Natural Language Processing
N-Gram Language Models

Ahmad Rio Adriansyah, M.Si.

STT Terpadu Nurul Fikri

*diadaptasi dari slide Raymond J. Mooney (Univ. Texas)
dan slide Graham Neubig (NAIST)

Language Models

- Formal grammars (e.g. regular, context free) sulit untuk digunakan sebagai model kalimat yang baku dari sebuah bahasa alami.
- Untuk NLP, lebih berguna model ***probabilistik*** dari sebuah bahasa (yang berisi nilai peluang dari masuknya sebuah string dalam suatu bahasa).
- Untuk menentukan distribusi peluang yang tepat, peluang seluruh kalimat harus berjumlah 1.

Uses of Language Models

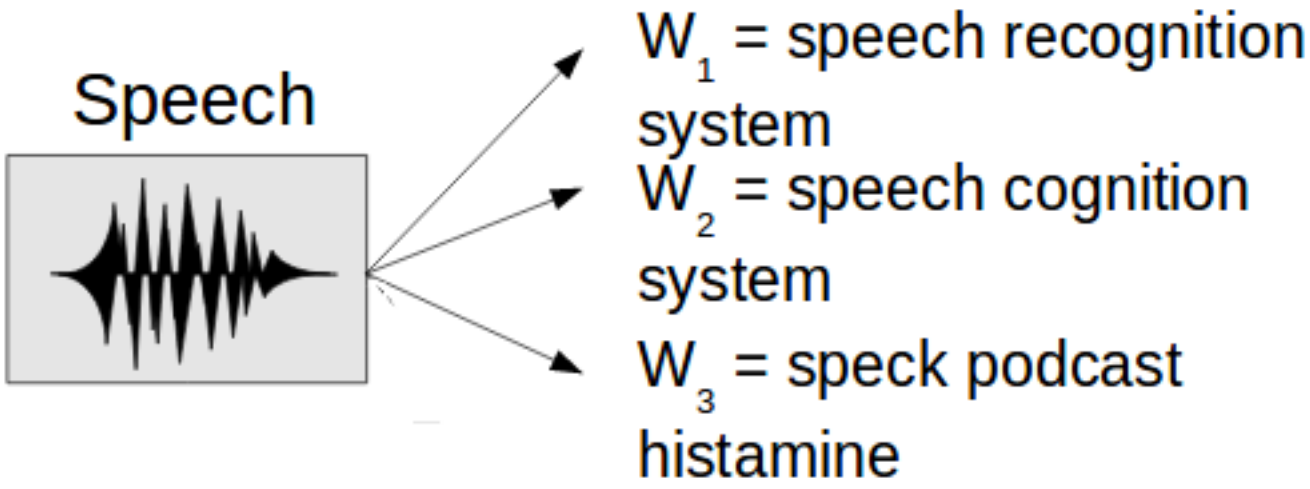
- Speech recognition
  - "Papan tulis warna putih" lebih berbentuk seperti kalimat dibandingkan "Pak, bantu Lis warna putih"
- OCR & Handwriting recognition
  - Kalimat dengan peluang tinggi, lebih tepat
- Machine translation
  - Kalimat yang lebih wajar (muncul di keadaan yang sebenarnya) merupakan terjemahan yang lebih baik.
- Generation
  - Kalimat yang peluangnya lebih tinggi merupakan pembangkitan bahasa alami yang lebih baik.
- Context sensitive spelling correction
  - "Their are problems wit this sentence."

Completion Prediction

- Model bahasa juga digunakan untuk memprediksi kata untuk mengisi kalimat yang tidak lengkap
  - Please turn off your cell _____
  - Your program does not _____
- Sistem *Predictive text input* dapat menduga apa yang sedang diketik oleh user dan memberikan pilihan kata mana yang akan digunakan.

Model Bahasa Probabilistik

Mana yang paling baik hasil transkripsinya?



Model Bahasa Probabilistik

No	Nama / w	Transkripsi	Peluang / P(w)
1	W_1	Speech recognition system	$4.021 * 10^{-3}$
2	W_2	Speech cognition system	$8.932 * 10^{-4}$
3	W_3	Speck podcast histamine	$2.432 * 10^{-7}$

N-Gram Models

- Peluang dari kata tertentu yang diawali dengan konteks sebelumnya.
  - P(phone | Please turn off your cell)
- Banyaknya parameter yang dibutuhkan naik secara eksponensial terhadap banyaknya kata pada konteks sebelumnya.
- Model N-gram menggunakan N−1 kata pada konteks sebelumnya.
  - Unigram: P(phone)
  - Bigram: P(phone | cell)
  - Trigram: P(phone | your cell)

N-Gram Models

- ***Markov assumption*** adalah praduga bahwa perilaku yang akan datang adalah sebuah sistem dinamik yang hanya bergantung kepada sejarah perilaku yang terjadi sebelumnya.
- Secara khusus, pada ***Markov model orde ke-k***, keadaan berikutnya hanya tergantung kepada k buah perilaku yang terjadi sebelumnya, jadi model N-gram adalah Markov model orde ke-(N-1).

