

بسم الله الرحمن الرحيم

پردازش زبانهای طبیعی

جلسه ۸

احسان الدین عسگری

۱۴۰۳ فروردین

<http://language.ml/>

دانشکده مهندسی کامپیوتر

آزمایشگاه پردازش هوشمند متن و زبان و علوم انسانی محاسباتی

asgari@berkeley.edu



مدل زبانی

- مدل زبانی

$$p(\text{start}, w_1, w_2, \dots, w_n, \text{stop}) = \prod_{i=1}^{n+1} \gamma(w_i | w_1, w_2, \dots, w_{i-1})$$

$P($ ”تاب بنفسه می دهد طرہ مشک سای تو” $) =$
 $P(\text{تاب بنفسه} | \text{می دهد}) \times P(\text{تاب} | \text{بنفسه}) \times P(\text{تاب})$
 $(\text{تاب بنفسه می دهد طرہ} | \text{مشک سای}) \times (\text{تاب بنفسه می دهد} | \text{طرہ}) \times$
 $\times P(\text{تاب بنفسه می دهد طرہ مشک سای} | \text{تو}) \times P(\text{تو})$

- مدل زبانی
- انواع مدل زبانی
 - مدل‌های زبانی n-gram
 - ارزیابی
 - هموار کردن
 - مدل‌های زبانی شبکه عصبی

مدل زبانی

- مدل زبانی
 - انواع مدل زبانی
 - مدل‌های زبانی n-gram
 - ارزیابی
 - هموار کردن
 - مدل‌های زبانی شبکه عصبی

• مدل زبانی یا unigram

$$p(\text{start}, w_1, w_2, \dots, w_n, \text{stop}) = \prod_{i=1}^{n+1} \gamma(w_i)$$

= ("تاب بنفسه می دهد طرہ مشک سای تو")

$P(\text{تاب بنفسه} | \text{می دهد}) = P(\text{تاب} | \text{بنفسه}) \times P(\text{تاب})$

\times (تاب بنفسه می دهد طرہ | مشک سای) $P \times$ (تاب بنفسه می دهد | طرہ) P

تاب بنفسه می دهد طرہ مشک سای | تو) P ×

$$= P_{(توب)} \times P_{(مشک سای)} \times P_{(طره)} \times P_{(بنفسه)} \times P_{(می دهد)} \times P_{(تاب)}$$

مدل زبانی

- مدل زبانی **bigram** یا مدل مارکف مرتبه ۱

$$p(\text{start}, w_1, w_2, \dots, w_n, \text{stop}) = \prod_{i=1}^{n+1} \gamma(w_i | w_{i-1})$$

$P(\text{"تاب بنفسه می دهد طرہ مشک سای تو"})$ =

$= P(\text{تاب بنفسه} | \text{می دهد}) \times P(\text{تاب} | \text{بنفسه})$

$\times P(\text{تاب بنفسه می دهد طرہ} | \text{مشک سای}) \times P(\text{تاب بنفسه می دهد} | \text{طرہ})$

$\times P(\text{تاب بنفسه می دهد طرہ مشک سای} | \text{تو}) \times P(\text{تو})$

$= P(\text{بنفسه} | \text{می دهد}) \times P(\text{تاب} | \text{بنفسه}) \times P(\text{شروع} | \text{تاب})$

$\times P(\text{طرہ} | \text{مشک سای}) \times P(\text{می دهد} | \text{طرہ})$

$\times P(\text{تو} | \text{پایان}) \times P(\text{مشک سای} | \text{تو}) \times P(\text{تو})$

- مدل زبانی
- انواع مدل زبانی
- مدل‌های زبانی **n-gram**
- ارزیابی
- هموار کردن
- مدل‌های زبانی شبکه عصبی

مدل زبانی

- مدل زبانی bigram یا مدل مارکف مرتبه ۱

$$p(\text{start}, w_1, w_2, \dots, w_n, \text{stop}) = \prod_{i=1}^{n+1} \gamma(w_i | w_{i-1})$$

$P = ("تاب بنفسه می دهد طرہ مشک سای تو")$

$= (\text{تاب بنفسه} | \text{می دهد}) P \times (\text{تاب} | \text{بنفسه}) P \times (\text{تاب})$

$\times (\text{تاب بنفسه می دهد طرہ} | \text{مشک سای}) P \times (\text{تاب بنفسه می دهد} | \text{طرہ}) P$

$\times (\text{تاب بنفسه می دهد طرہ مشک سای} | \text{تو}) P$

$= (\text{بنفسه} | \text{می دهد}) P \times (\text{تاب} | \text{بنفسه}) P \times (\text{شروع} | \text{تاب}) P$

$\times (\text{طرہ} | \text{مشک سای}) P \times (\text{می دهد} | \text{طرہ}) P$

$\times (\text{مشک سای} | \text{تو}) P$

- تخمین ماکزیمم likelihood

$$P(w_i | w_{i-1}) = \frac{\text{count}(w_{i-1}, w_i)}{\text{count}(w_{i-1})}$$

- مدل زبانی

- انواع مدل زبانی

- مدل‌های زبانی n-gram

- ارزیابی

- هموار کردن

- مدل‌های زبانی شبکه عصبی

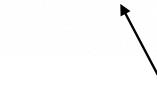
هموار سازی در مثال مدل bigram

MLE estimate

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

Add-1 estimate لایپلاس

$$P_{\text{Laplace}}^*(w_n | w_{n-1}) = \frac{C(w_{n-1} w_n) + 1}{\sum_w (C(w_{n-1} w) + 1)} = \frac{C(w_{n-1} w_n) + 1}{C(w_{n-1}) + V}$$



تعداد کل کلمات متمایز

- مدل زبانی
- انواع مدل زبانی
- مدل‌های زبانی n-gram
- ارزیابی
- هموار کردن
- مدل‌های زبانی شبکه عصبی

هموار سازی در مثال مدل bigram

MLE estimate

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

Add-k estimate

$$P_{\text{Add-k}}^*(w_n | w_{n-1}) = \frac{C(w_{n-1} w_n) + k}{C(w_{n-1}) + kV}$$

- مدل زبانی
- انواع مدل زبانی
- مدل‌های زبانی n-gram
- ارزیابی
- هموار کردن
- مدل‌های زبانی شبکه عصبی

- مدل زبانی

- انواع مدل زبانی

- مدل‌های زبانی **n-gram**

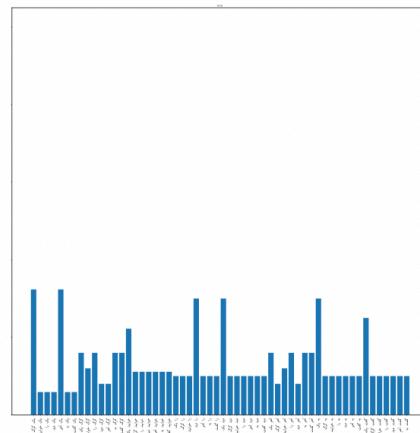
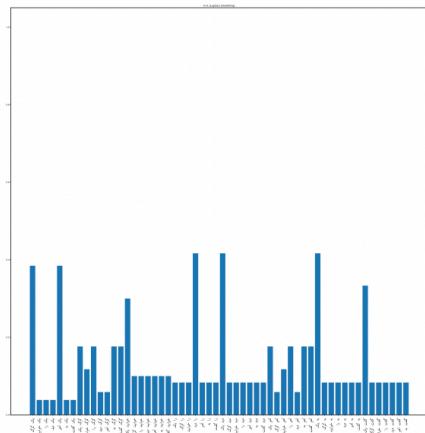
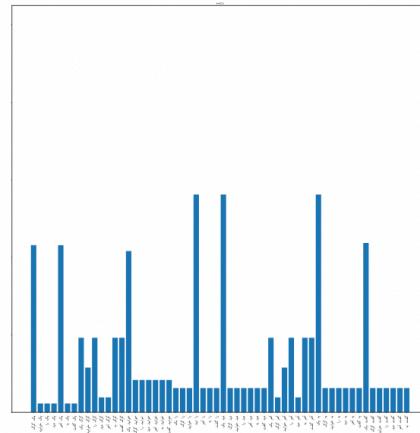
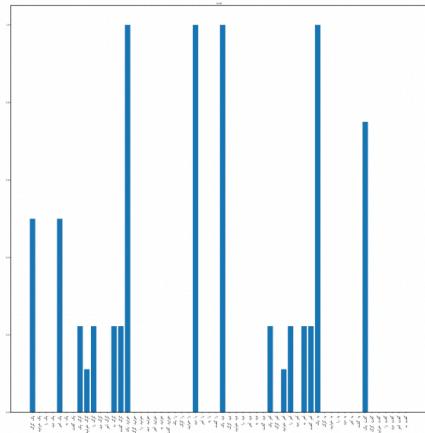
- ارزیابی

- هموار کردن

- مدل‌های زبانی شبکه عصبی

چگونه انتخاب کنیم؟

$K=0, 1, 0.5, 2?$



مدل زبانی

- مدل زبانی
- انواع مدل زبانی
- مدل‌های زبانی **n-gram**
 - ارزیابی
 - هموار کردن
- مدل‌های زبانی شبکه عصبی

- مدل زبانی **m-gram** یا مرتبه $m-1$

$$p(\text{start}, w_1, w_2, \dots, w_n, \text{stop}) = \prod_{i=1}^{n+1} \gamma(w_i | w_{i-m}, \dots, w_{i-1})$$

$$\begin{aligned} P(\text{تاب بنفسه می دهد طرہ مشک سای تو}) &= \\ P(\text{تاب بنفسه} | \text{می دهد}) P &\times (P(\text{تاب} | \text{بنفسه}) \times (P(\text{تاب})) \\ (P(\text{تاب بنفسه می دهد طرہ} | \text{مشک سای}) P &\times (P(\text{تاب بنفسه می دهد} | \text{طرہ}) P \\ \times P(\text{تاب بنفسه می دهد طرہ مشک سای} | \text{تو}) &\times \dots \end{aligned}$$

- تخمین ماکزیمم likelihood

مدل زبانی

- مدل زبانی **m-gram** یا مرتبه m

$$p(\text{start}, w_1, w_2, \dots, w_n, \text{stop}) = \prod_{i=1}^{n+1} \gamma(w_i | w_{i-m}, \dots, w_{i-1})$$

= ("تاب بنفسه می دهد طرہ مشک سای تو") P

P (تاب بنفسه | می دهد) P × (تاب | بنفسه) P × (تاب)

(تاب بنفسه می دهد طرہ | مشک سای) P × (تاب بنفسه می دهد | طرہ) P

× (تاب بنفسه می دهد طرہ مشک سای | تو) P

= ...

- تخمین ماکزیمم likelihood

$$P(W_n | W_{n-m} \dots W_{n-1}) = \frac{C(W_{(n-m):(n-1)} W_n)}{\sum_{* \in V} C(W_{(n-m):(n-1)}, *)}$$

- مدل زبانی
- انواع مدل زبانی
- مدل‌های زبانی **n-gram**
- ارزیابی
- هموار کردن
- مدل‌های زبانی شبکه عصبی

مدل زبانی

- مدل زبانی

- انواع مدل زبانی

- مدل‌های زبانی **n-gram**

- ارزیابی

- هموار کردن

- مدل‌های زبانی شبکه عصبی

- مدل زبانی **m-gram** یا مرتبه m

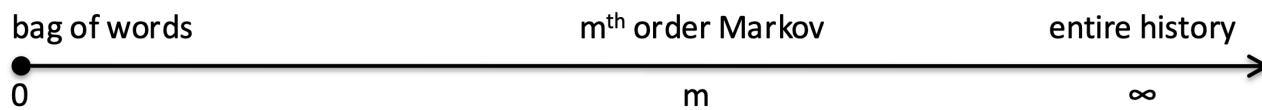
$$p(\text{start}, w_1, w_2, \dots, w_n, \text{stop}) = \prod_{i=1}^{n+1} \gamma(w_i | w_{i-m}, \dots, w_{i-1})$$

= ("تاب بنفسه می دهد طره مشک سای تو")
 $P(\text{تاب بنفسه} | \text{می دهد}) P \times (\text{تاب} | \text{بنفسه}) P \times (\text{تاب})$

$\times (\text{تاب بنفسه} | \text{می دهد طره} | \text{مشک سای}) P \times (\text{تاب بنفسه} | \text{می دهد طره} | \text{طره}) P$

$\times (\text{تاب بنفسه} | \text{می دهد طره مشک سای} | \text{تو}) P$

= ...



مدل زبانی

- مدل زبانی **m-gram** یا مرتبه m

$$p(\text{start}, w_1, w_2, \dots, w_n, \text{stop}) = \prod_{i=1}^{n+1} \gamma(w_i | w_{i-m}, \dots, w_{i-1})$$

= ("تاب بنفسه می دهد طرہ مشک سای تو") P

P (تاب بنفسه | می دهد) P × (تاب | بنفسه) P × (تاب)

(تاب بنفسه می دهد طرہ | مشک سای) P × (تاب بنفسه می دهد | طرہ) P

× (تاب بنفسه می دهد طرہ مشک سای | تو) P

= ...

- تخمین ماکزیمم likelihood

$$P(W_n | W_{n-m} \dots W_{n-1}) = \frac{C(W_{(n-m):(n-1)} W_n)}{\sum_{* \in V} C(W_{(n-m):(n-1)}, *)}$$

- مدل زبانی
- انواع مدل زبانی
- مدل‌های زبانی **n-gram**
- ارزیابی
- هموار کردن
- مدل‌های زبانی شبکه عصبی

مدل زبانی

- مدل زبانی

- انواع مدل زبانی

- مدل‌های زبانی **n-gram**

- ارزیابی

- هموار کردن

- مدل‌های زبانی شبکه عصبی

- مدل زبانی **m-gram** یا مرتبه m

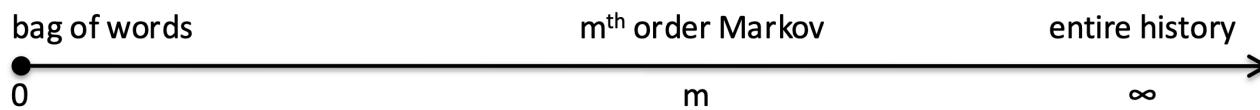
$$p(\text{start}, w_1, w_2, \dots, w_n, \text{stop}) = \prod_{i=1}^{n+1} \gamma(w_i | w_{i-m}, \dots, w_{i-1})$$

= ("تاب بنفسه می دهد طره مشک سای تو")
 $P(\text{تاب بنفسه} | \text{می دهد}) P \times (\text{تاب} | \text{بنفسه}) P \times (\text{تاب})$

$\times (\text{تاب بنفسه می دهد طره} | \text{مشک سای}) P \times (\text{تاب بنفسه می دهد طره} | \text{طره}) P$

$\times (\text{تاب بنفسه می دهد طره مشک سای سای تو}) P$

= ...



هموار سازی در مثال مدل bigram

MLE estimate

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

Add-k estimate

$$P_{\text{Add-k}}^*(w_n | w_{n-1}) = \frac{C(w_{n-1} w_n) + k}{C(w_{n-1}) + kV}$$

- مدل زبانی
- انواع مدل زبانی
- مدل‌های زبانی n-gram
- ارزیابی
- هموار کردن
- مدل‌های زبانی شبکه عصبی

بسم الله الرحمن الرحيم

پردازش زبانهای طبیعی

جلسه ۹

احسان الدین عسگری

۱۴۰۳ فروردین

<http://language.ml/>

دانشکده مهندسی کامپیوتر

آزمایشگاه پردازش هوشمند متن و زبان و علوم انسانی محاسباتی

asgari@berkeley.edu



جایگزینی

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

ساده backoff $\lambda_{\text{argmax}} (\lambda_i > 0) = 1$. Otherwise $\lambda_j = 0$

پیچیده

- مدل زبانی
- انواع مدل زبانی
- مدل‌های زبانی n-gram
- ارزیابی
- هموار کردن
- مدل‌های زبانی شبکه عصبی

جایگزینی

- مدل زبانی
- انواع مدل زبانی
- مدل‌های زبانی **n-gram**
- ارزیابی
- هموار کردن
- مدل‌های زبانی شبکه عصبی

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

ساده backoff $\lambda_{\text{argmax}} (\lambda_i > 0) = 1$. Otherwise $\lambda_j = 0$

Katz-backoff

$$P_{\text{BO}}(w_n | w_{n-N+1:n-1}) = \begin{cases} P^*(w_n | w_{n-N+1:n-1}), & \text{if } C(w_{n-N+1:n}) > 0 \\ \alpha(w_{n-N+1:n-1}) P_{\text{BO}}(w_n | w_{n-N+2:n-1}), & \text{otherwise.} \end{cases}$$

پیچیده

جایگزینی

- مدل زبانی
- انواع مدل زبانی
- مدل‌های زبانی n-gram
- ارزیابی
- هموار کردن
- مدل‌های زبانی شبکه عصبی

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

ساده backoff

$$\lambda_{\text{argmax}} (\lambda_i > 0) = 1. \text{ Otherwise } \lambda_j = 0$$

Katz-backoff

$$P_{\text{BO}}(w_n | w_{n-N+1:n-1}) = \begin{cases} P^*(w_n | w_{n-N+1:n-1}), & \text{if } C(w_{n-N+1:n}) > 0 \\ \alpha(w_{n-N+1:n-1}) P_{\text{BO}}(w_n | w_{n-N+2:n-1}), & \text{otherwise.} \end{cases}$$

درون یابی

$$\begin{aligned}\hat{P}(w_n | w_{n-2} w_{n-1}) = & \lambda_1 P(w_n) \\ & + \lambda_2 P(w_n | w_{n-1}) \\ & + \lambda_3 P(w_n | w_{n-2} w_{n-1})\end{aligned} \quad \sum_i \lambda_i = 1$$

پیچیده

جایگزینی

$$\begin{aligned}\hat{P}(w_n | w_{n-2} w_{n-1}) &= \lambda_1 P(w_n) \\ &\quad + \lambda_2 P(w_n | w_{n-1}) \\ &\quad + \lambda_3 P(w_n | w_{n-2} w_{n-1})\end{aligned}$$

ساده

backoff

$$\lambda_{\text{argmax}} (\lambda_i > 0) = a. \text{ Otherwise } \lambda_j = 0$$

Katz-backoff

$$P_{\text{BO}}(w_n | w_{n-N+1:n-1}) = \begin{cases} P^*(w_n | w_{n-N+1:n-1}), & \text{if } C(w_{n-N+1:n}) > 0 \\ \alpha(w_{n-N+1:n-1}) P_{\text{BO}}(w_n | w_{n-N+2:n-1}), & \text{otherwise.} \end{cases}$$

درون یابی

$$\begin{aligned}\hat{P}(w_n | w_{n-2} w_{n-1}) &= \lambda_1 P(w_n) \\ &\quad + \lambda_2 P(w_n | w_{n-1}) \\ &\quad + \lambda_3 P(w_n | w_{n-2} w_{n-1})\end{aligned} \quad \sum_i \lambda_i = 1$$

پیچیده

$$\begin{aligned}\hat{P}(w_n | w_{n-2} w_{n-1}) &= \lambda_1(w_{n-2:n-1}) P(w_n) \\ &\quad + \lambda_2(w_{n-2:n-1}) P(w_n | w_{n-1}) \\ &\quad + \lambda_3(w_{n-2:n-1}) P(w_n | w_{n-2} w_{n-1})\end{aligned} \quad \sum_i \lambda_i = 1$$

- مدل زبانی

- انواع مدل زبانی

- مدل‌های زبانی n-gram

- ارزیابی

- هموار کردن

- مدل‌های زبانی شبکه عصبی

انواع ارزیابی

- Intrinsic – ذاتی
 - این که مدل چه خواص جالبی در مدل ایجاد شده.
- extrinsic – غیرذاتی
 - این که مدل در تسک‌ها چقدر خوب عمل می‌کند.

