Introduction

- Four papers, roughly classified to two different method types, are surveyed.
  - Word segmentation by unsupervised dictionary.
    - Ref1: An Iterative Algorithm to Build Chinese Language Models, Xiaoqiang Luo et al., ANLP ’96.
    - Ref2: Discovering Chinese Words from Unsegmented Text, Xianping Ge et al., SIGIR ’99 poster.
    - Ref3: Establishing Unsupervised Dynamic Word Segmentation Dictionary Based on Information Theory, Jun Gao et al.,
  - Grouping character pairs with statistical indicators.
    - Ref4: Chinese Word Segmentation without Using Lexicon and Handcrafted Training Data, Sun Maosung et al., ACL-COLING ’98
Word Segmentation - Ref1

- **Idea:**
  - chicken-and-egg problem
    Use word-based language model to segment sentences, but build a word-based LM require word segmentation.

- **Segmentation assumption:**
  - assume there is a word-based LM.

- **Segmentation**
  - use viterbi-like segmentation algorithm.
  - A sentence should be segmented into \( w_1, w_2, \ldots, w_k \) such that \( \log P(w_i | h_i) \) is maximized. Where \( h_i \) is the history word \( w_{i-1} \).
  - Use tri-gram model, therefore \( h_i = w_{i-2} w_{i-1} \).

---

Word Segmentation - Ref1

- **Segmentation algorithm**

1. \( S = C_1 C_2 \ldots C_{n-1} C_n \) each \( C_i \) \( 1 \leq i \leq n \)
   
   \( = (C_1 \ldots C_{x_1}) \ldots (C_{x_{i-1}} \ldots C_{x_i}) \ldots (C_{x_{m-1}} \ldots C_{x_m}) \)

   \( = w_1 w_2 \ldots w_n \)

   \( w_k = C_{x_{i-1}} \ldots C_{x_i} \) \( (k=1, 2, \ldots, m) \), \( x_0 = 0, x_m = n \)

2. \( G(S) = \{ (x_1, \ldots, x_m) : 1 \leq x_1 \leq \ldots \leq x_m, m \leq n \} \) is the set of all possible segmentations of sentence \( S \).

   \( g(S) \) = \( (x_1, \ldots, x_m) \) \( G(S) \), and assign a score to \( g(S) \) by

   \( L(g(S)) = \log P_g(w_1 \ldots w_m) = \sum_{i=1}^{m} \log P(w_i | h_i) \)

   \( g^* = \arg \max_{g \in G(S)} L(g(S)) \)

   \( = \arg \max_{g \in G(S)} \log P_g(w_1 \ldots w_m) \)

3. dynamic programming to get optimal segmentation \( g^* \)

   \( L(k) = \max_{1 \leq i \leq k} \{ L(i) + \log P(C_{i+1} \ldots C_k | h_i) \} \)
Learning Processing – Ref1

- Initialization:
  - splits the data into two parts, $T_1$ and $T_2$.
  - segment $T_1$ by a simple greedy algorithm, to construct a $LM_0$ with an initial vocabulary $V_0$. $i=1$

- Iteration:
  - $j=i \mod 2$
  - for each sentence $S$ in the set $T_j$
    1. segment it using $LM_{i-1}$
    2. for each unseen word in the optimal segmentation, increment its counter.
    3. let $A$=the set of unseen words with counter greater than $c$. Set $V_i=V_{i-1} \cup A$
    4. construct another $LM_i$ using the segmented set $T_j$ and $V_i$
  - $i=i+1$ and repeat

Word Segmentation – Ref2

- System Assumption:
  - There are a finite number of words of length $1, 2, ..., k$ (e.g. $k=4$)
  - Each word has an unknown probability of occurrence.
  - Words are independent of each other, i.e., any two words can occur together, governed only by their respective probability of occurrence.

- Segmentation Assumption:
  - Having a dictionary with all possible words and their occurrence probabilities.