N-Gram Model Formulas

- Word sequences

$$w_1^n = w_1 \ldots w_n$$

- Chain rule of probability

$$P(w_1^n) = P(w_1)P(w_2|w_1)P(w_3|w_1^2)\ldots P(w_n|w_1^{n-1}) = \prod_{k=1}^{n} P(w_k|w_1^{k-1})$$

- Bigram approximation

$$P(w_1^n) = \prod_{k=1}^{n} P(w_k|w_{k-1})$$

- N-gram approximation

$$P(w_1^n) = \prod_{k=1}^{n} P(w_k|w_{k-N+1}^{k-1})$$

Estimating Probabilities

- Peluang bersyarat dari N-gram bisa dihitung dari teks yang ada berdasarkan frekuensi relatif dari barisan katanya.

Bigram: $$P(w_n|w_{n-1}) = \frac{C(w_{n-1}w_n)}{C(w_{n-1})}$$

N-gram: $$P(w_n|w_{n-N+1}^{n-1}) = \frac{C(w_{n-N+1}^{n-1}w_n)}{C(w_{n-N+1}^{n-1})}$$

- Untuk mendapat model peluang yang konsisten, gunakan simbol di awal (<s>) dan di akhir (</s>) ke setiap kalimat dan dianggap sebagai sebuah kata.

Generative Model & MLE

- Model N-gram dapat dilihat sebagai automata yang membuat kalimat secara probabilistik.

  Initialize sentence with N−1 <s> symbols
  Until </s> is generated do:
  Stochastically pick the next word based on the conditional probability of each word given the previous N −1 words.

- Frekuensi relatif pada slide sebelumnya merupakan ***maximum likelihood estimates*** (MLE) karena memaksimalkan peluang model M akan menghasilkan corpus training T.

Contoh

- Kita menggunakan database dari Berkeley Restauran Project (speech based restaurant consultant)
- User bertanya tentang restoran di Berkeley, sistemnya akan memunculkan informasi dari database restoran lokal.

- Jurafsky dan Martin. Speech and Language Processing. 1999

Contoh

- I’m looking for Cantonese food.
- I’d like to eat dinner someplace nearby.
- Tell me about Chez Panisse.
- Can you give me a listing of the kinds of food that are available?
- I’m looking for a good place to eat breakfast.
- I definitely do not want to have cheap Chinese food.
- When is Caffe Venezia open during the day?
- I don’t wanna walk more than ten minutes.

Contoh

- Bigram grammar dari kata yang mengikuti kata ‘eat’

eat on	.16	eat Thai	.03
eat some	.06	eat breakfast	.03
eat lunch	.06	eat in	.02
eat dinner	.05	eat Chinese	.02
eat at	.04	eat Mexican	.02
eat a	.04	eat tomorrow	.01
eat Indian	.04	eat dessert	.007
eat today	.03	eat British	.001

- Beberapa bigram grammar tambahan

<s> I	.25	I want	.32	want to	.65	to eat	.26	British food	.60
<s> I’d	.06	I would	.29	want a	.05	to have	.14	British restaurant	.15
<s> Tell	.04	I don’t	.08	want some	.04	to spend	.09	British cuisine	.01
<s> I’m	.02	I have	.04	want thai	.01	to be	.02	British lunch	.01

Contoh

Akan dihitung peluang kalimat “I want english food” jika diketahui bahwa :

- Peluang muncul kata ‘I’ di awal kalimat adalah 0.25 (P(i | <s>) = 0.25)
- Peluang kata ‘want’ muncul setelah kata ‘I’ adalah 0.32
- Peluang kata ‘english’ muncul setelah kata ‘want’ adalah 0.0011
- Peluang kata ‘food’ muncul setelah kata ‘english’ adalah 0.5
- Peluang kata ‘food muncul diakhir kalimat adalah 0.68

- P(<s> i want english food </s>)

  = P(i | <s>) P(want | i) P(english | want) P(food | english) P(</s> | food)

