Tokenization

Vũ Việt Trung

Tokenization

Hiệp hội Vận tải hàng hóa VN chính thức có văn bản báo cáo và kiến nghị Văn phòng Quốc hội

Hiệp_1 hội_3 Vận_1 tải_3 hàng_1 hóa_3 VN_0 chính_1 thức_3 có_0 văn_1 bản_3 báo_1 cáo_3 và_0 kiến_1 nghị_3 Văn_1 phòng_3 Quốc_1 hội_3

- 0: outside (0)
- 1: start (S)
- 2: inside (I)
- 3: end (E)

Language model

- Xác suất xuất hiện một cụm từ

Chain rule

$P(x, y) = P(x)P(y \mid x)$

$$
\begin{aligned}
P(w_1 \ldots w_n) &= P(w_1)P(w_2 \ldots w_n \mid w_1) \\
&= P(w_1)P(w_2 \mid w_1)P(w_3 \ldots w_n \mid w_1 w_2) \\
&= P(w_1)P(w_2 \mid w_1)P(w_3 \mid w_1 w_2)P(w_4 \ldots w_n \mid w_1 w_2 w_3) \\
&= \ldots \\
&= \prod_{i=1}^{n} P(w_i \mid w_1 \ldots w_{i-1})
\end{aligned}
$$

N-grams

- Markov assumption
  - history
- Maximum likelihood estimation (MLE)
  - Count f
- Unigram:

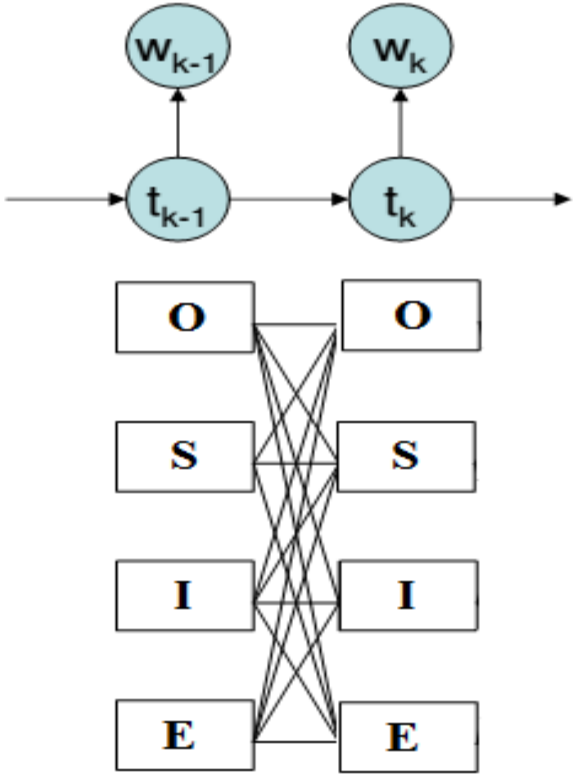
$$P(w_1 \ldots w_n) = \prod_{i=1}^{n} P(w_i \mid w_1 \ldots w_{i-1}) \approx \prod_{i=1}^{n} P(w_i)$$

- Bigram:

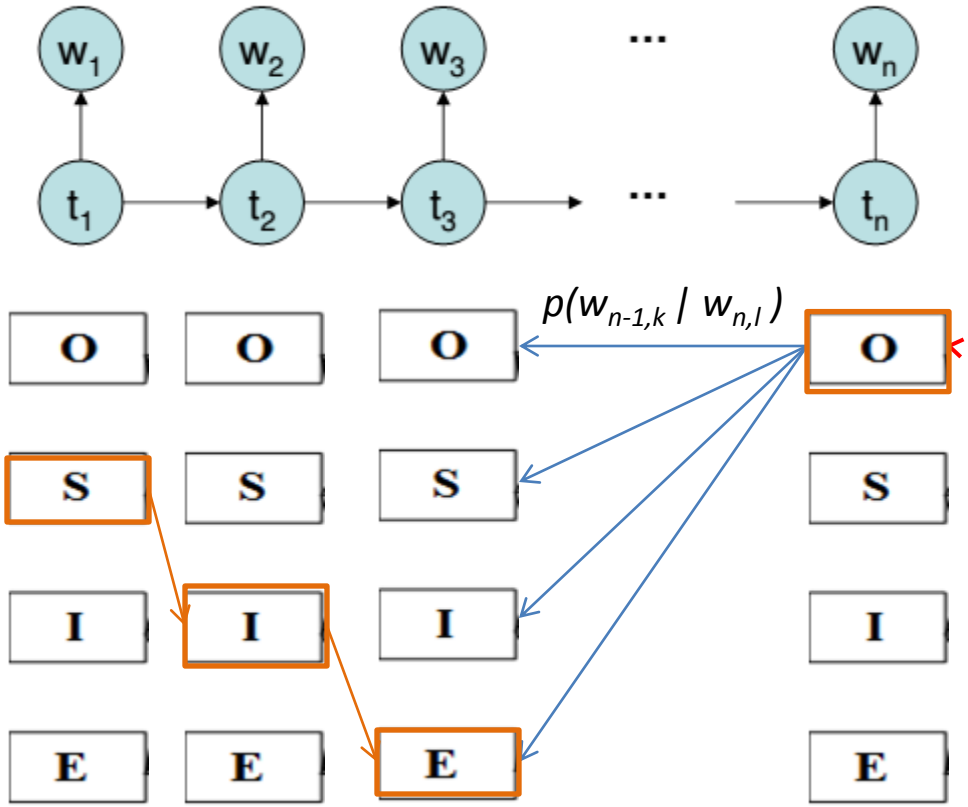
$$P(w_1 \ldots w_n) = \prod_{i=1}^{n} P(w_i \mid w_1 \ldots w_{i-1}) \approx \prod_{i=1}^{n} P(w_i \mid w_{i-1})$$

Labeled-bigram

- Dựa trên ý tưởng N-grams:
  - Mỗi từ được gắn thêm nhãn (0-3): *kiến_1 nghị_3*
  - Bài toán tách từ → Bài toán gán nhãn từ sao cho câu có xác suất lớn nhất
  - Tương tự giải thuật trong POS tagging (Viterbi)



Dynamic Programming



$V_{n,l} = max_k(V_{n-1,k} * p(w_{n-1,k} \mid w_{n,l}))$

$p_{n,l} = arg max_k(V_{n-1,k} * p(w_{n-1,k} \mid w_{n,l}))$

Tìm đường đi có xác suất lớn nhất (v_n)

Dùng log → nhân thành cộng, số không quá bé → tăng hiệu năng

Tính trọng số $p(w_{n-1,k} \mid w_{n,l})$

- Unsmooth

$$P(w_k) = \frac{C(w_k)}{\sum_w C(w_k)}$$

$$P(w_k \mid w_{k-1}) = \frac{C(w_{k-1}w_k)}{C(w_{k-1})}$$

- Smooth
  - Nhiều phương pháp:
    - Add one, Good Turing, Interpolation, Katz (backoff), Absolute Discouting,…
  - Kneser-Ney – hiệu quả nhất

Modified Kneser-Ney Smoothing

$$p_{\mathrm{KN}}(w_i|w_{i-n+1}^{i-1}) = \frac{c(w_{i-n+1}^{i}) - D(c(w_{i-n+1}^{i}))}{\sum_{w_i} c(w_{i-n+1}^{i})} + \gamma(w_{i-n+1}^{i-1}) p_{\mathrm{KN}}(w_i|w_{i-n+2}^{i-1})$$

$$D(c) = \begin{cases} 0 & \text{if } c = 0 \\ D_1 & \text{if } c = 1 \\ D_2 & \text{if } c = 2 \\ D_{3+} & \text{if } c \geq 3 \end{cases}$$

$$\begin{aligned} Y &= \frac{n_1}{n_1 + 2n_2} \\ D_1 &= 1 - 2Y\frac{n_2}{n_1} \\ D_2 &= 2 - 3Y\frac{n_3}{n_2} \\ D_{3+} &= 3 - 4Y\frac{n_4}{n_3} \end{aligned}$$