- Segmentation
  - A sentence should be segmented into $w_1, w_2, ..., w_k$ such that $\prod p(w_i)$ is maximized.
  - Use zero-th order hidden Markov Model
  - Use vertibi-like algorithm.
Learning Processing - Ref2

- **Initialization:**
  - An imperfect dictionary with all possible words with equal occurrence probabilities (e.g., $10^{-12}$).

- **Iteration:**
  - Soft-counting algorithm:
    - For all possible segmentations of a sentence, compute their probability $p_i$.
    - Increase the word count of the words in that sentence by $\sum p_i / p_j$.
    - For example, given a sentence $C_1C_2C_3...C_n$ for any word $C_{j1}...C_{j2}$ inside the sentence, its count should be increased by $\sum p_i(C_{j1}...C_{j2}) / \sum p_i$, where
      - $\sum p_i(C_{j1}...C_{j2})$ is the sum of the likelihood of all possible segmentations of the substring to the left of $C_{j1}$.
      - $P(C_{j1}...C_{j2})$ is the estimate of the probability of word $C_{j1}...C_{j2}$.
      - $\sum p_i$ is the sum of the likelihood of all possible segmentations of the substring to the right of $C_{j2}$.
      - $\sum p_i$ is the normalizing constant, which is the sum of the likelihood of all the possible segmentations of this sentence. It is equal to $\sum p_i$.

Learning Processing - Ref2

- Use the soft-counting algorithm to update the imperfect dictionary and word probabilities.
- Complexity of the algorithm is $O(kIN)$, $k$ is the maximum word length, $I$ is the number of iterations (usually 5-10), $N$ is the size of the corpus.

- **Evaluation**
  - Definition: recall = $\frac{c}{n_1}$, precision = $\frac{c}{n_2}$, where $n_1$ is the number of segmented words, $n_2$ is the number of words of correct segmentation. $C$ is the number of words in common.
  - Recall/precision = 65.65%/71.91%, after post processing 97.72%/91.05%.
  - Post processing: removal of 20 single-character auxiliary words that occur together with other words.
Word Segmentation - Ref3

• Assumption:
  - sentence is independent from one another, while the word relationship in the sequence is dependent.

• variable distance word segmentation method
  \[ S = w_1w_2...w_q \] (where \( s_i = C_{i1}C_{i2}...C_{ik+1} \)) is corresponding to \( I_s = \{i_1,i_2,...i_q\} \) and
  \[ I_{max} = \arg \max (w) \],
  where \( \arg \max (w) = \max_{1 \leq k \leq L} P(s_k) = \max_{1 \leq k \leq L} P(w_{ki}w_{ki+1}...w_{k+1}) \)

• algorithm
  - calculate all possible Chinese character strings' prob. \( N_{init}(s) \) denote # of character string \( s \) in the corpus.
  \[ P_{init}(s) = N_{init}(s) / N_{total} \] , where \( N_{total} \)
  - use viterbi algorithm to calculate \( \Omega (w) \)?
  \[ \Omega (w_1w_2...w_L) = \max \{P(w_{L-1},w_{L+1}) \Omega (w_1w_2...w_{L-1})\} \]

Learning Processing - Ref3

• Initialization:
  - use VDWS method to segment corpus, and the result is put into WS system again.
  - Repeat the step N times, finally let the result be initializing word segmentation dictionary(WSD).

