Locating and Filling Missing Words in Sentences Based on Language Models

Presented By: Tianlong Song & Zhe Wang
Department of Electrical & Computer Engineering
Michigan State University
East Lansing, Michigan 48824, USA.
Email: {songtia6,wangzh34}@msu.edu

Apr 29, 2015

Outline

- **Motivation**
- **Problem Statement**
- **Missing Word Location**
- **Missing Word Filling**
- **Experimental Results**
- **Conclusions**

Motivation

- **Why words missing?**
  - Speech recognition under noisy environments
  - Sentence completion problems in SAT tests
- **Challenges**
  - Unawareness of missing word locations
  - Large vocabulary size
  - Multiple missing words in a sentence

Problem Statement

- **Assumptions**
  - Only one missing word in a sentence
  - The missing word is neither the first nor the last word
- **Objectives**
  - Locate the missing word
  - Fill the missing word

Locating the Missing Word: N-gram Model

- **Definitions**
  - $C(w_1, w_2)$: # of occurrences of bigram pattern (w_1, w_2)
  - $C(w_1, w, w_2)$: # of occurrences of trigram pattern (w_1, w, w_2)
  - $D(w_1, w_2) = \sum_{w \in V} C(w_1, w, w_2)$: # of occurrences of the pattern, where there is exactly one word between w_1 and w_2
- **Illustration**
  - Consider location l in an incomplete sentence

$$..., w_{l-2}, w_{l-1}, w_l, w_{l+1}, ... \tag{1}$$

  - $C(w_{l-1}, w_l)$: negative votes for missing word at location l
  - $D(w_{l-1}, w_l)$: positive votes for missing word at location l

Locating the Missing Word: N-gram Model

- **Score indicating a missing word at location l**

$$S_l = \frac{D(w_{l-1}, w_l)^{1+\gamma}}{C(w_{l-1}, w_l) + D(w_{l-1}, w_l)} - \frac{C(w_{l-1}, w_l)^{1+\gamma}}{C(w_{l-1}, w_l) + D(w_{l-1}, w_l)} \quad (2)$$

- **Illustration**
  - $\gamma = 0$: percentage of positive votes v.s. percentage of negative votes
  - $\gamma > 0$: break ties (e.g., 80 positive v.s. 20 negative for location A, and 8 positive v.s. 2 negative for location B)

Locating the Missing Word: N-gram Model

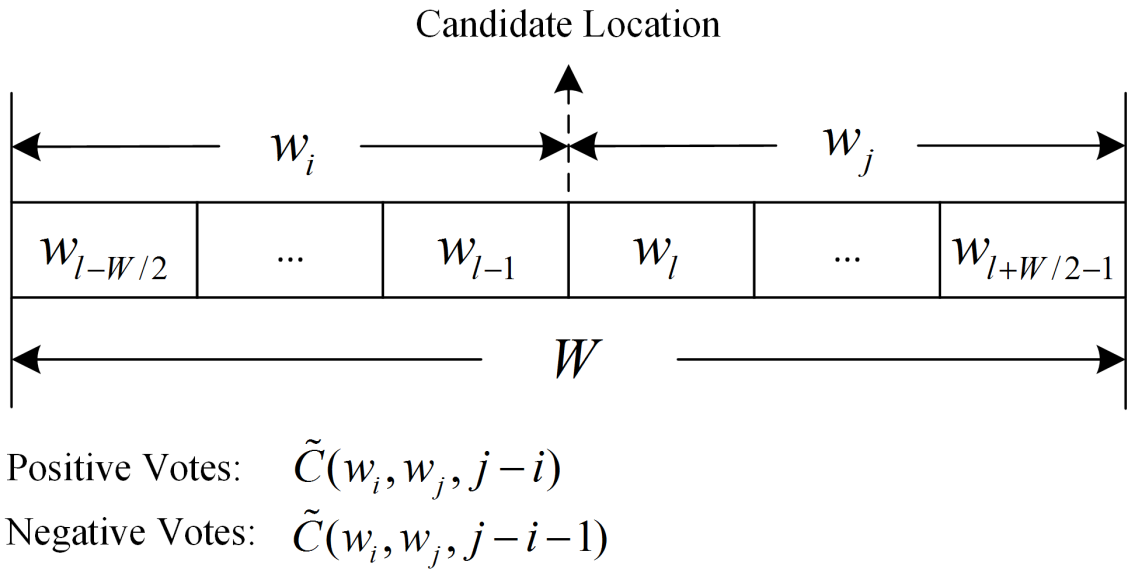
- **Missing word location**

$$\hat{l} = \arg\max_{1 \le l \le L-1} S_l \tag{3}$$

- **Zero positive & zero negative?**
  - Assign a zero score for any candidate location with zero positive votes and zero negative votes!