$n_1 = N_1(\bullet\bullet)$

$$p_{\mathrm{KN}}(w_i) = \frac{N_{1+}(\bullet w_i)}{N_{1+}(\bullet\bullet)}$$

$$N_{1+}(\bullet w_i) = |\{w_{i-1} : c(w_{i-1} w_i) > 0\}|$$

$$N_{1+}(\bullet\bullet) = |\{(w_{i-1}, w_i) : c(w_{i-1} w_i) > 0\}|$$

$$\gamma(w_{i-n+1}^{i-1}) = \frac{D_1 N_1(w_{i-n+1}^{i-1}\bullet) + D_2 N_2(w_{i-n+1}^{i-1}\bullet) + D_{3+} N_{3+}(w_{i-n+1}^{i-1}\bullet)}{\sum_{w_i} c(w_{i-n+1}^{i})}$$

Kết hợp vào bài toán tokenize

- Dictionary
  - Theo từng topic, p_{dict}=1 nếu có xuất hiện
  - $p_{mix} = 0.5\ p_{KN} + 0.5\ p_{dict}$
- Mix các corpus
  - $p_{KN} = w_1 * p_{KN1} + w_2 * p_{KN2} + \ldots + w_n * p_{KNn}$
    - $w_1 + w_2 + \ldots + w_n = 1$
  - Trọng số w_i tỷ lệ với số quy mô của corpus i

Training & Test

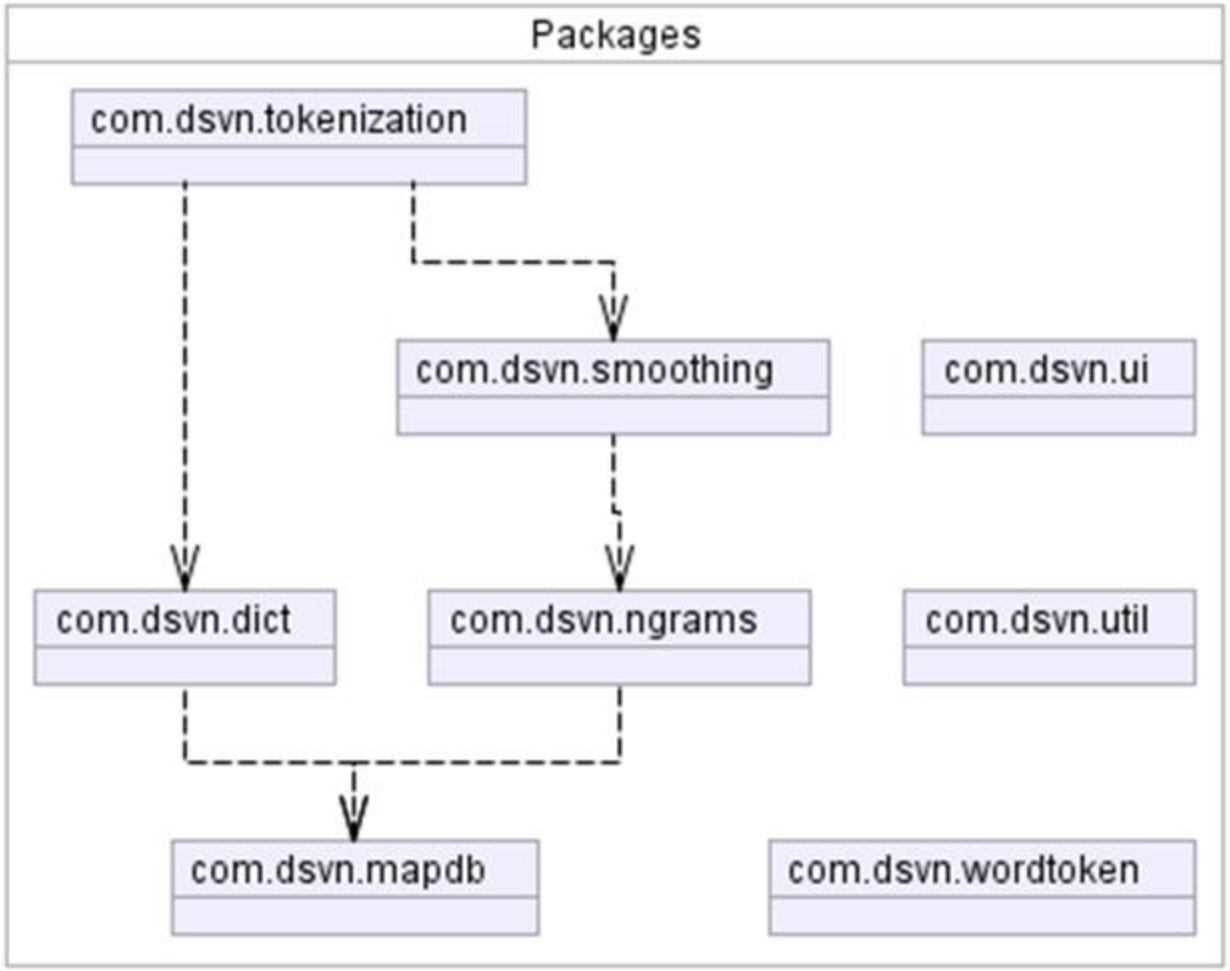
- Chia dữ liệu thành:
  - 80% train (học mô hình p_{KN})
  - 10% held-out (điều chỉnh tham số)
  - 10% test
    - Thử gán nhãn từ
      - sau đó có thể sửa lại = heuristic để làm tập training mới
    - Đánh giá: **perplexity**
      - pp càng nhỏ càng tốt