• Substantiating WSD:
  - on the basis of current WSD, using VDWS on the same time, segment new corpus.
  - according to the ratio between WSD and text scale, also, the value of occurring probabilities of certain word, assign weight to WSD and dictionary obtained from VDWS.
  - regard the result of segmentation as one part of the WSD, adding it into the original WSD by means of increasing the value of word prob. Occurring the WSD originally and augmenting the new words.
**Word Segmentation - Ref4**

- **Idea:**
  - use mutual information (MI), difference of t-score to find word boundary.
  - Mutual information
    \[ MI(x:y) = \log \frac{P(x,y)}{P(x)P(y)} \]
  - t-score of y relevant to characters x and z (ie. 'xyz')
    \[ ts_{x,z}(y) = \frac{P(x|y)P(y|x)}{\sqrt{\text{var}(P(x|y)) \cdot \text{var}(P(y|x))}} \]
  - difference of t-score between x and y (given 'vxyw')
    - \[ dts(x:y) = ts_{v,y}(x) - ts_{x,w}(y) \]
    - \[ dts(x:y) > 0 \] then x,y tends to be bound
    - \[ dts(x:y) < 0 \] then x,y tends to be separated

\[ \begin{align*}
(1) & \quad ts_{v,y}(x) > 0 \quad ts_{x,w}(y) < 0 \\
(2) & \quad ts_{v,y}(x) < 0 \quad ts_{x,w}(y) > 0 \\
(3a) & \quad ts_{v,y}(x) > 0 \quad ts_{x,w}(y) > 0 \\
(3b) & \quad ts_{v,y}(x) < 0 \quad ts_{x,w}(y) < 0
\end{align*} \]

**Result:**
- 濟合 將 是 對 目前 世界 經 濟 趨 勢 的 一 個 適 當 回答
- 經 合 將 是 對 目 前 世 界 經 濟 趨 勢 的 一 個 適 當 回答
- 法國 網 球 公 開 賽 今 天 在 巴 黎 西 郊 拉 開 戰 幕
- 法 國 網 球 公 開 賽 今 天 在 巴 黎 西 郊 拉 開 戰 幕

**connect**  **break**
Result: 法國 網球 公開賽 今天 在 巴黎 西郊 拉開 戰幕

**Word Segmentation — Ref4**

- **Local maximum and local minimum of dts (given ‘vxvw’)**
  - local maximum : if dts(x:y)>dts(v:x) and dts(x:y)>dts(y:w)
    - height of the local maximum
    \[ h(dts(x:y)) = \min \{ dts(x:y) - dts(v:x), dts(x:y) - dts(y:w) \} \]
  - local minimum : if dts(x:y)<dts(v:x) and dts(x:y)<dts(y:w)
    - height of the local minimum
    \[ h(dts(x:y)) = \min \{ dts(v:x) - dts(x:y), dts(y:w) - dts(x:y) \} \]

- **Second local maximum and second local minimum of dts (given ‘vxyzw’)**
  - dts(y:z) is said to be the "right" second local maximum of dts(x:y) if dts(y:z)>dts(v:x) and dts(y:z)>dts(z:w)
    - the distance between local maximum and the second local maximum is
    \[ \text{dis(locmax,y:z)} = dts(x:y) - dts(y:z) \]
  - dts(y:z) is said to be the "right" second local minimum of dts(x:y) if dts(y:z)<dts(v:x) and dts(y:z)<dts(z:w)
    - the distance between local minimum and the second local minimum is
    \[ \text{dis(locmin,y:z)} = dts(y:z) - dts(x:y) \]
  - "left" second local maximum/minimum are defined in this way.
Algorithm - Ref4

Given an input sentence $S$, let
- $\mu_{mi}$: the mean of $mi$ of all locations in $S$
- $\sigma_{mi}$: standard deviation of $mi$ of all locations in $S$
- $\mu_{dts}$: the mean of $dts$ of all locations in $S$
- $\sigma_{dts}$: standard deviation of $dts$ of all locations in $S$

- Region A $dts(x:y) > \mu_{dts}$
  - Region B $0 < dts(x:y) \leq \sigma_{dts}$
  - Region C $-\sigma_{dts} < dts(x:y) \leq 0$
  - Region D $dts(x:y) \leq -\sigma_{dts}$
- Region a $mi(x:y) > \mu_{mi} + \sigma_{mi}$
- Region b $\mu_{mi} < mi(x:y) \leq \mu_{mi} + \sigma_{mi}$
- Region c $\mu_{mi} - \sigma_{mi} < mi(x:y) \leq \mu_{mi}$
- Region d $mi(x:y) \leq \mu_{mi} - \sigma_{mi}$