- مدل زبانی
- انواع مدل زبانی
- مدل‌های زبانی n-gram
- ارزیابی
- هموار کردن
- مدل‌های زبانی شبکه عصبی

یادگیری مدل زبانی

داده train

داده dev

داده test

- تنظیم پارامترها
- یادگیری مدل زبانی

- ارزیابی

- مدل زبانی
- انواع مدل زبانی
- مدل‌های زبانی n-gram
- ارزیابی
- هموار کردن
- مدل‌های زبانی شبکه عصبی

ارزیابی ذاتی مدل زبانی

- اگر داده یادگیری و داده آزمون داشته باشیم؟
- ویژگی‌هایی مدل زبانی خوب؟

احتمال معقول نسبت دادن به جملاتی که هنوز ندیده است.

- چه چالش‌هایی داریم؟



- مدل زبانی
- انواع مدل زبانی
- مدل‌های زبانی **n-gram**
- ارزیابی
- هموار کردن
- مدل‌های زبانی شبکه عصبی

یک ارزیابی ذاتی Perplexity

- احتمالی که مدل زبانی به جمله‌های زبان $x^{(i)}$ که هنوز ندیده نسبت می‌دهد.

$$x^{(1)}, x^{(2)}, \dots, x^{(m)} \quad length(x^{(i)}) = n_i \quad M = \sum_{i=1}^m n_i$$

$$\prod_1^m P(x^{(i)}) \longrightarrow \prod_1^m \frac{1}{P(x^{(i)})} \longrightarrow \sqrt[M]{\frac{1}{\prod_1^m P(x^{(i)})}}$$

$$\sqrt[M]{\frac{1}{\prod_1^m P(x^{(i)})}} = 2^{\log_2 \sqrt[M]{\frac{1}{\prod_1^m P(x^{(i)})}}} = 2^{-\frac{1}{M} \sum_{i=1}^m \log_2 P(x^{(i)})}$$

- مدل زبانی
- انواع مدل زبانی
- مدلهای زبانی **n-gram**
 - ارزیابی
 - هموار کردن
 - مدلهای زبانی شبکه عصبی

یک ارزیابی ذاتی Perplexity

- احتمالی که مدل زبانی به جمله‌های زبان $x^{(i)}$ که هنوز ندیده نسبت می‌دهد.

$$x^{(1)}, x^{(2)}, \dots, x^{(m)} \quad length(x^{(i)}) = n_i \quad M = \sum_{i=1}^m n_i$$

$$\prod_1^m P(x^{(i)}) \longrightarrow \prod_1^m \frac{1}{P(x^{(i)})} \longrightarrow \sqrt[M]{\frac{1}{\prod_1^m P(x^{(i)})}}$$

$$\sqrt[M]{\frac{1}{\prod_1^m P(x^{(i)})}} = 2^{\log_2 \sqrt[M]{\frac{1}{\prod_1^m P(x^{(i)})}}} = 2^{-\frac{1}{M} \sum_{i=1}^m \log_2 P(x^{(i)})}$$

- مدل زبانی
- انواع مدل زبانی
- مدلهای زبانی **n-gram**
 - ارزیابی
 - هموار کردن
 - مدلهای زبانی شبکه عصبی

یک ارزیابی ذاتی Perplexity

- احتمالی که مدل زبانی به جمله‌های زبان $x^{(i)}$ که هنوز ندیده نسبت می‌دهد.

$$x^{(1)}, x^{(2)}, \dots, x^{(m)} \quad length(x^{(i)}) = n_i \quad M = \sum_{i=1}^m n_i$$

$$\prod_1^m P(x^{(i)}) \longrightarrow \prod_1^m \frac{1}{P(x^{(i)})} \longrightarrow \sqrt[M]{\frac{1}{\prod_1^m P(x^{(i)})}}$$

$$\sqrt[M]{\frac{1}{\prod_1^m P(x^{(i)})}} = 2^{\log_2 \sqrt[M]{\frac{1}{\prod_1^m P(x^{(i)})}}} = 2^{-\frac{1}{M} \sum_{i=1}^m \log_2 P(x^{(i)})}$$

$P(w_i | w_{i-1})$

- با فرض مارکف مرتبه ۱
اتکا پرپلکسیتی به الگوهای دیده شده داده یادگیری

- مدل زبانی
- انواع مدل زبانی
- مدلهای زبانی **n-gram**
 - ارزیابی
 - هموار کردن
 - مدلهای زبانی شبکه عصبی

توصیه‌های عملی برای چالش‌های مدل زبانی

- سمبول شروع و پایان ($\langle S \rangle$ و $\langle /S \rangle$)
- سمبول کلمات خارج از محدوده (out-of-vocabulary)
- $\langle num_tel \rangle$ یا انواع مشخص $\langle unk \rangle$
- توجه به تغییرات زبانی
- تغییرات واژگانی در گذر زمان، مکان، حوزه استفاده
- تغییرات دستوری

- مدل زبانی
- انواع مدل زبانی
- مدل‌های زبانی n-gram
- ارزیابی
- هموار کردن
- مدل‌های زبانی شبکه عصبی

مدل زبانی

$$\begin{aligned} P(x_1, x_2, x_3, x_4, x_5) &= P(x_1) \\ &\quad \times P(x_2 \mid x_1) \\ &\quad \times P(x_3 \mid x_1, x_2) \\ &\quad \times P(x_4 \mid x_1, x_2, x_3) \\ &\quad \times P(x_5 \mid x_1, x_2, x_3, x_4) \end{aligned}$$

ساده سازی

$$P(x_i \mid x_1, \dots, x_{i-1}) \approx P(x_i)$$

$$P(x_i \mid x_1, \dots, x_{i-1}) \approx P(x_i \mid x_{i-1})$$

$$P(x_i \mid x_1, \dots, x_{i-1}) \approx P(x_i \mid x_{i-2}, x_{i-1})$$

- مدل زبانی

- انواع مدل زبانی

- مدل‌های زبانی **n-gram**

- ارزیابی

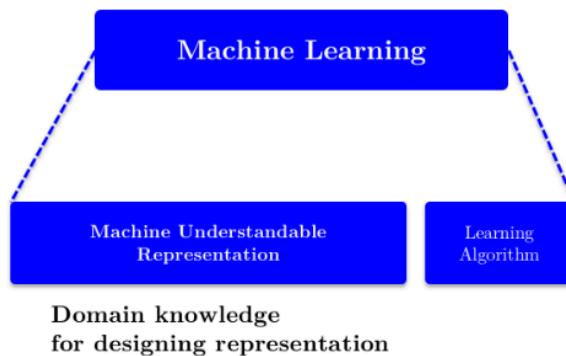
- هموار کردن

- مدل‌های زبانی شبکه عصبی

نمایش داده

- فرمت و نمایش داده متناسب با مخاطب. برای ماشین وکتور، ماتریس.

- مدل زبانی
- انواع مدل زبانی
- مدل‌های زبانی **n-gram**
- ارزیابی
- هموار کردن
- مدل‌های زبانی شبکه عصبی
- نمایش واژگان

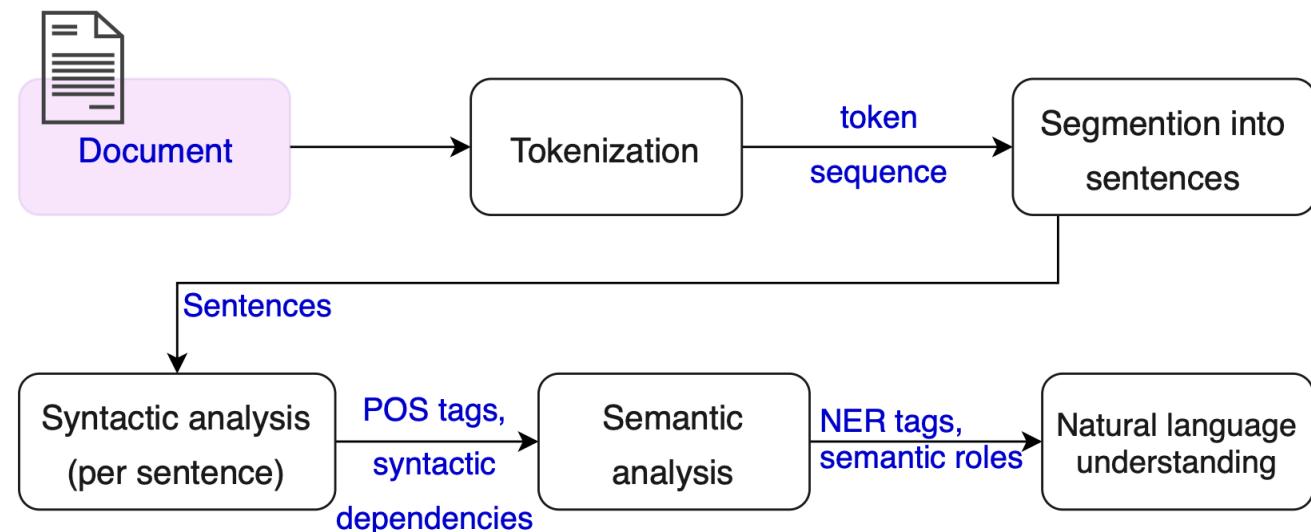


– Tokenization



- Why Subword?
- Char-level
- BPE
- Subword Reg.
- BPE-Dropout
- Multi-granularity
- Multilinguality

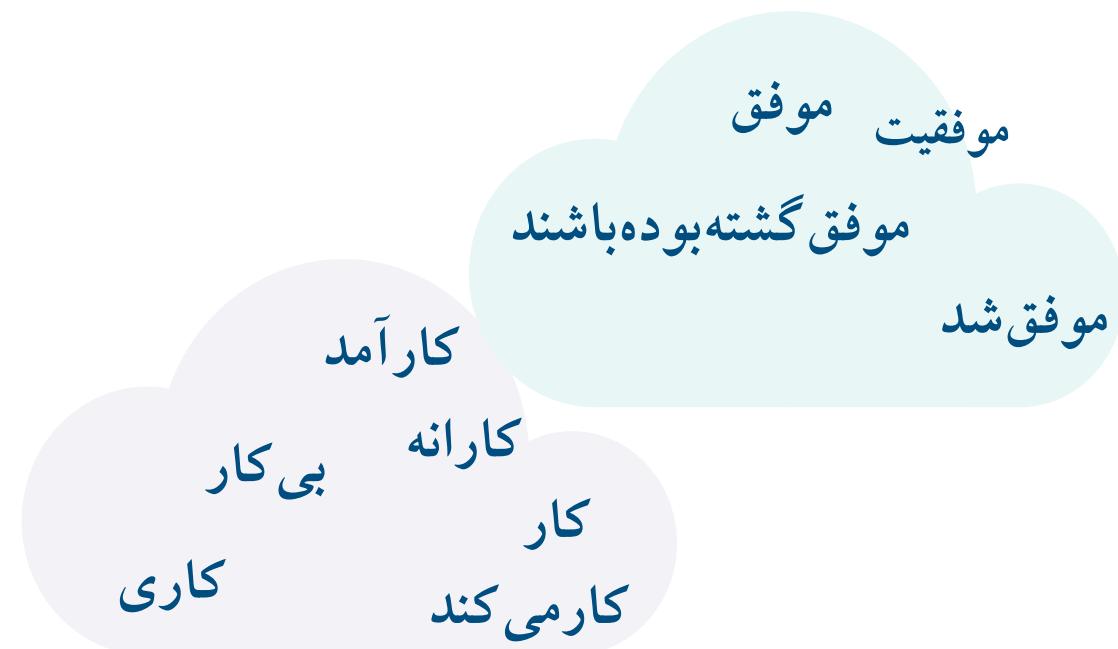
NLP Traditional Pipeline



Tokenization issue?

- Why Subword?
- Char-level
- BPE
- Subword Reg.
- BPE-Dropout
- Multi-granularity
- Multilinguality

Shared Morphemes within the Language



بسم الله الرحمن الرحيم

پردازش زبانهای طبیعی

جلسه ۱۰

احسان الدین عسگری

فرودین ۱۴۰۳

<http://language.ml/>

دانشکده مهندسی کامپیوتر

آزمایشگاه پردازش هوشمند متن و زبان و علوم انسانی محاسباتی

asgari@berkeley.edu



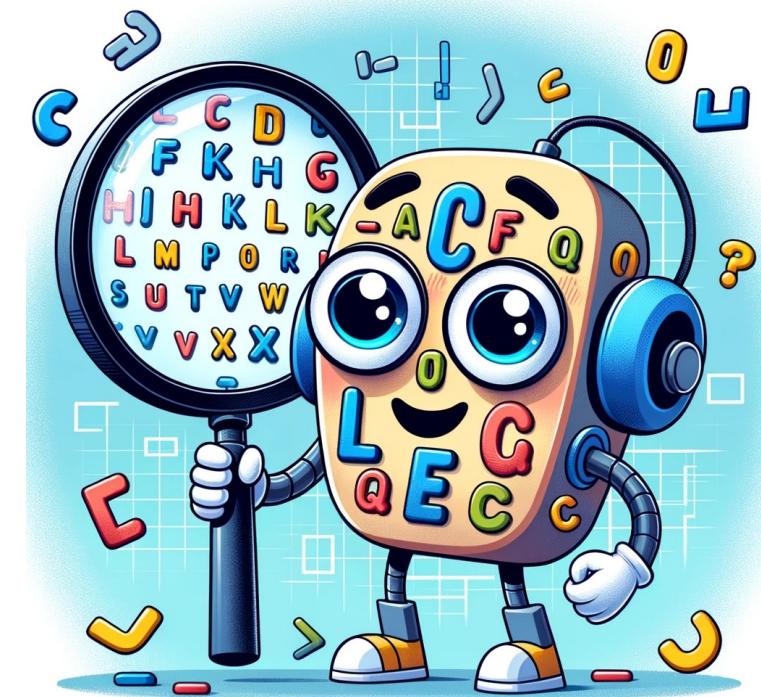
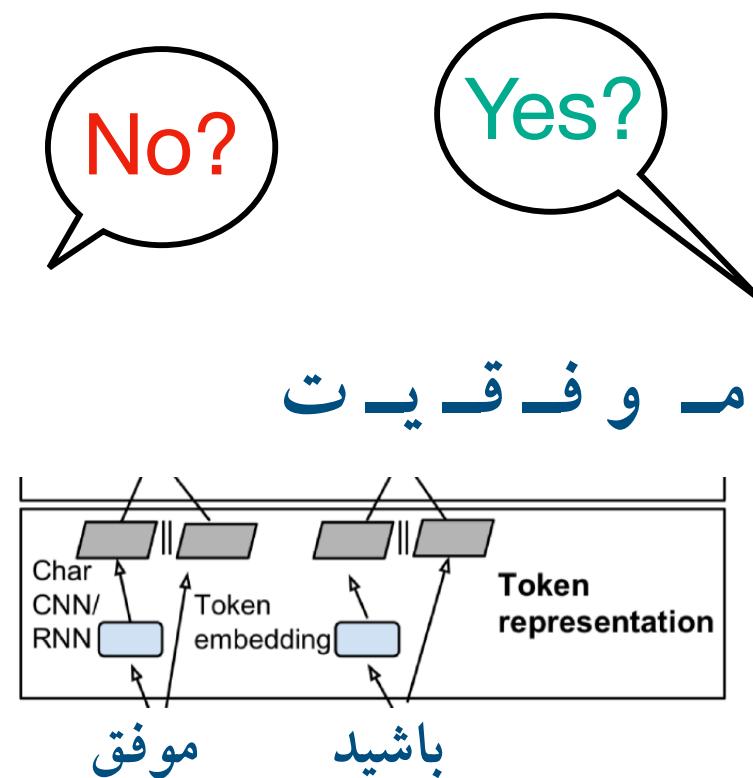
- Why Subword?
- Char-level
- BPE
- Subword Reg.
- BPE-Dropout
- Multi-granularity
- Multilinguality

Shared Morphemes among Languages

English Suffix	German	English	French	Italian	Spanish	Latin	Romanian
-tion	Information	Information	Information	Informazione	Información	Informatio	Informație
-ity	Qualität	Quality	Qualité	Qualità	Calidad	Qualitas	Calitate
-al	Global	Global	Global	Globale	Global	Globalis	Global
-ist	Spezialist	Specialist	Spécialiste	Specialista	Especialista	Specialistus	Specialist
-ism	Kapitalismus	Capitalism	Capitalisme	Capitalismo	Capitalismo	Capitalismus	Capitalism

- Why Subword?
- Char-level
- BPE
- Subword Reg.
- BPE-Dropout
- Multi-granularity
- Multilinguality

Character-level



- Why Subword?
- Char-level
- BPE
- Subword Reg.
- BPE-Dropout
- Multi-granularity
- Multilinguality

Character-level

Advantages	Disadvantages
1. Smaller Vocabulary Size	1. Longer Sequences
2. Handles OOV Words	2. Limited Context Understanding
3. Captures Morphological Patterns	3. Training Difficulty
4. Language Agnosticism	4. Slower Processing Speed
5. Robustness to Noise	5. Suboptimal for Certain Tasks

Adel, Heike, Ehsaneddin Asgari, and Hinrich Schütze.
[Overview of character-based models for natural language processing.](#)
Computational Linguistics and Intelligent Text Processing 2017.