  = .25 x .32 x .0011 x .5 x .68 = .000031

Contoh

- Dengan cara yang sama, peluang kalimat “I want chinese food”
- P(<s> i want chinese food </s>)

  = P(i | <s>) P(want | i) P(chinese | want)

  P(food | chinese) P(</s> | food)

  = .25 x .32 x .0065 x .52 x .68 = .00019

Contoh Trigram

- Dari corpus Europarl, banyaknya (counts for) trigram dan peluang katanya

the green (total: 1748)

word	c.	prob.
paper	801	0.458
group	640	0.367
light	110	0.063
party	27	0.015
ecu	21	0.012

the red (total: 225)

word	c.	prob.
cross	123	0.547
tape	31	0.138
army	9	0.040
card	7	0.031
,	5	0.022

the blue (total: 54)

word	c.	prob.
box	16	0.296
.	6	0.111
flag	6	0.111
,	3	0.056
angel	3	0.056

- 225 trigram pada Europarl dimulai dengan kata ‘the red’
- 123 diantaranya dilanjutkan dengan kata ‘cross’
- Jika seseorang mengetik ‘the red…’, peluangnya bahwa yang dimaksud adalah ‘the red cross’ sebesar 0.547 (maximum likelyhood probability)

Train and Test Corpora

- Model bahasa harus ditraining ke corpus teks yang besar agar mendapatkan nilai parameter yang baik
- Model bisa dievaluasi berdasarkan kemampuannya untuk memprediksi peluang test corpus yang berbeda dari training corpus. (jika ditest ke training corpus akan memberikan nilai yang optimistically biased)
- Idealnya, training dan test corpus merepresentasikan data sesungguhnya
- Kadang diperlukan untuk mengadaptasi model umum ke ***in-domain*** data dengan menambahkan corpus berbobot tinggi ke data trainingnya

Unknown Words

- Bagaimana menangani kata pada test corpus yang tidak muncul pada training corpus? Kata-kata seperti itu disebut sebagai kata ***out of vocabulary*** (OOV)?
- Buat model training yang termasuk simbol kata untuk kata yang tidak diketahui (<UNK>).
  - Pilih satu buah kamus (vocabulary) dan ganti kata pada training corpus selain kata dalam kamus tersebut dengan <UNK>.
  - Ganti kemunculan pertama dari setiap kata pada training corpus dengan <UNK>.

Evaluation of Language Models

- Idealnya, model dievaluasi penggunaannya menggunakan aplikasi yang akan digunakan (***extrinsic***, ***in vivo***)
  - Realistic
  - Expensive
- Dievaluasi menggunakan model test corpus (***intrinsic***).
  - Less realistic
  - Cheaper
- Memastikan bahwa setidaknya satu kali evaluasi intrinsik berkorelasi dengan yang ekstrinsik

Perplexity

- Mengukur seberapa baik model untuk mendekati test data dapat menggunakan perplexity atau cross entropy.
- Peluang dari model yang diterapkan ke test corpus
- Dinormalisasi terhadap banyak kata pada corpus dan diinverskan

$$PP(W) = \sqrt[N]{\frac{1}{P(w_1 w_2 \ldots w_N)}}$$

- Mengukur *weighted average branching factor* dalam memprediksi kata selanjutnya (lower is better).

Cross Entropy

- Cross entropy dari kata W adalah

$$H(W) = 1/n \log (P(W_1^n))$$

- Sehingga perplexity dapat dihitung menggunakan

$$PP(W) = 2^{H(W)}$$

Sample Perplexity Evaluation

- Model ditraining ke 38 juta kata dari Wall Street Journal (WSJ) menggunakan kamus berisi 19,979 kata.
- Dievaluasi terhadap himpunan dari 1,5 juta kata yang berbeda dari WSJ.

	Unigram	Bigram	Trigram
Perplexity	962	170	109

Smoothing

- Urutan kata pada kalimat ada banyak kombinasinya yang mungkin sehingga bisa jadi kalimat yang langka (tapi bukan kalimat yang tidak valid) tidak muncul pada training. MLE secara salah memberikan nilai 0 ke banyak parameter (sparse data).
- Jika ada kombinasi baru (yang tidak pernah muncul di training) muncul waktu testing, maka peluangnya diberikan nol dan seluruh barisan kata tersebut peluangnya nol (infinite perplexity).
- Dalam prakteknya, parameternya dibuat ***smoothed*** (atau ***regularized***) untuk memberikan nilai peluang ke kejadian yang tidak pernah dilihat
  - Memberi nilai peluang ke kejadian yang tidak pernah dilihat harus diimbangi dengan mengurangi nilainya dari yang pernah dilihat (***discounting***) agar distribusi peluangnya tetap 1.