The first round for $S$

For any location $(x:y)$ in $S$, do:

1. In cases that $dts(x:y)$ falls into:
   - 1.1 $dts(x:y)$ in Region Aa or Ba or Ca or Db or Ab: mark $(x:y)$ 'bound'
   - 1.2 $dts(x:y)$ in Region Ad or Bd or Cd or Dd or Dc: mark $(x:y)$ 'separated'
   - 1.3 $dts(x:y)$ in Region Ac or Cb: if $dts(x:y)$ is local maximum then
     - if $h(dts(x:y)) > \delta_1$ then mark $(x:y)$ 'bound' else '?'
     - if $d(dts(x:y)) > \xi_1$ then mark $(x:y)$ 'separated' else '?'
   - 1.4 $dts(x:y)$ in Region Bc or Db: if $dts(x:y)$ is local maximum then
     - if $h(dts(x:y)) > \delta_2$ then mark $(x:y)$ 'bound' else '?'
     - if $d(dts(x:y)) > \xi_2$ then mark $(x:y)$ 'separated' else '?'
   - 1.5 $dts(x:y)$ in Region Cc: if $(dts(x:y)$ is local maximum) and $h(dts(x:y)) > \delta_3$ then mark $(x:y)$ 'bound' else '?'
     - if $d(dts(x:y)) > \xi_3$ then mark $(x:y)$ 'separated' else '?'
   - 1.6 $dts(x:y)$ in Region Bb: if $dts(x:y)$ is local maximum then mark $(x:y)$ 'bound' else '?'
     - if $(dts(x:y)$ is local minimum and $(d(dts(x:y)) > \xi_3)$ then mark $(x:y)$ 'separated' else '?'

For $(x:y)$ unmarked so far, mark it as '?' except that:

- if $dts(x:y)$ is the second local maximum then
  - if $d(dts(x:y)) < 0.5 x lrmin(loc,x:y)$ then mark $(x:y)$ 'bound' else '
  - if $(x:y)$ is the right second local max or mark '' if $(x:y)$ is the left second local max
- if $dts(x:y)$ is the second local minimum then
  - if $d(dts(x:y)) < 0.5 x lrmin(loc,x:y)$ then mark $(x:y)$ 'bound' else '
  - if $(x:y)$ is the right second local min or mark '' if $(x:y)$ is the left second local min
**Algorithm - Ref4**

The second round for S

if \((x:y)\) is marked '?' then if \(m(x:y) \geq \theta\) then mark \((x:y)\) 'bound'
else 'separated'

if \((x:y)\) is marked ' ' then the status of \((x:y)\) follows that of the adjacent location on the left side

if \((x:y)\) is marked ' ' then the status of \((x:y)\) follows that of the adjacent location on the right side

(The contents \(S_1, S_2, S_3, S_1, S_2, S_3\) are \(S_1 < S_2 < S_3\); \(S_1 < S_2 < S_3\) and \(S = 2.5\))

---

**Sample Difference between two performance value**

**Definition 1:**
\[
\text{recall} = \frac{\# \text{ of locations being correctly marked}}{\# \text{ of locations in text}}
\]

**Definition 2:**
\[
\text{recall} = \frac{C_1}{n_1}, \quad \text{precision} = \frac{C_2}{n_2}, \quad \text{where } n_1 \text{ is the number of segmented words, } n_2 \text{ is the number of words of correct segmentation.}
\]

\(C\) is the number of words in common.

<table>
<thead>
<tr>
<th></th>
<th>Def 1</th>
<th>Def 2</th>
<th>Def 2</th>
<th>Precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>Correct</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wrong</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wrong</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>