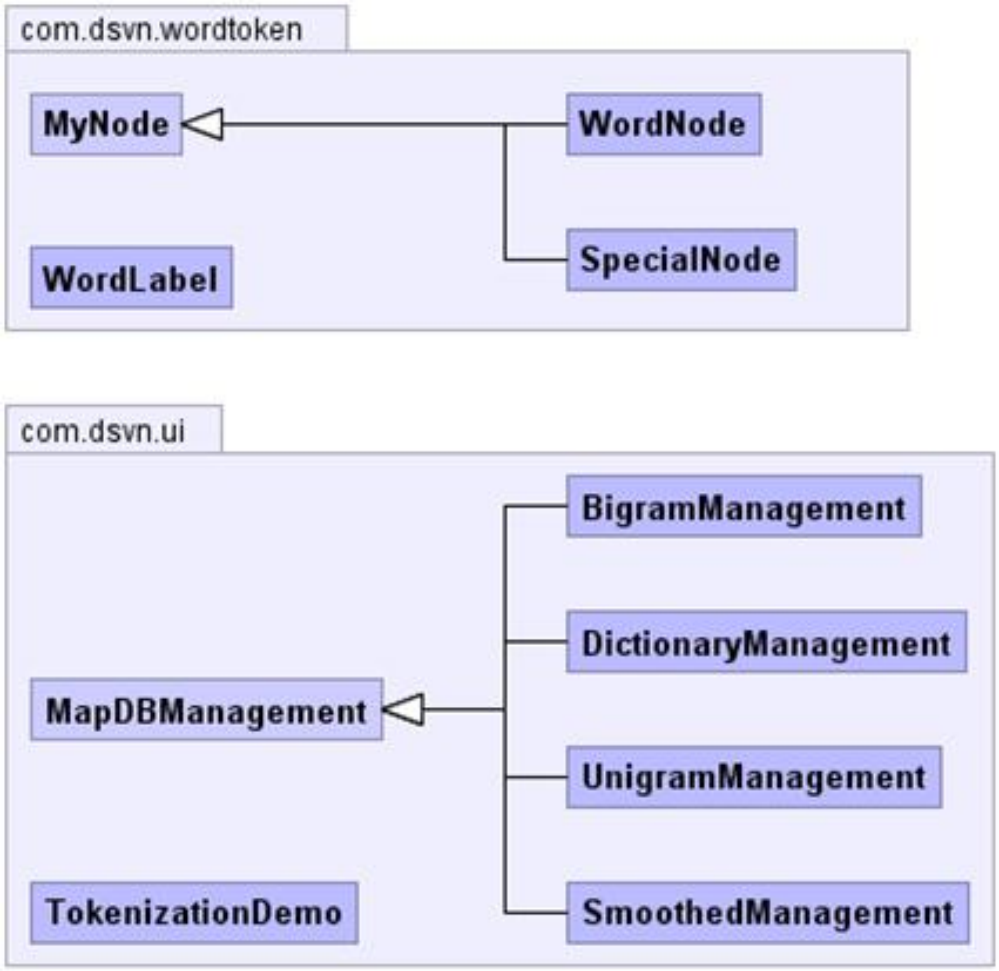
$$\begin{aligned}\text{Perplexity}(W) &= 2^{H(W)} \\ &= P(w_1 w_2 \ldots w_N)^{-\frac{1}{N}} \\ &= \sqrt[N]{\frac{1}{P(w_1 w_2 \ldots w_N)}} \\ &= \sqrt[N]{\prod_{i=1}^{N} \frac{1}{P(w_i | w_1 \ldots w_{i-1})}}\end{aligned}$$