Laplace (Add-One) Smoothing

- "Menipu" training data seakan akan setiap kemungkinan N-gram pernah muncul tepat sekali dan menyesuaikan nilai peluangnya.

Bigram: $$P(w_n|w_{n-1}) = \frac{C(w_{n-1}w_n)+1}{C(w_{n-1})+V}$$

N-gram: $$P(w_n|w_{n-N+1}^{n-1}) = \frac{C(w_{n-N+1}^{n-1}w_n)+1}{C(w_{n-N+1}^{n-1})+V}$$

dimana V adalah banyaknya (N−1)-grams yang mungkin (ukuran kamus dari model bigramnya).

Cenderung memberi nilai terlalu banyak ke kejadian yang tidak pernah dilihat, bisa disesuaikan dengan menambahkan variabel δ dengan nilai $0<\delta<1$ (dinormalisasi menggunakan δV, bukan V).

Advanced Smoothing

- Banyak teknik dikembangkan untuk meningkatkan smoothing dari model bahasa, diantaranya :
  - Good-Turing
  - Interpolation
  - Backoff
  - Kneser-Ney
  - Class-based (cluster) N-grams

Model Combination

- As N increases, the power (expressiveness) of an N-gram model increases, ***but*** the ability to estimate accurate parameters from sparse data decreases (i.e. the smoothing problem gets worse).
- A general approach is to combine the results of multiple N-gram models of increasing complexity (i.e. increasing N).

Interpolation

- Linearly combine estimates of N-gram models of increasing order.

Interpolated Trigram Model:

$$\hat{P}(w_n|w_{n-2,}w_{n-1}) = \lambda_1 P(w_n|w_{n-2,}w_{n-1}) + \lambda_2 P(w_n|w_{n-1}) + \lambda_3 P(w_n)$$

Where: $\sum_i \lambda_i = 1$

- Learn proper values for λ_i by training to (approximately) maximize the likelihood of an independent ***development*** (a.k.a. ***tuning***) corpus.

Backoff

- Only use lower-order model when data for higher-order model is unavailable (i.e. count is zero).
- Recursively back-off to weaker models until data is available.

$$P_{katz}(w_n|w_{n-N+1}^{n-1}) = \begin{cases} P*(w_n|w_{n-N+1}^{n-1}) & \text{if } C(w_{n-N+1}^{n})>1 \\ \alpha(w_{n-N+1}^{n-1})\, P_{katz}(w_n|w_{n-N+2}^{n-1}) & \text{otherwise} \end{cases}$$

Where P* is a discounted probability estimate to reserve mass for unseen events and α's are back-off weights (see text for details).

A Problem for N-Grams: Long Distance Dependencies

- Many times local context does not provide the most useful predictive clues, which instead are provided by ***long-distance dependencies***.
  - Syntactic dependencies
    - "The ***man*** next to the large oak tree near the grocery store on the corner **is** tall."
    - "The ***men*** next to the large oak tree near the grocery store on the corner **are** tall."
  - Semantic dependencies
    - "The ***bird*** next to the large oak tree near the grocery store on the corner **flies** rapidly."
    - "The ***man*** next to the large oak tree near the grocery store on the corner **talks** rapidly."
- More complex models of language are needed to handle such dependencies.

Summary

- Model bahasa memberikan nilai peluang ke rangkaian kata (kalimat) yang merupakan anggota suatu bahasa
- Model tersebut berguna sebagai komponen dari banyak sistem NLP seperti ASR, OCR, dan MT.
- Simple N-gram model mudah ditraining ke unsupervised corpora dan memberikan perkiraan likelihood dari kalimat
- MLE memberikan nilai yang kurang akurat pada model yang ditraining di sparse data.
- Teknik smoothing menyesuaikan perkiraan parameter ke kejadian yang tidak pernah muncul sebelumnya (tapi mungkin untuk muncul)

Tugas

- Cari beberapa artikel yang sejenis (berita/artikel majalah/puisi/prosa/dokumen hukum atau yang lainnya)
- Hitung wordcountnya >100.000 kata (50~100 artikel)
- Jika ada sumbernya, tapi kesulitan untuk mengambilnya, bisa didiskusikan cara untuk crawling secara otomatis
- Hitung unigram, bigram, dan trigramnya