- Why Subword?
- Char-level
- BPE
- Subword Reg.
- BPE-Dropout
- Multi-granularity
- Multilinguality

Byte-pair Encoding (BPE)

- Step 0: Set up vocabulary.
- Step 1: Represent words using characters
- Step 2: Count character pairs in vocabulary.
- Step 3: Merge highest frequency pairs, new symbol.
- Step 4: Continue merging until reaching desired vocab size.

Words in the data:

Initial vocabulary: characters	word	count	Current merge table: (empty)
	c a t	4	
↓	m a t	5	
Split each word into characters	m a t s	2	
	m a t e	3	
	a t e	3	
	e a t	2	

Rico Sennrich, Barry Haddow, and Alexandra Birch.
[Neural Machine Translation of Rare Words with Subword Units.](#)
 ACL 2016

Gif from: <https://tinyurl.com/22xk95hj>

- Why Subword?
- Char-level
- BPE (& BBPE)
- Subword Reg.
- BPE-Dropout
- Multi-granularity
- Multilinguality

Byte-level Byte-pair Encoding (BBPE)

	Original		
	質問して 証明と証拠を求めるましょう		Ask_questions, demand_proof, demand_evidence.
Byte	E8 B3 AA E5 95 8F E3 81 97 E3 81 A6 E2 96 81 E8 A8 BC E6 98 8E E3 81 A8 E8 A8 BC E6 8B A0 E3 82 92 E6 B1 82 E3 82 81 E3 81 BE E3 81 97 E3 82 87 E3 81 86	41 73 6B E2 96 81 71 75 65 73 74 69 6F 6E 73 2C E2 96 81 64 65 6D 61 6E 64 E2 96 81 70 72 6F 6F 66 2C E2 96 81 64 65 6D 61 6E 64 E2 96 81 65 76 69 64 65 6E 63 65 2E	
	1K	E8 B3 AA E595 8F しE381 A6 _E8 A8 BC 明 E381 A8 E8 A8 BC E6 8B A0 をE6 B1 82 めE381 BE しょう	As k _questions , _dem and _pro of , _dem and _evidence .
	2K	E8 B3 AA 問 しE381 A6 _E8 A8BC 明 E381 A8 E8 A8BC E68B A0 を E6 B1 82 めE381 BE しょう	As k _questions , _dem and _pro of , _dem and _evidence .
	4K	E8 B3 AA 問 しE381 A6 _E8 A8BC 明 E381 A8 E8 A8BC 拠 をE6 B1 82 めE381 BE しょう	As k _questions , _dem and _pro of , _dem and _evidence .
	8K	E8 B3 AA 問 しE381 A6 _E8 A8BC 明 E381 A8 E8 A8BC 拠 をE6 B1 82 めE381 BE しょう	As k _questions , _demand _pro of , _demand _evidence .
	16K	E8 B3 AA 問 しE381 A6 _E8 A8BC 明 E381 A8 E8 A8BC 拠 をE6 B1 82 めE381 BE しょう	As k _questions , _demand _pro of , _demand _evidence .
BBPE	32K	E8 B3 AA 問しE381 A6 _E8 A8BC 明 E381 A8 E8 A8BC 拠 をE6 B1 82 めE381 BE しょう	As k _questions , _demand _pro of , _demand _evidence .
	CHAR	質問して 証明と証拠を求めるましょう	Ask_questions, demand_proof, demand_evidence.
	16K	質問して 証明と証拠を求めるましょう	As k _questions , _demand _pro of , _demand _evidence .
	32K	質問して 証明と証拠を求めるでしょう	As k _questions , _demand _pro of , _demand _evidence .

Wang, Changhan; Cho, Kyunghyun; Gu, Jiatao.
Neural machine translation with byte-level subwords.
In Proceedings of [AAAI 2020](#).

- Why Subword?
- Char-level
- BPE (& BBPE)
- Subword Reg.
- BPE-Dropout
- Multi-granularity
- Multilinguality

Byte-level Byte-pair Encoding (BBPE)

- Rare characters from noisy text or character-rich languages such as Japanese and Chinese however can unnecessarily take up vocabulary slots and limit its compactness. Representing text at the level of bytes and using the 256 byte set as vocabulary is a potential solution to this issue.
- We claim that contextualizing BBPE embeddings is necessary, which can be implemented by a convolutional or recurrent layer. Our experiments show that BBPE has comparable performance to BPE while its size is only 1/8 of that for BPE.
- In the multilingual setting, BBPE maximizes vocabulary sharing across many languages and achieves better translation quality..
 - Maybe because of various token granularities in multilingual parallel sentences at the token level
- BBPE enables transferring models between languages with non-overlapping character sets.

Wang, Changhan; Cho, Kyunghyun; Gu, Jiatao.
Neural machine translation with byte-level subwords.
In Proceedings of AAAI 2020.



Mengjiao Zhang and Jia Xu.
Byte-based Multilingual NMT for Endangered Languages.
In Proceedings of COLING 2022.

- Why Subword?
- Char-level
- BPE (& BBPE)
- Subword Reg.
- BPE-Dropout
- Multi-granularity
- Multilinguality

(B)?BPE

- Step 0: Set up vocabulary.
- Step 1: Represent words using characters / bytes
- Step 2: Count character/bytes pairs in vocabulary
- Step 3: Merge highest frequency pairs, new symbol.
- Step 4: Continue merging until reaching desired vocab size.

Issues?

Rico Sennrich, Barry Haddow, and Alexandra Birch.
[Neural Machine Translation of Rare Words with Subword Units](#).
ACL 2016

- Why Subword?
- Char-level
- BPE (& BBPE)
- Subword Reg.
- BPE-Dropout
- Multi-granularity
- Multilinguality

Subword Regularization

Unigram language model, which is capable of outputting multiple subword segmentations with probabilities.

Given Vocabulary V , we want to estimate $p(x_i)$

$$X^{(s)} \in D \rightarrow \text{"sentence"} \quad \mathbf{x} = (x_1, \dots, x_M) \rightarrow \text{"subword sequence"} \quad p(\mathbf{x}) = \prod_{i=1}^M p(x_i) \rightarrow \text{"unigram language model"}$$

Subwords (_ means spaces)	Vocabulary id sequence
_Hell/o/_world	13586 137 255
_H/ello/_world	320 7363 255
_He/llo/_world	579 10115 255
_/He/l/l/o/_world	7 18085 356 356 137 255
_H/el/l/o/_world	320 585 356 137 7 12295

Table 1: Multiple subword sequences encoding the same sentence “Hello World”



Taku Kudo.

Subword Regularization: Improving Neural Network Translation Models with Multiple Subword Candidates.
In Proceedings of the [ACL 2018](#).

- Why Subword?
- Char-level
- BPE (& BBPE)
- Subword Reg.
- BPE-Dropout
- Multi-granularity
- Multilinguality

Subword Regularization

1. Heuristically make a reasonably **big seed vocabulary** V
2. Repeat the following steps **until $|V|$ reaches a desired vocabulary size.**
 - (a) Fixing the set of vocabulary, optimize $p(x)$ with the EM algorithm.

$$\mathcal{L} = \sum_{s=1}^{|D|} \log \left(P(X^{(s)}) \right) = \sum_{s=1}^{|D|} \log \left(\sum_{x \in \mathcal{S}(X^{(s)})} P(x) \right) \rightarrow \text{Log likelihood}$$

$X^{(s)} \in D \rightarrow \text{"sentence"}$

$|D| \rightarrow \text{"size of the dataset"}$

$\mathcal{S}(X^{(s)}) \rightarrow \text{"set of segmentation candidates built from the input sentence "} X^{(s)}$

$x = (x_1, \dots, x_M) \rightarrow \text{"subword sequence"}$

$$p(x) = \prod_{i=1}^M p(x_i) \rightarrow \text{"unigram language model"}$$

- (b) Compute the $loss_i$ for each subword x_i , where $loss_i$ represents how likely the likelihood L is reduced when the subword x_i is removed from the current vocabulary.
- (c) Sort the symbols by $loss_i$ and keep top η % of subwords (η is 80, for example). Note that we always keep the subwords consisting of a single character to avoid out-of-vocabulary.

Taku Kudo.

Subword Regularization: Improving Neural Network Translation Models with Multiple Subword Candidates.
In Proceedings of the ACL 2018.

- Why Subword?
- Char-level
- BPE (& BBPE)
- Subword Reg.
- BPE-Dropout
- Multi-granularity
- Multilinguality

BPE-Dropout

BPE-dropout - simple and effective subword regularization method based on and compatible with conventional BPE.

It stochastically corrupts the segmentation procedure of BPE, which leads to producing multiple segmentations within the same fixed BPE framework.

Using BPE-dropout during training and the standard BPE during inference improves translation quality compared to the previous subword regularization.

Algorithm 1: BPE-dropout

```
current_split ← characters from input_word;
do
    merges ← all possible merges1 of tokens
    from current_split;
    for merge from merges do
        /* The only difference
        from BPE */
        remove merge from merges with the
        probability p;
    end
    if merges is not empty then
        merge ← select the merge with the
        highest priority from merges;
        apply merge to current_split;
    end
while merges is not empty;
return current_split;
```

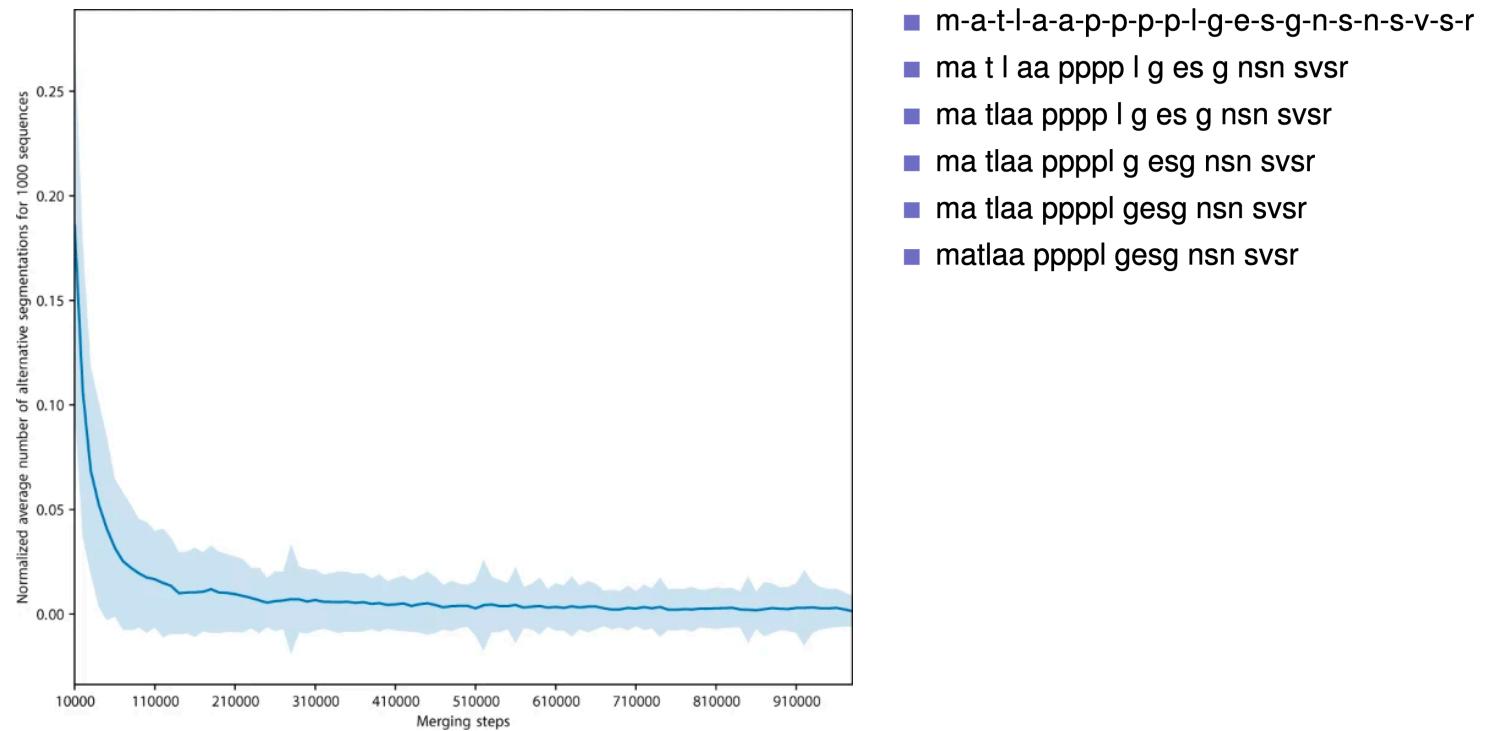
Ivan Prosvilov, Dmitrii Emelianenko, and Elena Voita.

BPE-Dropout: Simple and Effective Subword Regularization.

In Proceedings of the [ACL 2020](#).

- Why Subword?
- Char-level
- BPE (& BBPE)
- Subword Reg.
- BPE-Dropout
- Multi-granularity
- Multilinguality

Multi-Granularity BPE for Bioinformatics



Asgari, Ehsaneddin, Alice C. McHardy, and Mohammad RK Mofrad.

[Probabilistic variable-length segmentation of protein sequences for discriminative motif discovery...](#)

Scientific reports 2019.

- Why Subword?
- Char-level
- BPE (& BBPE)
- Subword Reg.
- BPE-Dropout
- Multi-granularity
- Multilinguality

XLM-V

- Large multilingual language models typically rely on a single vocabulary shared across 100+ languages.
- As these models have increased in parameter count and depth, vocabulary size has remained largely unchanged. This vocabulary bottleneck limits the representational capabilities of multilingual models like XLM-R.
- While multilingual language models have increased in parameter count and depth over time, vocabulary size has largely remained unchanged:
- 250K token vocabulary size as XLM-R base (Conneau et al., 2019), a 250M parameter model.

XLM-V: Overcoming the Vocabulary Bottleneck in Multilingual Masked Language Models
Arxiv Oct 2023

- Why Subword?
- Char-level
- BPE (& BBPE)
- Subword Reg.
- BPE-Dropout
- Multi-granularity
- Multilinguality

XLM-V

- Vocabulary bottleneck hinders the performance of multilingual models on question answering and sequence labeling where in-depth token-level and sequence-level understanding is essential (Wang et al., 2019).
- (1) vocabularies can be improved by de-emphasizing token sharing between languages with little lexical overlap
- (2) proper vocabulary capacity allocation for individual languages is crucial for ensuring that diverse languages are well-represented.

XLM-V: Overcoming the Vocabulary Bottleneck in Multilingual Masked Language Models
Arxiv Oct 2023

- Why Subword?
- Char-level
- BPE (& BBPE)
- Subword Reg.
- BPE-Dropout
- Multi-granularity
- Multilinguality

XLM-V

Finding language clusters

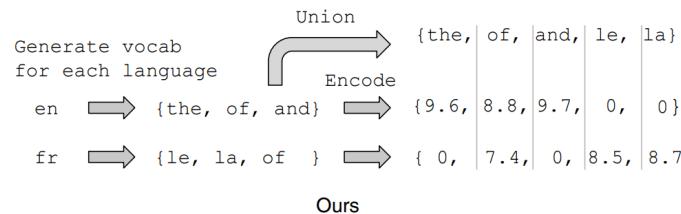
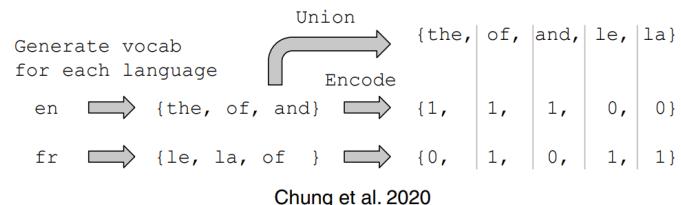


Figure 1: Similar to Chung et al. (2020), we also leverage the per-language sentencepiece vocabularies as a “lexical fingerprint” for clustering. However, instead of using binary vectors, we use the unigram log probability instead.

بسم الله الرحمن الرحيم

پردازش زبانهای طبیعی

جلسه ۱۱

احسان الدین عسگری

فرودین ۱۴۰۳

<http://language.ml/>

دانشکده مهندسی کامپیوتر

آزمایشگاه پردازش هوشمند متن و زبان و علوم انسانی محاسباتی

asgari@berkeley.edu



نمایش واژگان

- فرمت و نمایش داده متناسب با مخاطب. برای ماشین وکتور، ماتریس.

- نمایش one-hot

دوست = [0 0 0 0 0 0 0 0 0 1 0 0 0]

رفیق = [0 0 0 0 0 0 1 0 0 0 0 0 0]

- مدل زبانی
- انواع مدل زبانی
- مدل‌های زبانی n-gram
- ارزیابی
- هموار کردن
- مدل‌های زبانی شبکه عصبی
- نمایش واژگان

بازنمایی واژگان

- بر چه اساسی بازنمایی برای واژگان تولید کنیم؟



- مدل زبانی
- انواع مدل زبانی
- مدل‌های زبانی n-gram
- ارزیابی
- هموار کردن
- مدل‌های زبانی شبکه عصبی
- نمایش واژگان

- مدل زبانی

- انواع مدل زبانی

- مدل‌های زبانی **n-gram**

- ارزیابی

- هموار کردن

- مدل‌های زبانی شبکه عصبی

- نمایش واژگان

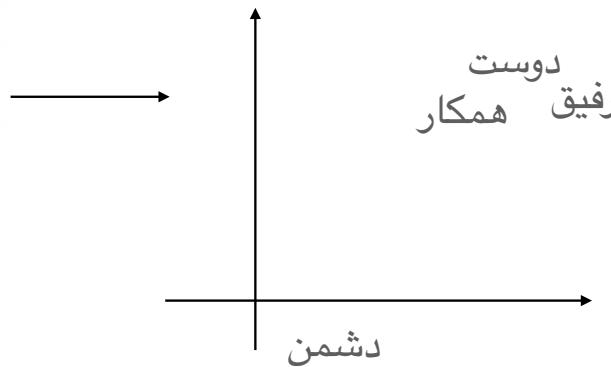
Distributional semantics

- معنای کلمه در کلمات پرسامد مجاور آن نهفته است!