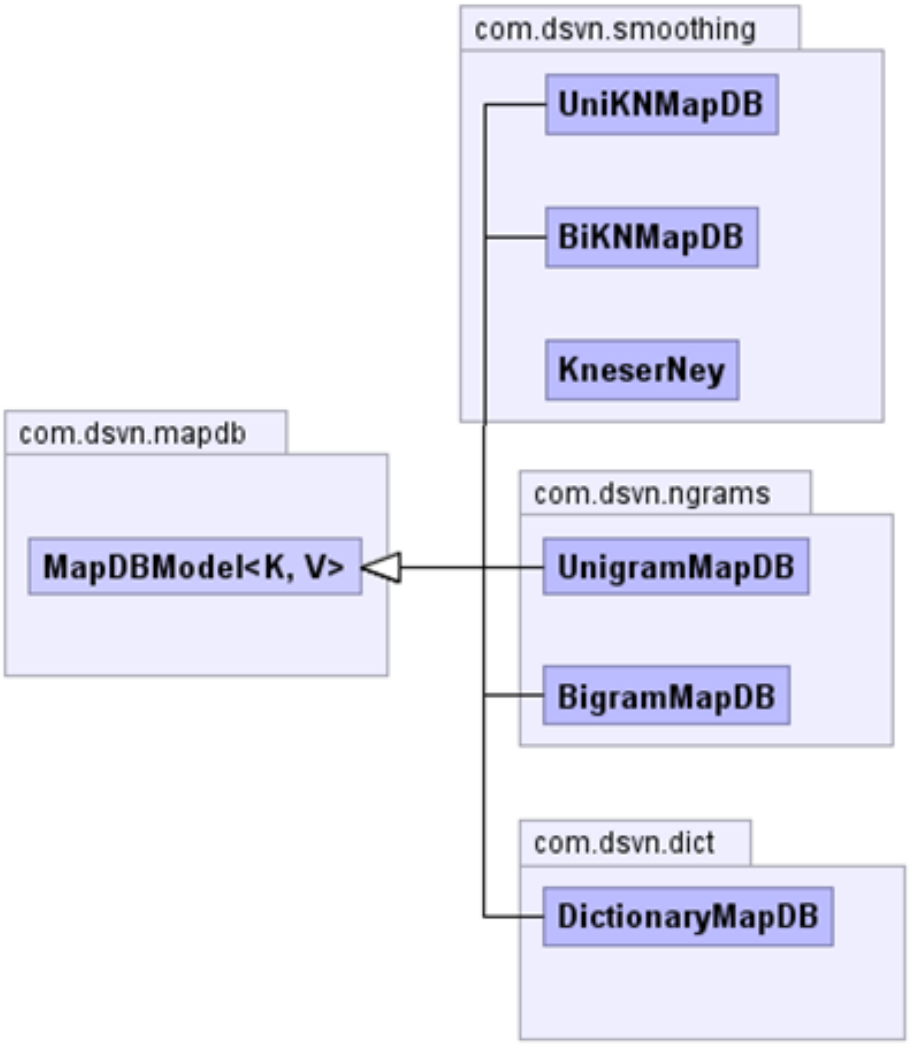
Giả thiết đầu vào – đầu ra

- Mỗi dòng là 1 câu
- Thêm 2 từ khóa START và BEGIN để đảm bảo phân bố xác suất
  - <BEGIN> Hôm nay trời đẹp <END>
- Thêm từ khóa <UNK> (xxx ?) cho các từ chưa có trong dictionary

Implementation




com.dsvn.wordtoken
MyNode
WordNode
WordLabel
SpecialNode
com.dsvn.ui
BigramManagement
DictionaryManagement
MapDBManagement
UnigramManagement
TokenizationDemo
SmoothedManagement


com.dsvn.smoothing
UniKNMapDB
BiKNMapDB
KneserNey
com.dsvn.mapdb
MapDBModel<K, V>
com.dsvn.ngrams
UnigramMapDB
BigramMapDB
com.dsvn.dict
DictionaryMapDB