Locating the Missing Word: WDS

- **Word distance statistics (WDS)**
  - $\widetilde{C}(w_1, w_2, m)$: # of occurrences of the pattern, where there is exactly m words between w_1 and w_2
- **Graphical illustration**



Locating the Missing Word: WDS

- **For any** $l - W/2 \leq i \leq l - 1$ **and** $l \leq j \leq l + W/2 - 1$

$$S_l(i,j) = \frac{\widetilde{C}(w_i, w_j, j-i)^{1+\gamma}}{\widetilde{C}(w_i, w_j, j-i) + \widetilde{C}(w_i, w_j, j-i-1)} - \frac{\widetilde{C}(w_i, w_j, j-i-1)^{1+\gamma}}{\widetilde{C}(w_i, w_j, j-i) + \widetilde{C}(w_i, w_j, j-i-1)} \tag{4}$$

- **Missing word location**

$$\hat{l} = \arg\max_{1 \leq l \leq L-1} \sum_{l-\frac{W}{2} \leq i \leq l-1} \sum_{l \leq j \leq l+\frac{W}{2}-1} v(i,j) S_l(i,j) \tag{5}$$

Locating the Missing Word: WDS

- **How to choose weights $v(i, j)$?**
  - Empirical setting: monotonically decreasing with respect to $|j - i|$
  - Machine learning: logistic regression
- **Little gain by machine learned weights**
  - Missing word location accuracy is dominated by the two words immediately adjacent to candidate locations
  - Weight trend is straightforward and clear, so a good empirical setting would probably be already good enough

Filling the Missing Word

- **Challenges?**
  - Candidate word space may be as large as the entire vocabulary
  - Reduce candidate word space first: consider high-freq words only
- **Further reduction on candidate word space**

Word Group	A	B	C
Coverage	35.20%	86.75%	88.35%

Table 1: Percentages of different word groups in terms of covering the ground-truth words to be filled.

  - Group A: any word w, s.t. trigram (w_{l-1}, w, w_l) occurs before
  - Group B: any word w, s.t. (w_{l-1}, w) or (w, w_l) occurs before
  - Group C: further loosen the constraint

Filling the Missing Word

- **Five conditional probabilities**

	Weight	Definition
P_1	1.0	$P(w\|w_{l-1} * w_l) = \frac{Cnt(w_{l-1}ww_l)}{Cnt(w_{l-1}*w_l)}$
P_2	0.5	$P(w\|w_{l-2}w_{l-1}*) = \frac{Cnt(w_{l-2}w_{l-1}w)}{Cnt(w_{l-2}w_{l-1})}$
P_3	0.5	$P(w\| * w_lw_{l+1}) = \frac{Cnt(ww_lw_{l+1})}{Cnt(w_lw_{l+1})}$
P_4	0.25	$P(w\|w_{l-1}*) = \frac{Cnt(w_{l-1}w)}{Cnt(w_{l-1})}$
P_5	0.25	$P(w\| * w_l) = \frac{Cnt(ww_l)}{Cnt(w_l)}$

- **Most probable missing word**

$$\hat{w} = \arg\max_{w \in B} \sum_{1 \le i \le 5} v_i P_i \tag{6}$$

Extra Efforts

- **POS Tagging: Failed Trial**
  - Motivation: infinite word sequences v.s. finite grammar rules
  - There are errors in POS tagging for incomplete sentences, especially for the word immediately after the removed word
  - POS tags for incomplete sentences are misleading, since the tagger still produces the most probable tag path, which always contains strong connection between neighboring tags, even at missing word locations. This leads to difficulties in the "anomaly" detection at missing word locations by our approaches.

Experimental Results

- **The Complete Data:**
  - $30,301,028$ complete sentences
  - The average sentence length is approximately 25s
  - The vocabulary size is $2,425,337$
- **High Frequency Words**
  - $14,216$ words that have occurred in at least 0.1% of total sentences
  - $58,417,315$ words are labeled as 'UNKA'
- **Cross-Validation**
  - 80% of the data set is applied as the TRAIN set for training
  - 20% of the data set is applied as the DEV set for testing

Accuracy of Missing Word Location

- **The estimation accuracy of the missing word locations for the two proposed approaches, N-gram and WDS.**

	Top 1	Top 2	Top 3	Top 5	Top 10
Chance	4%	8%	12%	20%	40%
N-gram	51.47%	63.70%	71.00%	80.26%	91.54%
WDS	52.06%	64.50%	71.76%	80.91%	91.93%

Accuracy v.s. Training Data Size

- Changes of the accuracy (N-gram approach) v.s. size of the training data. The parameter γ is set to be 0.01

Number	0.1	0.5	1	2	5	30
Accuracy	36.11%	41.06%	42.56%	43.95%	48.32%	51.47%

- Larger size of training data leads to higher classification accuracy

Accuracy of Missing Word Filling

- Accuracies of filling the missing word given the location:

	Top 1	Top 2	Top 3	Top 5	Top 10
Accuracy	32.15%	41.49%	46.23%	52.02%	59.15%

- When considering only the candidate word with the highest score, the accuracy is 32.15%. After taking more candidates into consideration, that probability that the correct word is covered in the returned list is increased to be above 50%

Discussion & Conclusion

- **Discussion**: Challenges and Differences with previous works
  - Missing place unknown
  - Huge number of candidate words
- **Conclusion**:
  - Two approaches, which are based on N-gram models and word distance statistics (WDS), respectively.
  - The proposed approaches achieve much higher accuracies than chance.

Thank you!

Questions?