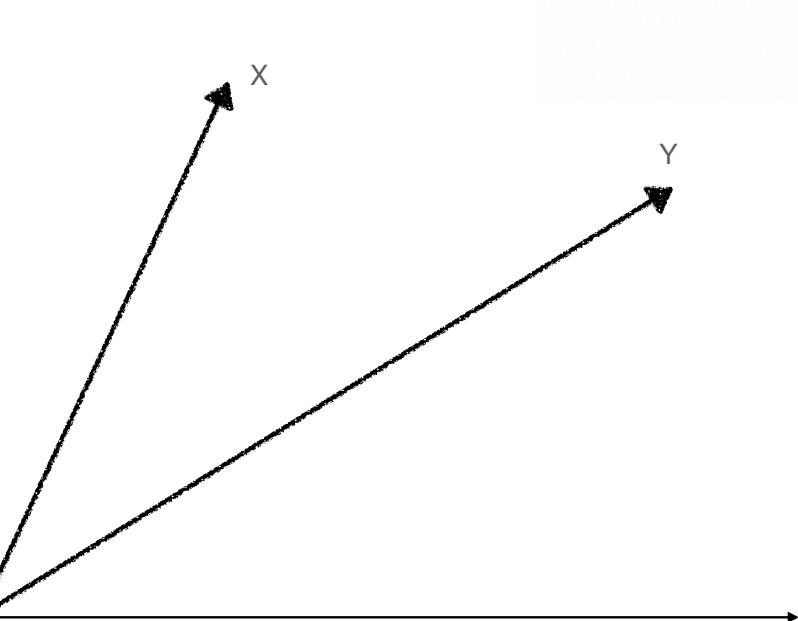
- نمایش برداری کلمات بر این اساس

= دوست
[000000000010000]

= رفیق
[000000010000000]



تعریف شباهت متنی



$$\cos(x, y) = \frac{\sum_{i=1}^F x_i y_i}{\sqrt{\sum_{i=1}^F x_i^2} \sqrt{\sum_{i=1}^F y_i^2}}$$

- مدل زبانی
- انواع مدل زبانی
 - مدل‌های زبانی n-gram
 - ارزیابی
 - هموار کردن
 - مدل‌های زبانی شبکه عصبی
 - نمایش واژگان

بازنمایی متنی

	W1	W2	...	Wn
Doc 1				
Doc 2			۱ . یا	
.				
Doc m				

[1 0 1 0 1]

[1 0 1 0 0]

بازنمایی متنی

	W1	W2	...	Wn
Doc 1				
Doc 2		Freq		
.				
.				
Doc m				

بازنمایی متن با استفاده از one-hot

	W1	W2	...	Wn
Doc 1		■		
Doc 2	■		■	
.				
Doc m			■	

بازنمایی متنی با استفاده از tf-idf

	W1	W2	...	Wm
Doc 1				
Doc 2			TF-idf	
.				
Doc N				

$$tf(t, D) = \frac{\#(t, D)}{\max_{t' \in D} \#(t', D)}$$

$$idf(t) = \log \frac{N}{\sum_{D:t \in D} 1}$$

$$tf \cdot idf(t, D) = tf(t, D) \cdot idf(t)$$

بازنمایی واژگان بر اساس رویداد



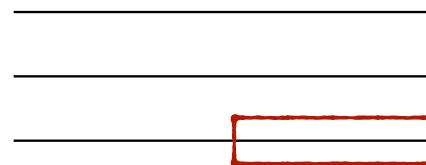
	v	w	...	wn
Doc 1		0.2		
Doc 2	0.4		0.3	
.				
Doc m		0.2	0.1	0.1

$$\text{cosine}(\mathbf{v}, \mathbf{w}) = \frac{\mathbf{v} \cdot \mathbf{w}}{|\mathbf{v}| |\mathbf{w}|} = \frac{\sum_{i=1}^N v_i w_i}{\sqrt{\sum_{i=1}^N v_i^2} \sqrt{\sum_{i=1}^N w_i^2}}$$

بازنمایی متن با استفاده از one-hot

	W1	W2	...	Wn
W 1		■		
W 2	■		■	
.				
W n			■	

Window-size = 5



استفاده از pointwise mutual information

-اطلاعات متقابل - MI: اندازه‌گیری میزان وابستگی متقابل دو متغیر تصادفی:

$$I(X, Y) = \sum_x \sum_y P(x, y) \log_2 \frac{P(x, y)}{P(x)P(y)}$$

-اطلاعات متقابل نقطه‌ای - PMI: اندازه‌گیری میزان وابستگی دو رویداد:

$$\log_2 \frac{P(x, y)}{P(x)P(y)}$$

- مدل زبانی
- انواع مدل زبانی

- مدل‌های زبانی n-gram
- مدل‌های زبانی و بازنمایی
- بازنمایی واژگان

- بازنمایی متن Tf-idf
- ارزیابی بازنمایی
- نظریه distributional semantics

- معیار PPMI

- بازنمایی بر اساس svd
- شبکه عصبی

- مدل skipgram

- Rnn-lstm

استفاده از pointwise mutual information

- اطلاعات متقابل نقطه‌ای - PMI: اندازه‌گیری میزان وابستگی یک واژه و یک سیاق:

$$\text{PMI}(w, c) = \log_2 \frac{P(w, c)}{P(w)P(c)}$$

w : کلمه
c : سیاق

$P(w|c) / p(w)$

PPMI -

$$PPMI = \max \left(\log_2 \frac{P(w, c)}{P(w)P(c)}, 0 \right)$$

- مدل زبانی
- انواع مدل زبانی
 - مدل‌های زبانی n-gram
 - مدل‌های زبانی و بازنمایی
 - بازنمایی واژگان
 - بازنمایی متن Tf-idf
 - ارزیابی بازنمایی
 - نظریه distributional semantics
 - معیار PPMI
 - بازنمایی بر اساس svd
 - شبکه عصبی
 - مدل skipgram
 - Rnn-Lstm

استفاده از PPMI

	Hamlet	Macbeth	Romeo & Juliet	Richard III	Julius Caesar	Tempest	Othello	King Lear	total
knife	1	1	4	2		2		2	12
dog	2		6	6		2		12	28
sword	17	2	7	12		2		17	57
love	64		135	63		12		48	322
like	75	38	34	36	34	41	27	44	329
total	159	41	186	119	34	59	27	123	748

$$PMI(\text{love}, \text{R\&J}) = \frac{\frac{135}{748}}{\frac{186}{748} \times \frac{322}{748}}$$

- مدل زبانی
- انواع مدل زبانی
- مدل‌های زبانی n-gram
- ارزیابی
- هموار کردن
- مدل‌های زبانی شبکه عصبی
- نمایش واژگان

- مدل زبانی

- انواع مدل زبانی

- مدل‌های زبانی n-gram

- ارزیابی

- هموار کردن

- مدل‌های زبانی شبکه عصبی

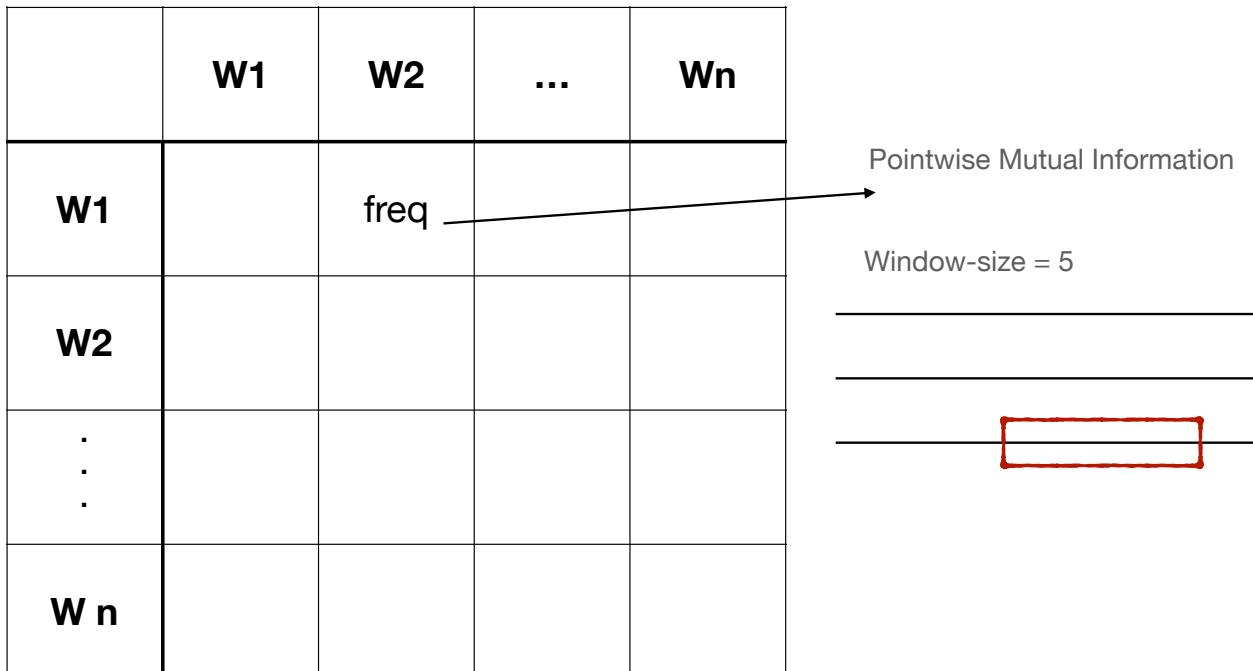
- نمایش واژگان

استفاده از term-context

term	contexts					
	L: the big	R: ate dinner	L: the small	L: the yellow	...	
dog	1	1	0	0	...	
cat	0	1	1	1	...	

- Each cell enumerates the number of time a directional context phrase appeared in a specific position around the term.

استفاده از ماتریس term-term (within context)



- مدل زبانی
- انواع مدل زبانی

- مدل‌های زبانی **n-gram**
- مدل‌های زبانی و بازنمایی
- بازنمایی واژگان
- بازنمایی متن **Tf-idf**
- ارزیابی بازنمایی
- نظریه **distributional semantics**
- معیار **PPMI**
- بازنمایی بر اساس **svd**
- شبکه عصبی
- **skipgram** مدل
- **Rnn-Lstm**

دوره SVD تجزیه ماتریسی

$$M = U S V^T$$

$$M \in \mathbb{R}^{|V| \times D}$$

$$U^T U = I_k$$

- S - ماتریس قطری (انحراف معیار نقاط داده در راستای سطر متناظر از U)

-معمولًا مرتب شده

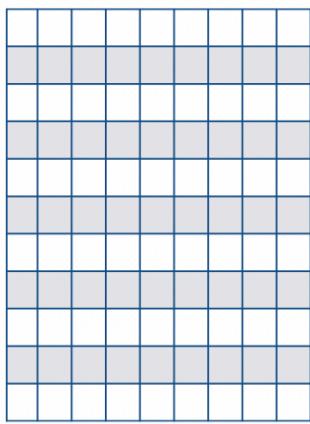
m یک ماتریس با rank M -

- U ماتریس اورتونورمال

- مدل زبانی
- انواع مدل زبانی

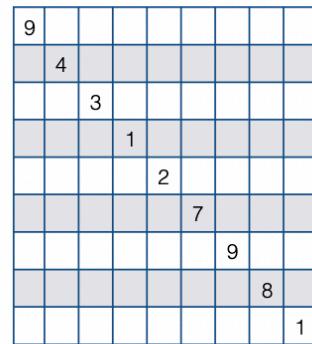
- مدل‌های زبانی n-gram
- مدل‌های زبانی و بازنمایی
- بازنمایی واژگان
- بازنمایی متن Tf-idf
- ارزیابی بازنمایی
- نظریه distributional semantics
- معیار PPMI
- بازنمایی بر اساس svd
- شبکه عصبی
- مدل skipgram
- Rnn-Lstm

استفاده از ماتریس (term–doc context)



A 10x10 grid of light blue squares representing the term matrix U .

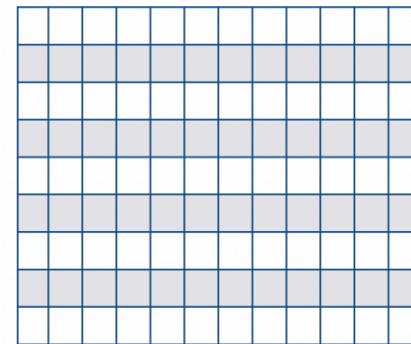
\times



A 9x10 grid of light blue squares representing the document matrix S . The columns are labeled with numbers 1 through 9 from bottom to top.

9									
4									
3									
	1								
	2								
		7							
			9						
				8					
					1				

\times



A 10x10 grid of light blue squares representing the transpose of the term matrix V^T .

U

S

V^T

- مدل زبانی
- انواع مدل زبانی

• مدل‌های زبانی **n-gram**

• مدل‌های زبانی و بازنمایی

• بازنمایی واژگان

• بازنمایی متن **Tf-idf**

• ارزیابی بازنمایی

• نظریه **distributional semantics**

• معیار **PPMI**

• بازنمایی بر اساس **svd**

• شبکه عصبی

• **skipgram** مدل

• **Rnn-lstm**

- مدل زبانی
- انواع مدل زبانی

- مدل‌های زبانی
- مدل‌های زبانی و بازنمایی

• بازنمایی واژگان

• بازنمایی متن **Tf-idf**

• ارزیابی بازنمایی

• نظریه **distributional semantics**

• معیار **PPMI**

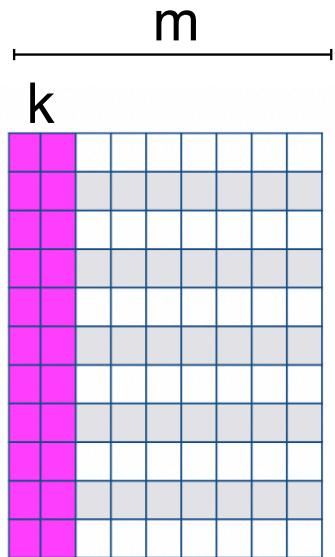
• بازنمایی بر اساس **svd**

• شبکه عصبی

• مدل **skipgram**

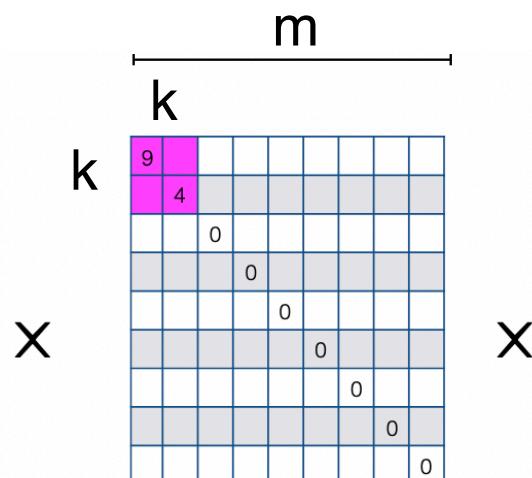
• **Rnn-lstm**

استفاده از ماتریس (context) term-doc

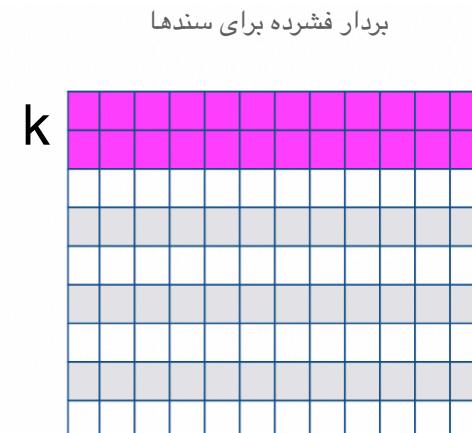


$|V| \times k$

بردار فشرده برای لغات



$k \times k$



$k \times D$

بسم الله الرحمن الرحيم

پردازش زبانهای طبیعی

جلسه ۱۲

احسان الدین عسگری

فرودین ۱۴۰۳

<http://language.ml/>

دانشکده مهندسی کامپیوتر

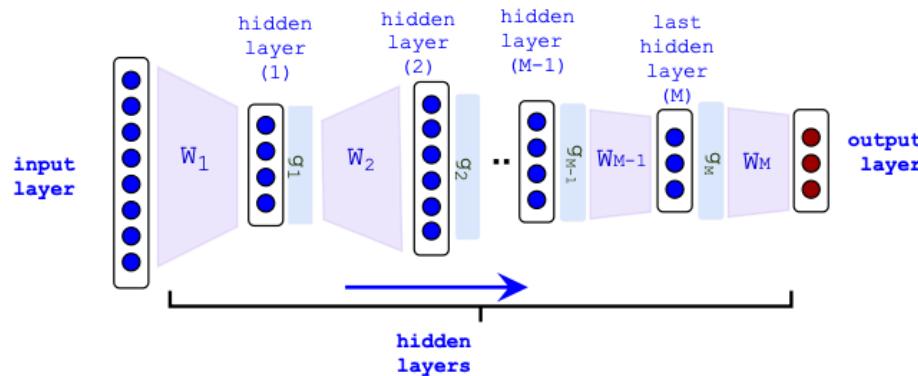
آزمایشگاه پردازش هوشمند متن و زبان و علوم انسانی محاسباتی

asgari@berkeley.edu



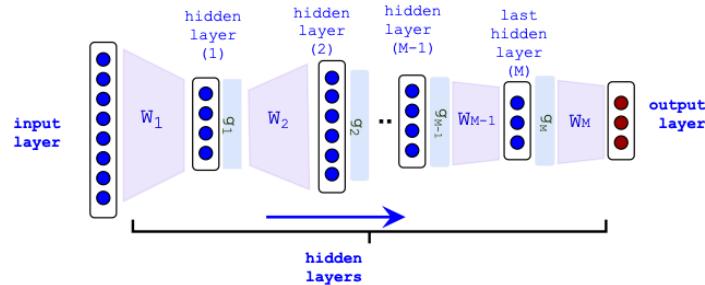
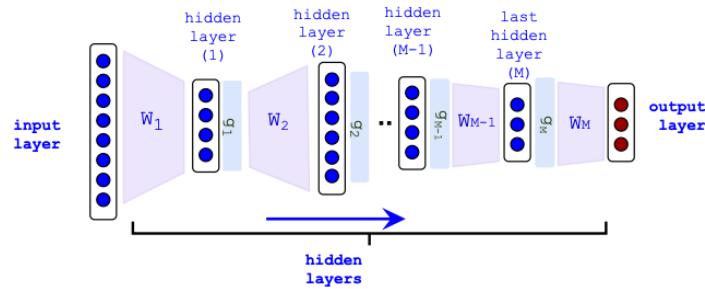
شبکه های عصبی

- خودکار کردن یافتن نمایش داده
- شروع از یک نمایش بسیار ساده و لایه لایه آن را بهبود دادن تا پیش بینی در لایه آخر.
- بروز رسانی نمایش های میانی با استفاده از خطای محاسبه شده در انتهای
- نیاز به داده های یادگیری بالا
- مدل زبانی گزینه ایده آل!



