni.cz/services/udpipe/` (use `chinese-gsd` for traditional characters and `chinese-gsdsimp` for simplified characters)