- مدل زبانی
- انواع مدل زبانی
 - مدل های زبانی **n-gram**
 - مدل های زبانی و بازنمایی
 - بازنمایی واژگان
- بازنمایی متن **Tf-idf**
- ارزیابی بازنمایی
- **distributional semantics** نظریه
- معیار **PPMI**
- بازنمایی بر اساس **svd**
- شبکه عصبی
- **skipgram** مدل
- **Rnn-Lstm**

شبکه های عصبی



- مدل زبانی
- انواع مدل زبانی

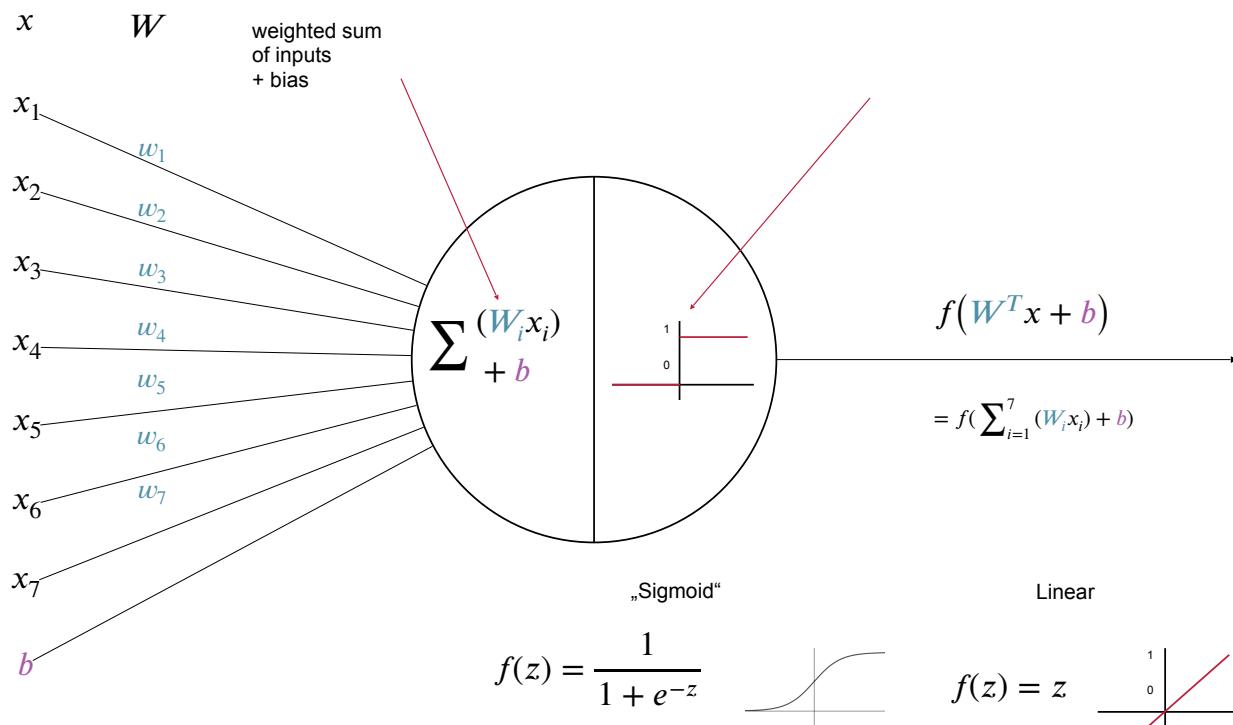
- مدل‌های زبانی
- مدل‌های زبانی و بازنمایی
- بازنمایی واژگان
- بازنمایی متن Tf-idf
- ارزیابی بازنمایی
- نظریه distributional semantics
- معیار PPMI

- بازنمایی بر اساس svd
- شبکه عصبی

- skipgram مدل

- Rnn-lstm

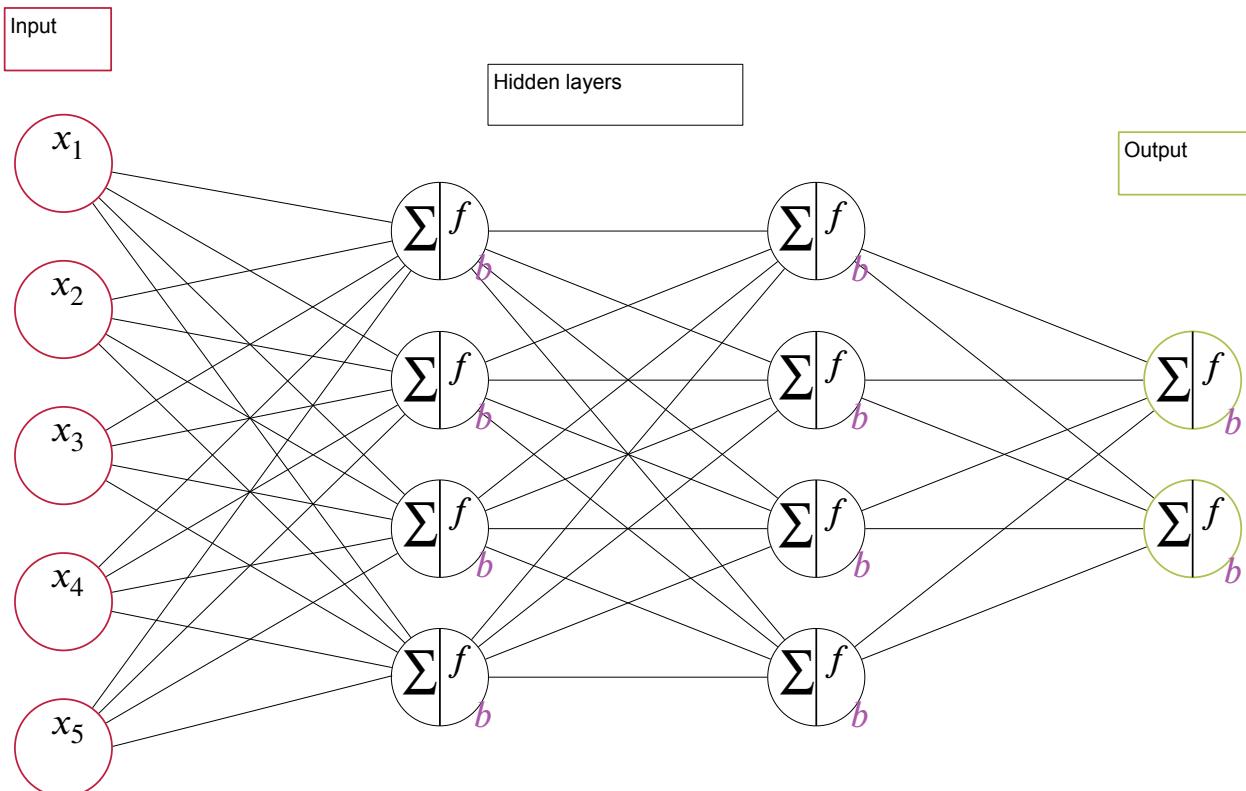
پرسپترون



- مدل زبانی
- انواع مدل زبانی

- مدل‌های زبانی
- مدل‌های زبانی و بازنمایی
- بازنمایی واژگان
- بازنمایی متن
- ارزیابی بازنمایی
- نظریه distributional semantics
- معیار PPMI
- بازنمایی بر اساس svd
- شبکه عصبی
- مدل skipgram
- Rnn-Lstm

پرسپترون چندلایه



- مدل زبانی
- انواع مدل زبانی

- مدل‌های زبانی **n-gram**
- مدل‌های زبانی و بازنمایی **Tf-idf**
- بازنمایی واژگان **distributional semantics**
- معیار **PPMI**
- بازنمایی بر اساس **svd**
- شبکه عصبی **skipgram**
- مدل **Rnn-Lstm**

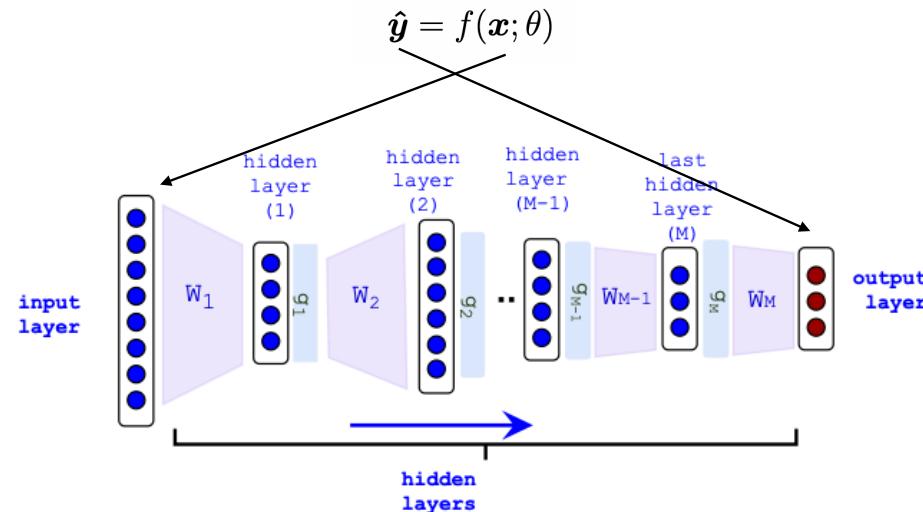
- مدل زبانی
- انواع مدل زبانی

- مدل‌های زبانی
- مدل‌های زبانی و بازنمایی
- بازنمایی واژگان
- بازنمایی متن Tf-idf
- ارزیابی بازنمایی
- نظریه distributional semantics
- معبار PPMI
- بازنمایی بر اساس svd
- شبکه عصبی

skipgram مدل

Rnn-Lstm

دوره شبکه‌های عصبی (مسیر forward)

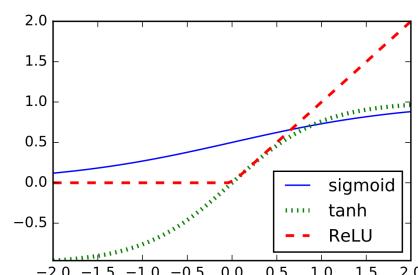


$$\hat{y} = f(\mathbf{x}; \theta) = f_M(\dots f_3(f_2(f_1(\mathbf{x}))))$$

هر لایه یک تبدیل غیر خطی

$$f_i(\mathbf{x}_i; \theta_i) = g_i(\mathbf{x}_i^T W_i + b_i)$$

$$\theta_i = \{W_i, b_i\}$$



گزینه‌های g_i

$$\text{sigmoid}(x) = \frac{e^x}{1 + e^x}$$

$$\tanh(x) = 2 \times \text{sgm}(x) - 1$$

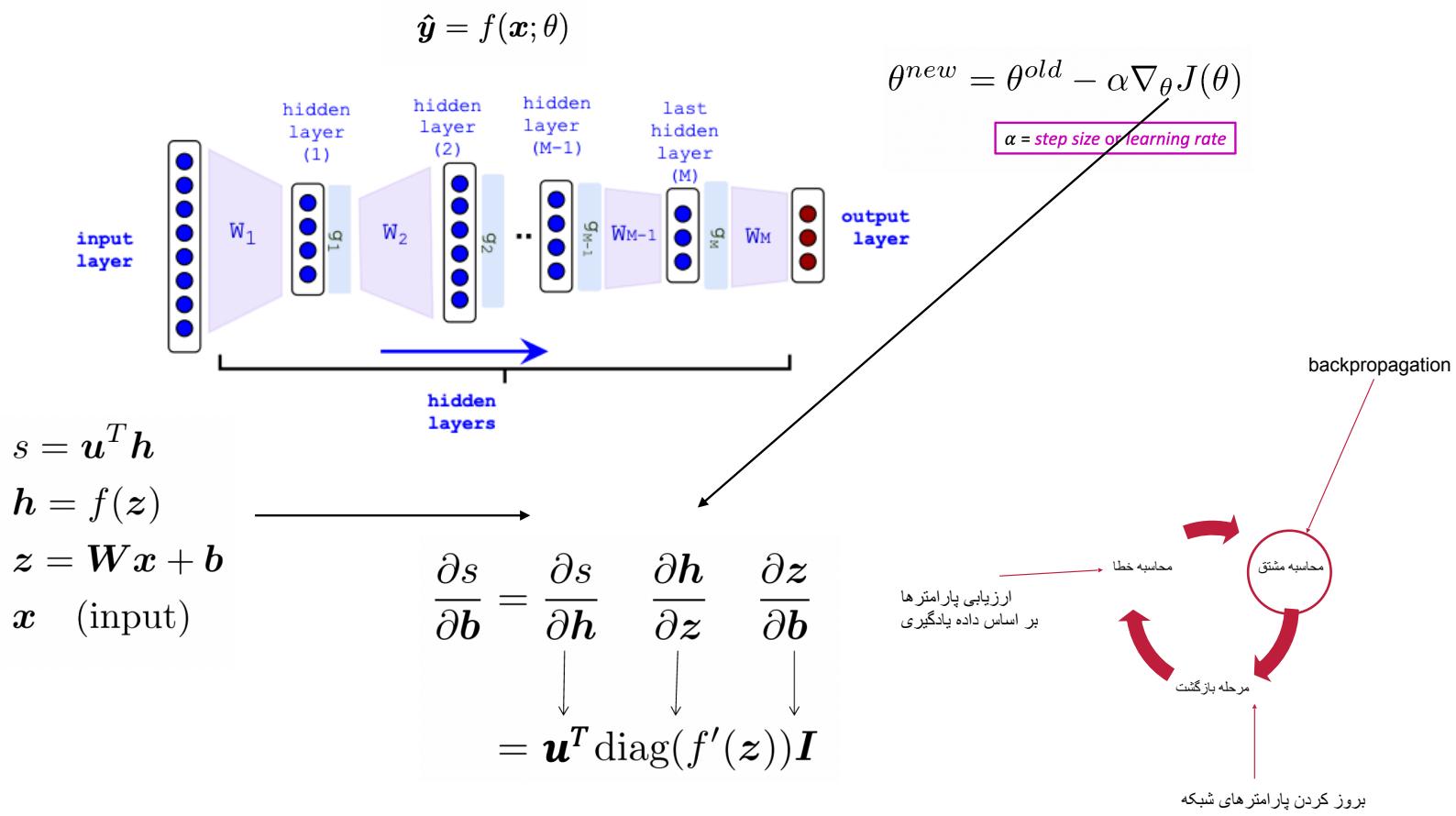
$$(x)_+ = \max(0, x)$$

a.k.a. "ReLU"

- مدل زبانی
- انواع مدل زبانی

- مدل‌های زبانی
- مدل‌های زبانی و بازنمایی
- بازنمایی واژگان
- بازنمایی متن Tf-idf
- ارزیابی بازنمایی
- نظریه distributional semantics
- معبار PPMI
- بازنمایی بر اساس svd
- شبکه عصبی
- skipgram مدل
- Rnn-Lstm

دوره شبکه‌های عصبی (مسیر backward)



- مدل زبانی
- انواع مدل زبانی

- مدل‌های زبانی **n-gram**
- مدل‌های زبانی و بازنمایی

- بازنمایی واژگان

- بازنمایی متن **Tf-idf**

- ارزیابی بازنمایی

- نظریه **distributional semantics**

- معیار **PPMI**

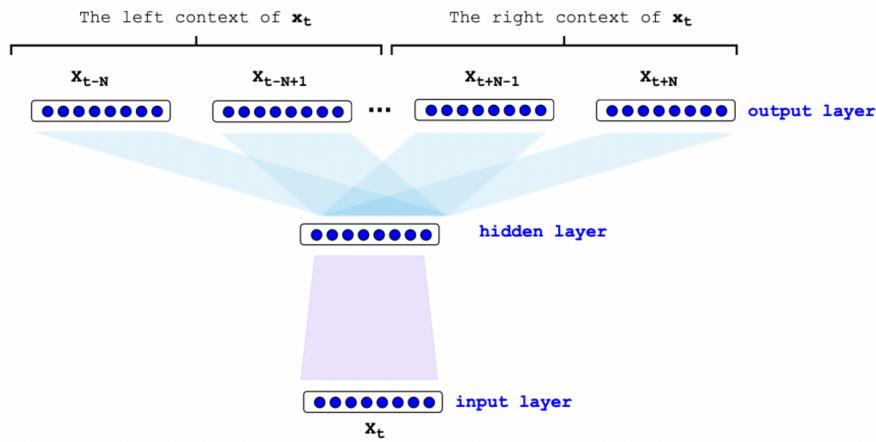
- بازنمایی بر اساس **svd**

- شبکه عصبی

- **skipgram** مدل

- **Rnn-Lstm**

Skip-gram مدل



هدف: بیشینه کردن likelihood زیر:

$$p(w_c | w_t; \theta) = \frac{e^{v_c \cdot v_t}}{\sum_{c' \in \mathcal{C}} e^{v_{c'} \cdot v_t}}$$

$$\sum_{t=1}^M \sum_{c \in [t-N, t+N]} \log p(w_c | w_t) \longrightarrow \left| \sum_{t=1}^T \left[\sum_{c \in [t-N, t+N]} -\log (1 + e^{-s(w_t, w_c)}) - \sum_{w_r \in \mathcal{N}_{t,c}} \log (1 + e^{s(w_t, w_r)}) \right] \right|$$

$$s(w_t, w_c) = v_t^\top \cdot v_c$$

بسم الله الرحمن الرحيم

پردازش زبانهای طبیعی

جلسه ۱۳

احسان الدین عسگری

اردیبهشت ۱۴۰۳

<http://language.ml/>

دانشکده مهندسی کامپیوتر

آزمایشگاه پردازش هوشمند متن و زبان و علوم انسانی محاسباتی

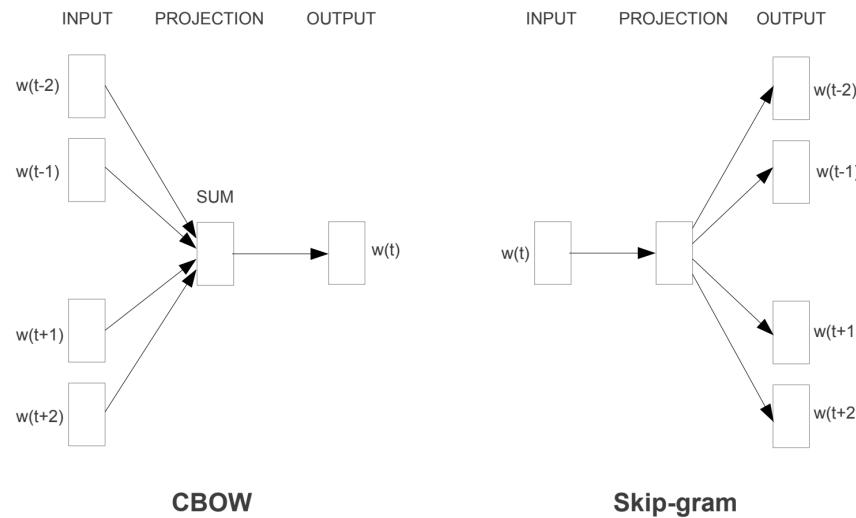
asgari@berkeley.edu



- مدل زبانی
- انواع مدل زبانی

- مدل‌های زبانی
- مدل‌های زبانی و بازنمایی
- بازنمایی واژگان
- بازنمایی متن
- ارزیابی بازنمایی
- نظریه distributional semantics
- معیار PPMI
- بازنمایی بر اساس svd
- شبکه عصبی

CBOW vs. Skip-gram مدل

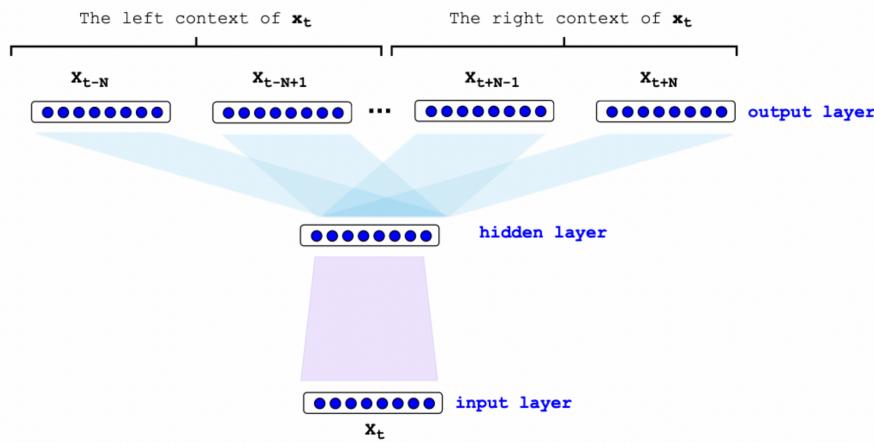


In the CBOW model, the distributed representations of context (or surrounding words) are combined to predict the word in the middle. In the Skip-gram model, the distributed representation of the input word is used to predict the context.

Mikolov, Tomas, Quoc V. Le, and Ilya Sutskever.

"**Exploiting similarities among languages for machine translation.**"
arXiv preprint arXiv:1309.4168 (2013).

Skip-gram مدل زبانی



هدف: بیشینه کردن likelihood زیر:

$$\sum_{t=1}^M \sum_{c \in [t-N, t+N]} \log p(w_c | w_t) \longrightarrow \arg \max_{\theta} \sum_{(w,c) \in D} \log p(c | w) = \sum_{(w,c) \in D} \left(\log e^{v_c \cdot v_w} - \log \sum_{c'} e^{v_{c'} \cdot v_w} \right)$$

تمام بافت‌های منفی

$$p(w_c | w_t; \theta) = \frac{e^{v_c \cdot v_t}}{\sum_{c' \in \mathcal{C}} e^{v_{c'} \cdot v_t}}$$

- مدل زبانی
- انواع مدل زبانی
- مدل‌های زبانی
- مدل‌های زبانی و بازنمایی
- بازنمایی واژگان
- بازنمایی متن
- ارزیابی بازنمایی
- نظریه distributional semantics
- معیار PPMI
- بازنمایی بر اساس svd
- شبکه عصبی

- مدل زبانی

- انواع مدل زبانی

- مدل‌های زبانی **n-gram**

- مدل‌های زبانی و بازنمایی

- بازنمایی واژگان

- بازنمایی متن **Tf-idf**

- ارزیابی بازنمایی

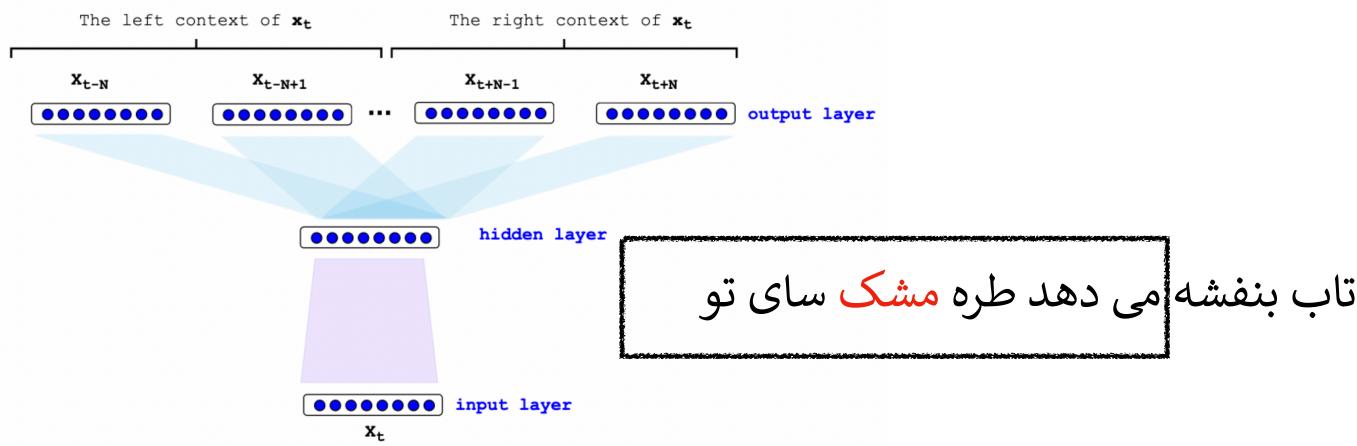
- نظریه **distributional semantics**

- معیار **PPMI**

- بازنمایی بر اساس **svd**

- شبکه عصبی

Skip-gram مدل



هدف: بیشینه کردن **likelihood** زیر با استفاده از نمونه‌گیری داده‌های منفی:

$$p(w_c | w_t; \theta) = \frac{e^{v_c \cdot v_t}}{\sum_{c' \in \mathcal{C}} e^{v_{c'} \cdot v_t}}$$

$$\sum_{t=1}^M \sum_{c \in [t-N, t+N]} \log p(w_c | w_t)$$

$$\longrightarrow \left| \sum_{t=1}^T \left[\sum_{c \in [t-N, t+N]} -\log (1 + e^{-s(w_t, w_c)}) - \sum_{w_r \in \mathcal{N}_{t,c}} \log (1 + e^{s(w_t, w_r)}) \right] \right|$$

$$s(w_t, w_c) = v_t^\top \cdot v_c$$

- مدل زبانی
- انواع مدل زبانی
- مدل‌های زبانی **n-gram**
- مدل‌های زبانی و بازنمایی
- بازنمایی واژگان
- بازنمایی **Tf-idf** متن
- ارزیابی بازنمایی
- نظریه distributional semantics
- معیار **PPMI**
- بازنمایی بر اساس **svd**
- شبکه عصبی

$$\begin{aligned}
 & \arg \max_{\theta} \prod_{(w,c) \in D} p(D = 1 \mid c, w; \theta) \prod_{(w,c) \in D'} p(D = 0 \mid c, w; \theta) \\
 &= \arg \max_{\theta} \prod_{(w,c) \in D} p(D = 1 \mid c, w; \theta) \prod_{(w,c) \in D'} (1 - p(D = 1 \mid c, w; \theta)) \\
 &= \arg \max_{\theta} \sum_{(w,c) \in D} \log p(D = 1 \mid c, w; \theta) + \sum_{(w,c) \in D'} \log(1 - p(D = 1 \mid c, w; \theta)) \\
 &= \arg \max_{\theta} \sum_{(w,c) \in D} \log \frac{1}{1 + e^{-v_c \cdot v_w}} + \sum_{(w,c) \in D'} \log \left(1 - \frac{1}{1 + e^{-v_c \cdot v_w}}\right) \\
 &= \arg \max_{\theta} \sum_{(w,c) \in D} \log \frac{1}{1 + e^{-v_c \cdot v_w}} + \sum_{(w,c) \in D'} \log \left(\frac{1}{1 + e^{v_c \cdot v_w}}\right)
 \end{aligned}$$

$\sigma(x) = \frac{1}{1 + e^{-x}}$ we get:

$$\begin{aligned}
 & \arg \max_{\theta} \sum_{(w,c) \in D} \log \frac{1}{1 + e^{-v_c \cdot v_w}} + \sum_{(w,c) \in D'} \log \left(\frac{1}{1 + e^{v_c \cdot v_w}}\right) \\
 &= \arg \max_{\theta} \sum_{(w,c) \in D} \log \sigma(v_c \cdot v_w) + \sum_{(w,c) \in D'} \log \sigma(-v_c \cdot v_w)
 \end{aligned}$$

زبان‌های چسبانشی یا Agglutinative

Muvaffakiyetsizleş(-mek)	(To) become unsuccessful
Muvaffakiyetsizleştir(-mek)	(To) make one unsuccessful
Muvaffakiyetsizleştirici	Maker of unsuccessful ones
Muvaffakiyetsizleştiricileş(-mek)	(To) become a maker of unsuccessful ones
Muvaffakiyetsizleştiricileştir(-mek)	(To) make one a maker of unsuccessful ones
Muvaffakiyetsizleştiricileştiriver(-mek)	(To) easily/quickly make one a maker of unsuccessful ones
Muvaffakiyetsizleştiricileştiriverebil(-mek)	(To) be able to make one easily/quickly a maker of unsuccessful ones
Muvaffakiyetsizleştiricileştiriveremeyebileceklerimi zdenmişsinizcesine	As though you happen to have been from among those whom we will not be able to easily/quickly make a maker of unsuccessful ones

- مدل زبانی
- انواع مدل زبانی

- مدل‌های زبانی **n-gram**
- مدل‌های زبانی و بازنمایی
- بازنمایی واژگان
- بازنمایی متن **Tf-idf**
- ارزیابی بازنمایی
- نظریه distributional semantics
- معیار **PPMI**
- بازنمایی بر اساس svd
- شبکه عصبی

ساختارهای مشترک

- باور
- باورناکردنی
- غیرقابل باور
- ناباورانه
- باورپذیر

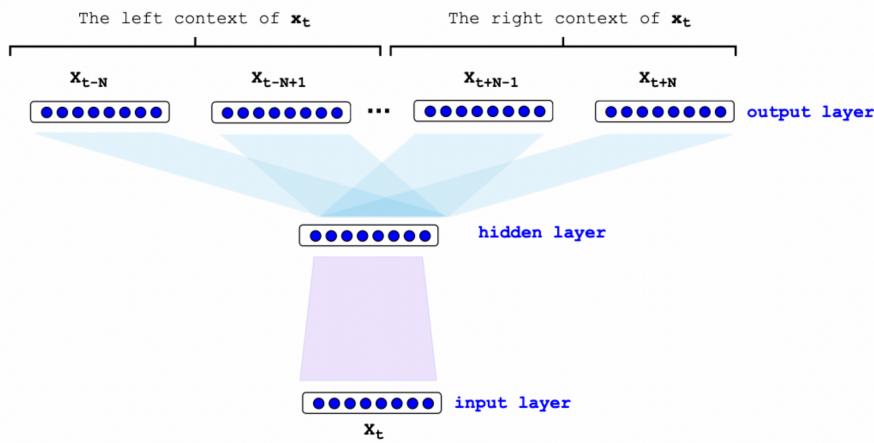
- مدل زبانی
- انواع مدل زبانی
 - مدل‌های زبانی **n-gram**
 - مدل‌های زبانی و بازنمایی
 - بازنمایی واژگان
 - بازنمایی متن **Tf-idf**
 - ارزیابی بازنمایی
 - نظریه distributional semantics
 - معیار **PPMI**
 - بازنمایی بر اساس **svd**
 - شبکه عصبی

Fasttext مدل



- مدل زبانی
- انواع مدل زبانی
 - مدل‌های زبانی **n-gram**
 - مدل‌های زبانی و بازنمایی
 - بازنمایی واژگان
- بازنمایی متن **Tf-idf**
- ارزیابی بازنمایی
- نظریه **distributional semantics**
- معیار **PPMI**
- بازنمایی بر اساس **svd**
- شبکه عصبی

Fasttext مدل



هدف: بیشینه کردن likelihood زیر:

$$p(w_c | w_t; \theta) = \frac{e^{v_c \cdot v_t}}{\sum_{c' \in \mathcal{C}} e^{v_{c'} \cdot v_t}}$$

$$\log p(w_c | w_t) \longrightarrow \left[\sum_{t=1}^T \left[-\log \left(1 + e^{-s(w_t, w_c)} \right) - \sum_{w_r \in \mathcal{N}_{t,c}} \log \left(1 + e^{s(w_t, w_r)} \right) \right] \right]$$

$$s(w_t, w_c) = \sum_{x \in \mathcal{S}_{w_t}} v_x^\top v_c$$

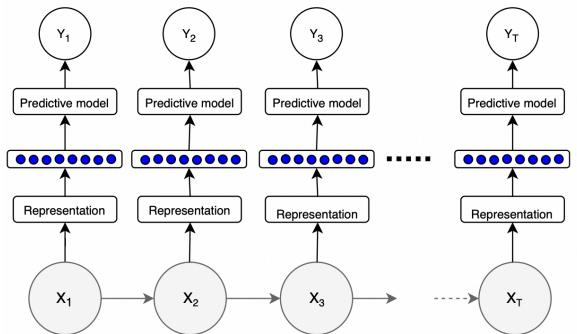
- مدل زبانی
- انواع مدل زبانی
- مدل‌های زبانی **n-gram**
- مدل‌های زبانی و بازنمایی
- بازنمایی واژگان
- بازنمایی متن **Tf-idf**
- ارزیابی بازنمایی
- distributional semantics نظریه
- **PPMI** معیار
- بازنمایی بر اساس **svd**
- شبکه عصبی

ارزیابی بازنمایی واژگان

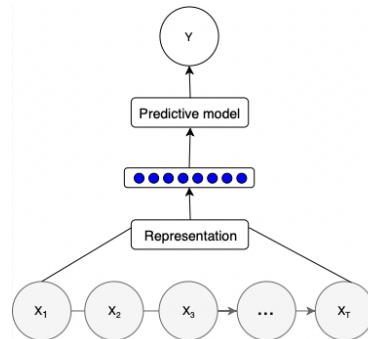
- ارزیابی ذاتی (ایترینسیک)
- ارزیابی مستقیم در مساله (اکسترینسیک)

- مدل زبانی
- انواع مدل زبانی
 - مدل‌های زبانی **n-gram**
 - مدل‌های زبانی و بازنمایی
 - بازنمایی واژگان
- بازنمایی متن **Tf-idf**
- ارزیابی بازنمایی
- نظریه **distributional semantics**
- معیار **PPMI**
- بازنمایی بر اساس **svd**
- شبکه عصبی

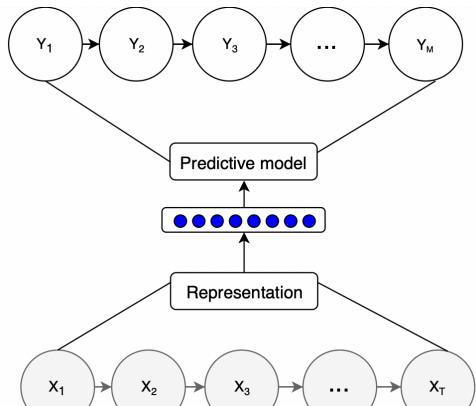
کاربرد امبدینگ‌های استاتیک



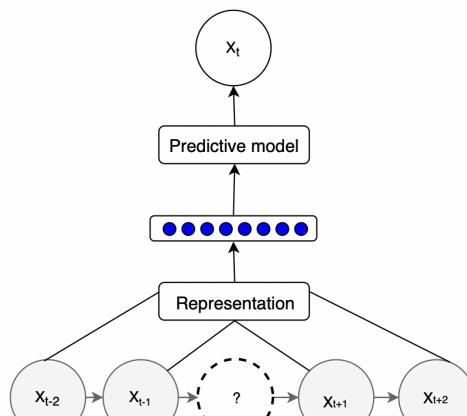
برچسب زنی



طبقه‌بندی



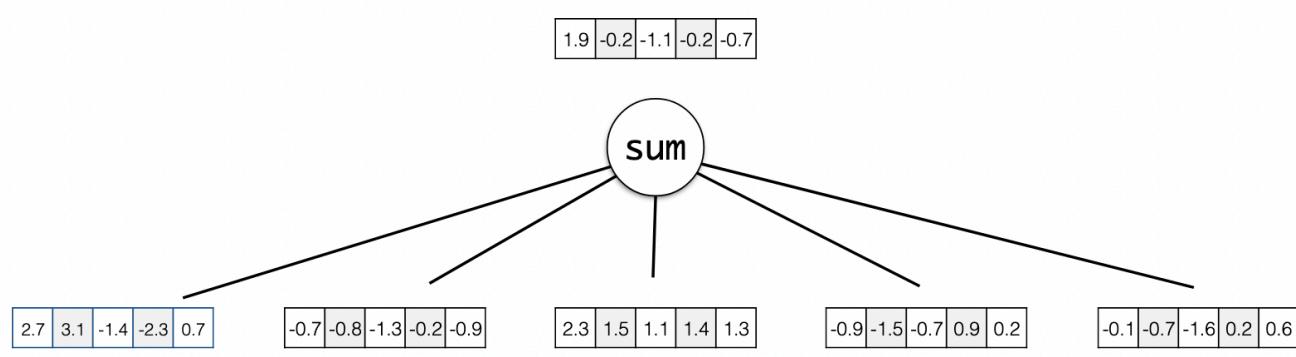
تولید متن



مدل زبانی

- مدل زبانی
- انواع مدل زبانی
 - مدل‌های زبانی **n-gram**
 - مدل‌های زبانی و بازنمایی
 - بازنمایی واژگان
 - بازنمایی متن **Tf-idf**
 - ارزیابی بازنمایی
 - نظریه **distributional semantics**
 - معیار **PPMI**
 - بازنمایی بر اساس **svd**
 - شبکه عصبی

امبدينگ سند



- مدل زبانی
- انواع مدل زبانی

• مدل‌های زبانی **n-gram**

• مدل‌های زبانی و بازنمایی

• بازنمایی واژگان

• بازنمایی متن **Tf-idf**

• ارزیابی بازنمایی

• نظریه distributional semantics

• معیار **PPMI**

• بازنمایی بر اساس **svd**

• شبکه عصبی

امبدینگ متن

Published as a conference paper at ICLR 2017

A SIMPLE BUT TOUGH-TO-BEAT BASELINE FOR SENTENCE EMBEDDINGS

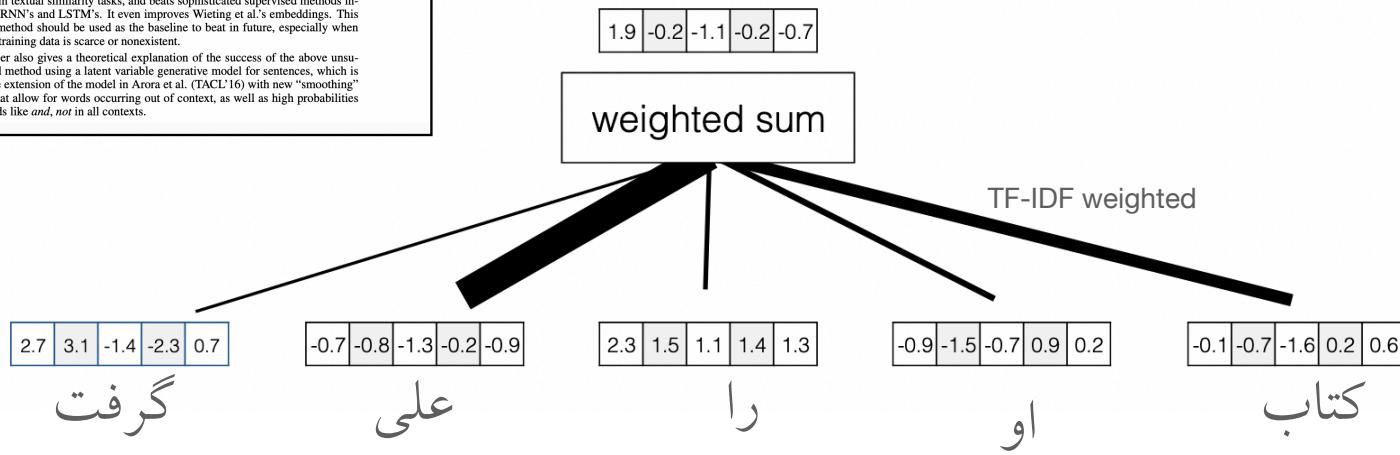
Sanjeev Arora, Yingyu Liang, Tengyu Ma
Princeton University
{arora,yingyul,tengyu}@cs.princeton.edu

ABSTRACT

The success of neural network methods for computing word embeddings has motivated methods for generating semantic embeddings of longer pieces of text, such as sentences and paragraphs. Surprisingly, Wieting et al (ICLR'16) showed that some complicated methods are outperformed, especially in out-of-domain (transfer learning) settings, by simpler methods involving mild retraining of word embeddings and basic linear regression. The method of Wieting et al. requires retraining with a substantial labeled dataset such as Paraphrase Database (Gankevitch et al., 2013).

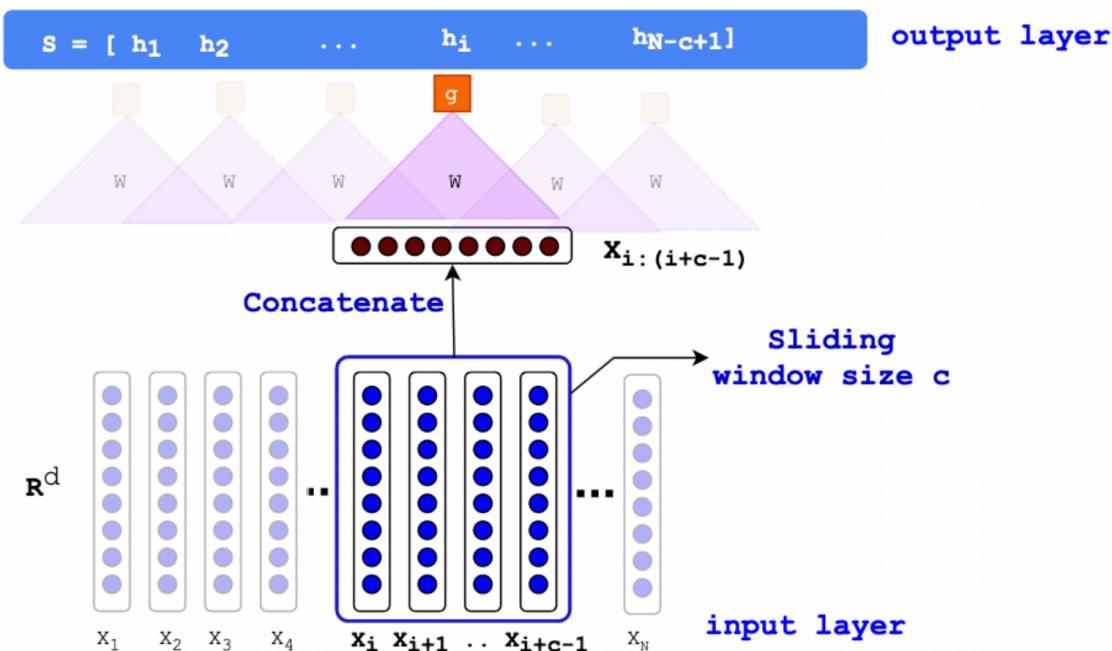
The current paper goes further, showing that the following completely unsupervised sentence embedding is a formidable baseline: Use word embeddings computed using one of the popular methods on unlabeled corpus like Wikipedia, represent the sentence by a weighted average of the word vectors, and then modify them a bit using PCA/SVD. This weighting improves performance by about 10% to 30% in textual similarity tasks, and beats sophisticated supervised methods including RNN's and LSTM's. It even improves Wieting et al.'s embeddings. This simple method should be used as the baseline to beat in future, especially when labeled training data is scarce or nonexistent.

The paper also gives a theoretical explanation of the success of the above unsupervised method using a latent variable generative model for sentences, which is a simple extension of the model in Arora et al. (TACL'16) with new “smoothing” terms that allow for words occurring out of context, as well as high probabilities for words like *and*, *not* in all contexts.



- مدل زبانی
- انواع مدل زبانی
- مدل‌های زبانی **n-gram**
- مدل‌های زبانی و بازنمایی
- بازنمایی واژگان
- بازنمایی متن **Tf-idf**
- ارزیابی بازنمایی
- نظریه distributional semantics
- معیار **PPMI**
- بازنمایی بر اساس svd
- شبکه عصبی

لایه کانولوشن



$$h_i = g(W^T \mathbf{x}_{i:(i+c-1)} + b)$$

امبیینگ جمله به طول N

- مدل زبانی
- انواع مدل زبانی
- مدل‌های زبانی **n-gram**
- مدل‌های زبانی و بازنمایی
- بازنمایی واژگان
- بازنمایی متن **Tf-idf**
- ارزیابی بازنمایی
- نظریه **distributional semantics**
- معیار **PPMI**
- بازنمایی بر اساس **svd**
- شبکه عصبی

- مدل زبانی

- انواع مدل زبانی

- مدل‌های زبانی **n-gram**

- مدل‌های زبانی و بازنمایی

- بازنمایی واژگان

- بازنمایی متن **Tf-idf**

- ارزیابی بازنمایی

- نظریه **distributional semantics**

- معیار **PPMI**

- بازنمایی بر اساس **svd**

- شبکه عصبی



بسم الله الرحمن الرحيم

پردازش زبانهای طبیعی

جلسه ۱۴

احسان الدین عسگری

اردیبهشت ۱۴۰۳

<http://language.ml/>

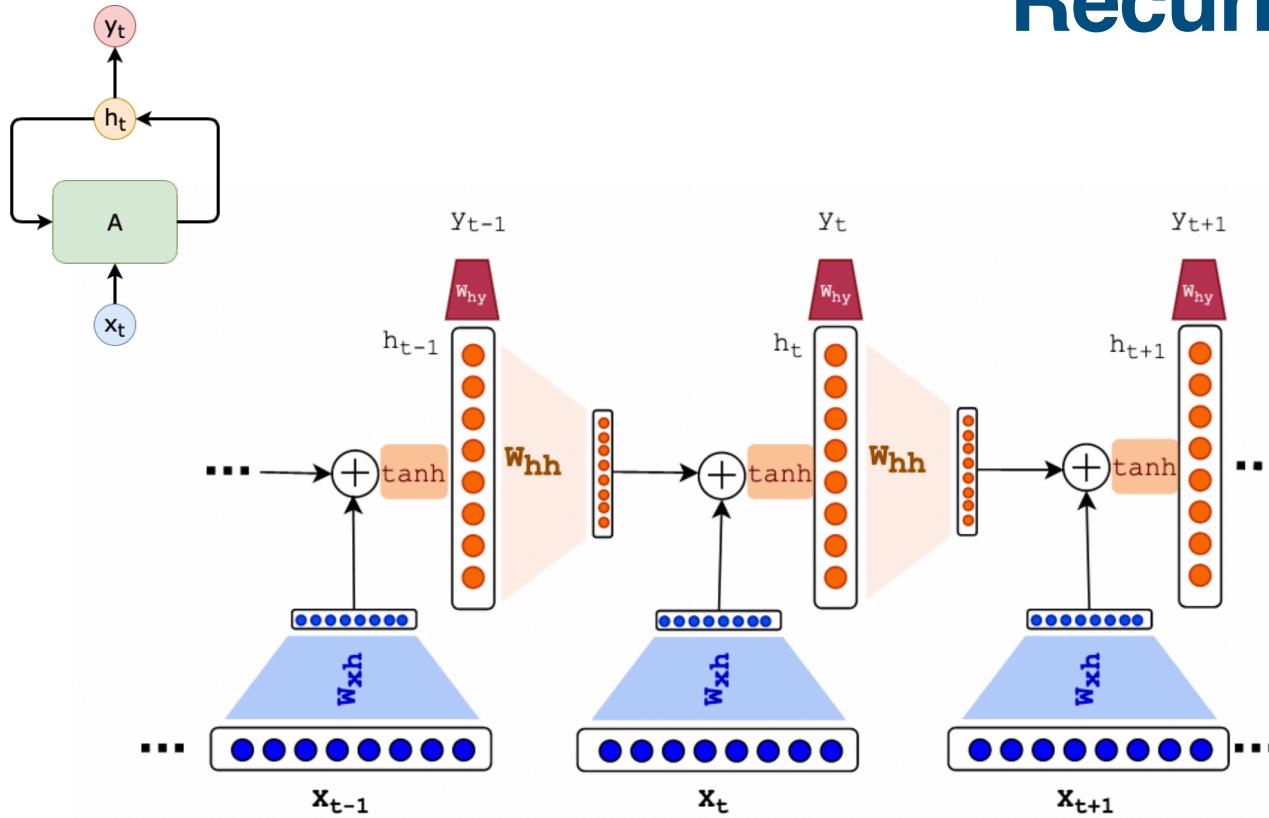
دانشکده مهندسی کامپیوتر

آزمایشگاه پردازش هوشمند متن و زبان و علوم انسانی محاسباتی

asgari@berkeley.edu



لایه Recurrent

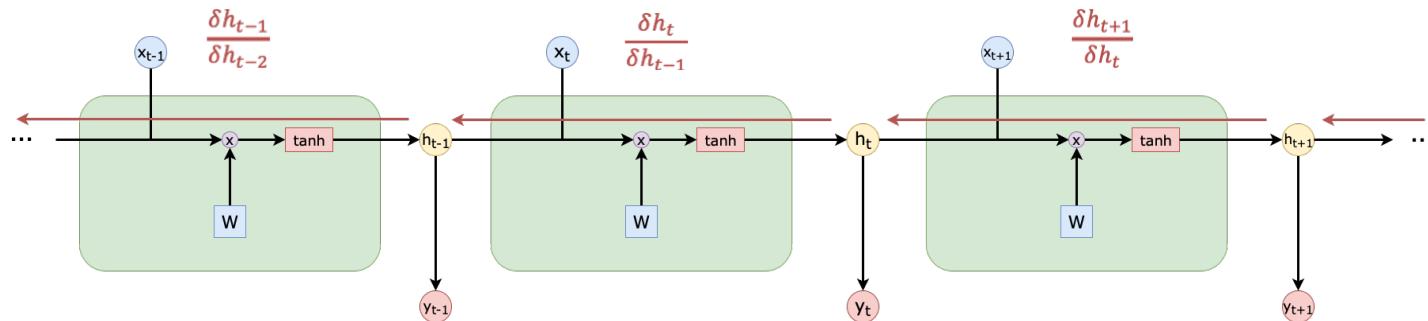
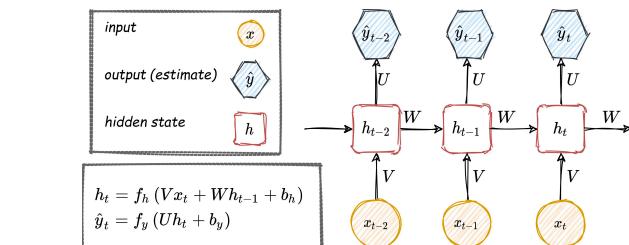


$$h_t = \tanh(W_{hh}h_{t-1} + x_t^T W_{xh})$$

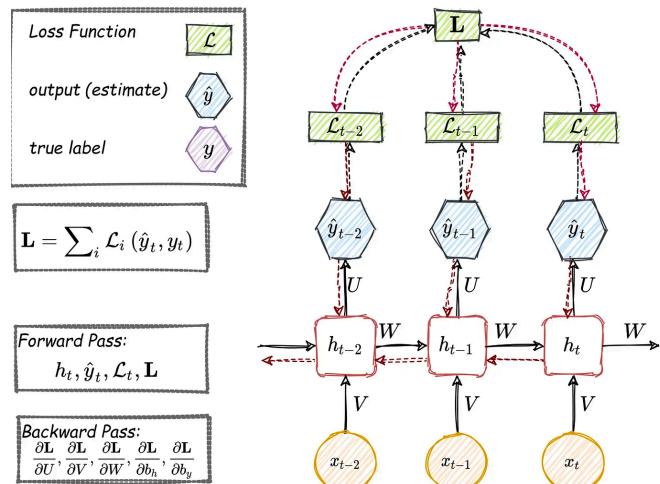
$$\hat{y}_t = \text{softmax}(h_t^T W_{hy}),$$

- مدل زبانی
- انواع مدل زبانی
- مدل‌های زبانی **n-gram**
- مدل‌های زبانی و بازنمایی
- بازنمایی واژگان
- **Tf-idf** بازنمایی متن
- ارزیابی بازنمایی
- نظریه **distributional semantics**
- **PPMI** معیار
- بازنمایی بر اساس **svd**
- شبکه عصبی

شبکه Recurrent



Backpropagation در طول زمان



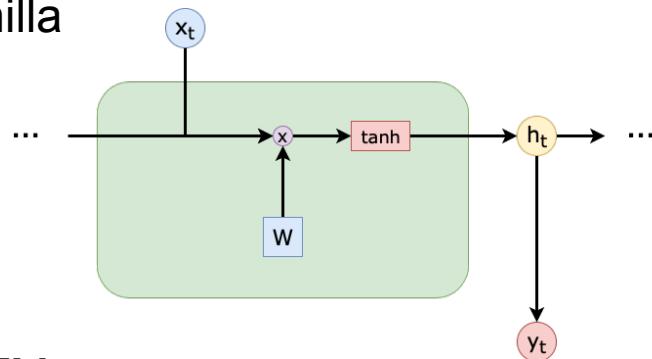
$$\frac{\partial L}{\partial W} = \sum_{i=0}^T \frac{\partial L_i}{\partial W} \propto \sum_{k=0}^T \left(\prod_{i=k+1}^y \frac{\partial h_i}{\partial h_{i-1}} \right) \frac{\partial h_k}{\partial W}$$

- 1. Vanishing gradient** $\left\| \frac{\partial h_i}{\partial h_{i-1}} \right\|_2 < 1$
- 2. Exploding gradient** $\left\| \frac{\partial h_i}{\partial h_{i-1}} \right\|_2 > 1$

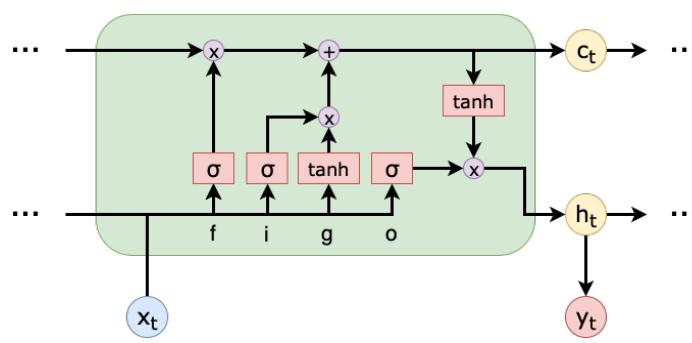


RNN در مقایسه با LSTM

Vanilla



LSTM



$$h_t = f_w(h_{t-1}, x_t)$$

$$h_t = \tanh(W \begin{pmatrix} h_{t-1} \\ x_t \end{pmatrix} + b)$$

$$y_t = W_{hy} h_t$$

$$f_t = \sigma(W_f \begin{pmatrix} h_{t-1} \\ x_t \end{pmatrix} + b_f)$$

$$i_t = \sigma(W_i \begin{pmatrix} h_{t-1} \\ x_t \end{pmatrix} + b_i)$$

$$o_t = \sigma(W_o \begin{pmatrix} h_{t-1} \\ x_t \end{pmatrix} + b_o)$$

$$\tilde{c}_t = \tanh(W_{\tilde{c}} \begin{pmatrix} h_{t-1} \\ x_t \end{pmatrix} + b_{\tilde{c}})$$

- مدل زبانی
- انواع مدل زبانی

• مدل‌های زبانی

• مدل‌های زبانی و بازنمایی

• بازنمایی واژگان

• بازنمایی متن

• ارزیابی بازنمایی

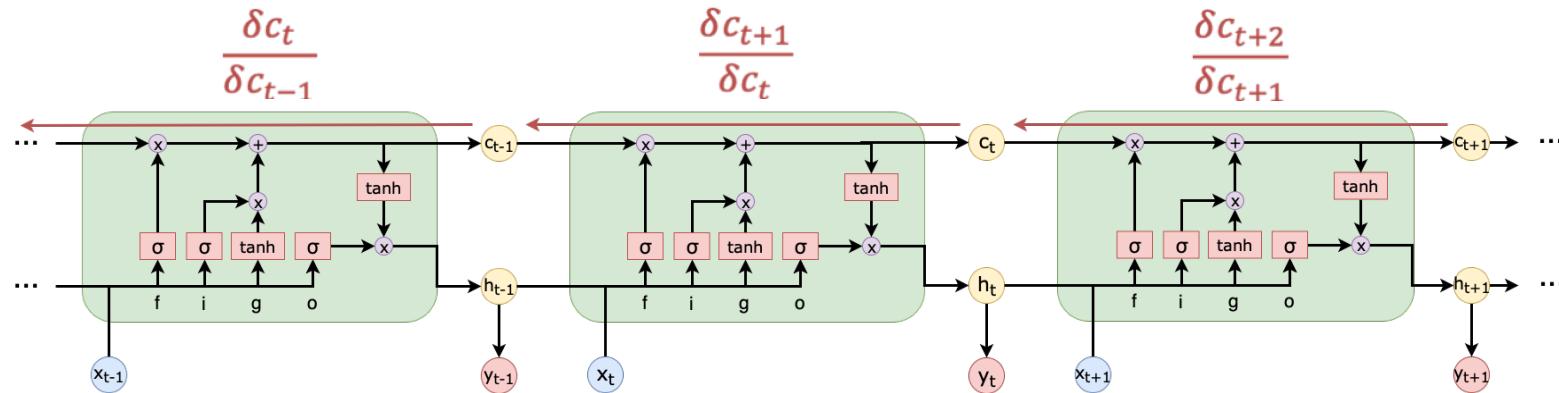
• نظریه distributional semantics

• معیار PPMI

• بازنمایی بر اساس svd

• شبکه عصبی

LSTM



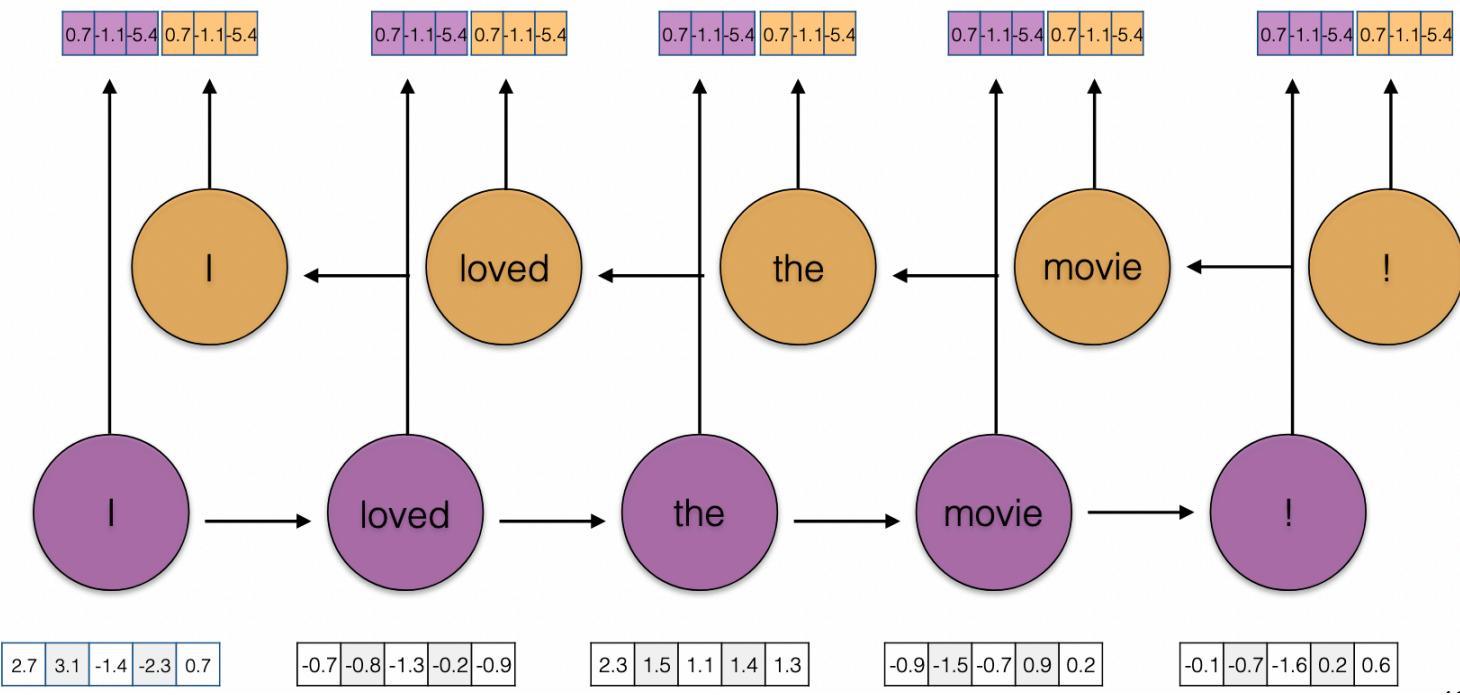
- LSTMs introduce a cell state, which is maintained along with the hidden state
- The cell state serves as long term memory, while the hidden state functions as working memory or short term memory
- Four gates are responsible for modifying the cell state and the hidden state
 - **Forget gate** filters previous information
 - **Input gate** filters new information
 - **Input modulation gate** processes new information
 - **Output gate** filters the cell state information for the hidden state
- Widely used for RNN tasks, e.g. Google Translate, Siri, Amazon Alexa

- مدل زبانی
- انواع مدل زبانی

- مدل‌های زبانی
- مدل‌های زبانی و بازنمایی
- بازنمایی واژگان
- بازنمایی متن
- ارزیابی بازنمایی
- نظریه distributional semantics

- معیار PPMI
- بازنمایی بر اساس svd
- شبکه عصبی

LSTM



- مدل زبانی

- انواع مدل زبانی

- مدل‌های زبانی

- مدل‌های زبانی و بازنمایی

- بازنمایی واژگان

- بازنمایی متن

- ارزیابی بازنمایی

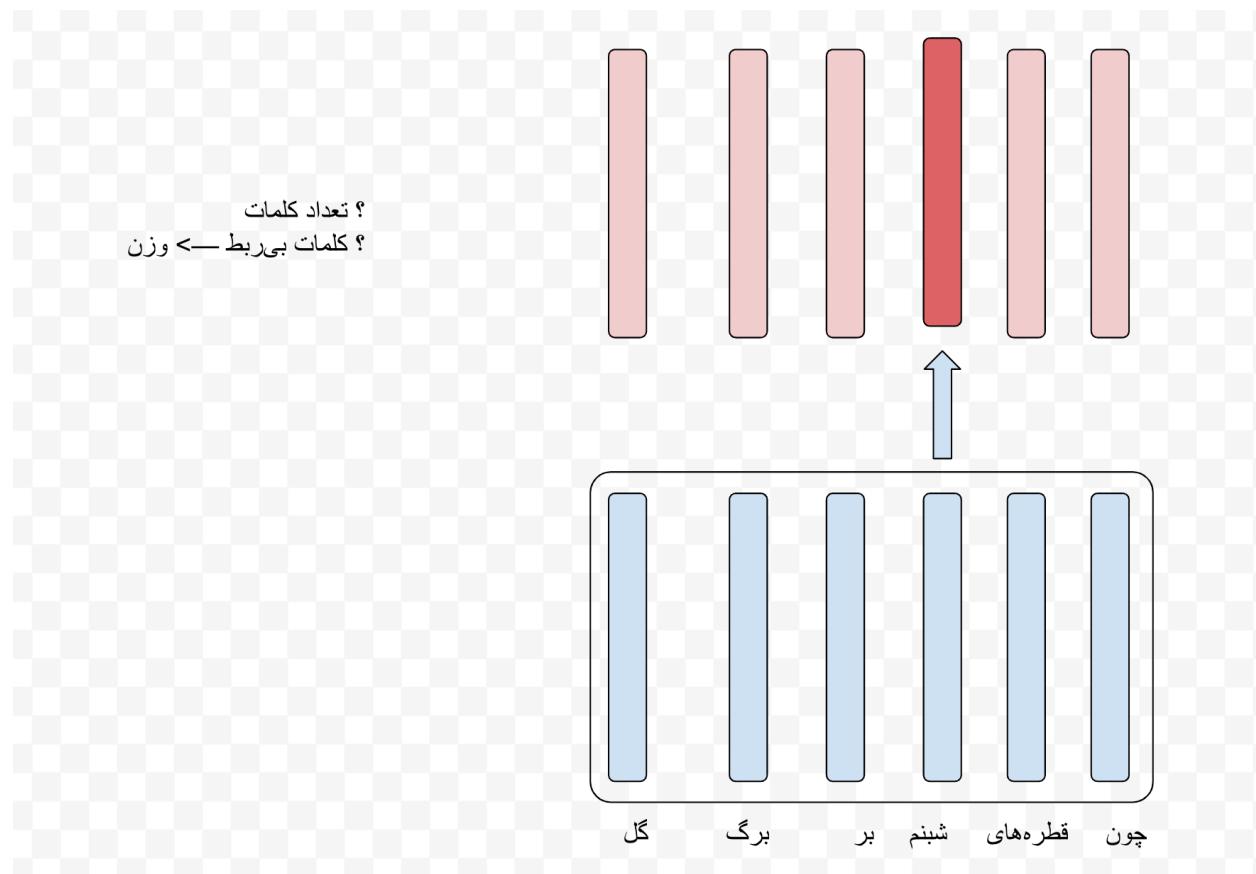
- نظریه distributional semantics

- معیار PPMI

- بازنمایی بر اساس svd

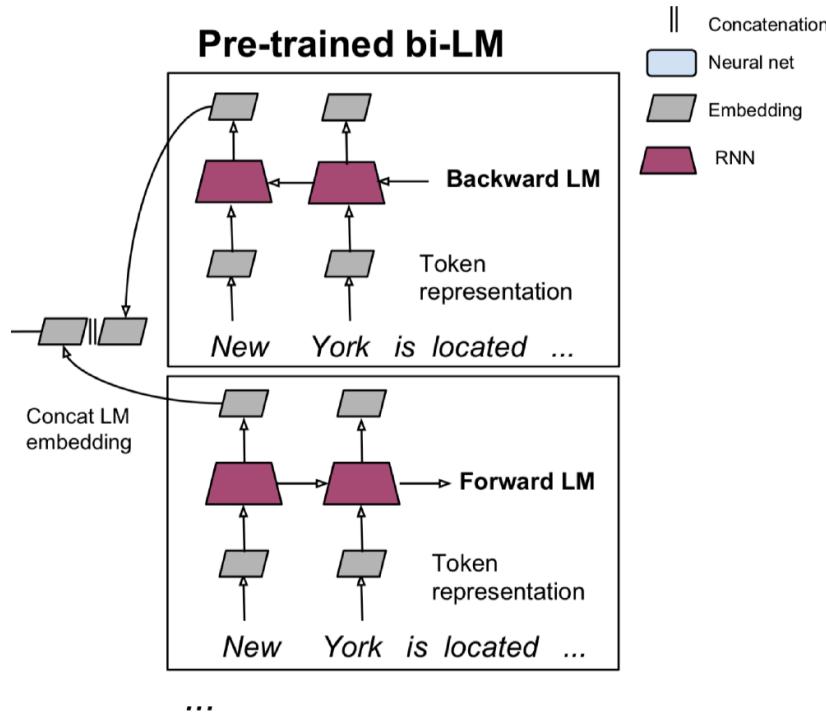
- شبکه عصبی

امبدينگ‌های سیاق محور



- بازنمایی بر اساس مدل زبانی
- دوره جلسه قبل
- امبدینگ‌های استاتیک
- **skip-gram**
- **Fasttext**
- **Doc Embedding**
- امبدینگ‌های مبتنی بر سیاق
- **ELMO**
- **ULMFit**
- **BERT**

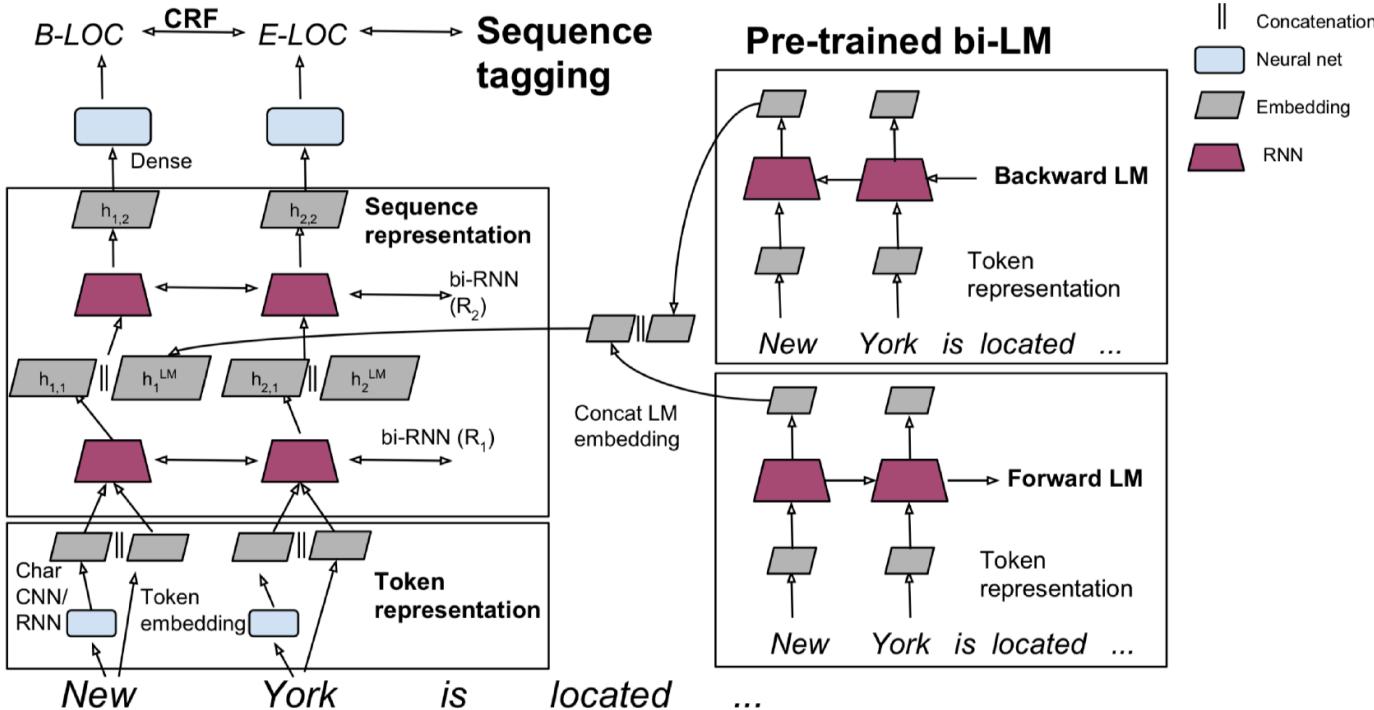
امبدينگ‌های سیاق محور



- مدل زبانی
- انواع مدل زبانی
 - مدل‌های زبانی
 - مدل‌های زبانی و بازنمایی
 - بازنمایی واژگان
 - بازنمایی متن
 - ارزیابی بازنمایی
 - نظریه distributional semantics
 - معیار PPMI
 - بازنمایی بر اساس svd
 - شبکه عصبی

امبدينگ‌های سیاق محور

TagLM - PreELMO



- مدل زبانی
- انواع مدل زبانی
- مدل‌های زبانی
- مدل‌های زبانی و بازنمایی
- بازنمایی واژگان
- بازنمایی متن
- ارزیابی بازنمایی
- نظریه distributional semantics
- معیار PPMI
- بازنمایی بر اساس svd
- شبکه عصبی

- مدل زبانی
- انواع مدل زبانی

• مدل‌های زبانی

• مدل‌های زبانی و بازنمایی

• بازنمایی واژگان

• بازنمایی متن

• ارزیابی بازنمایی

• نظریه

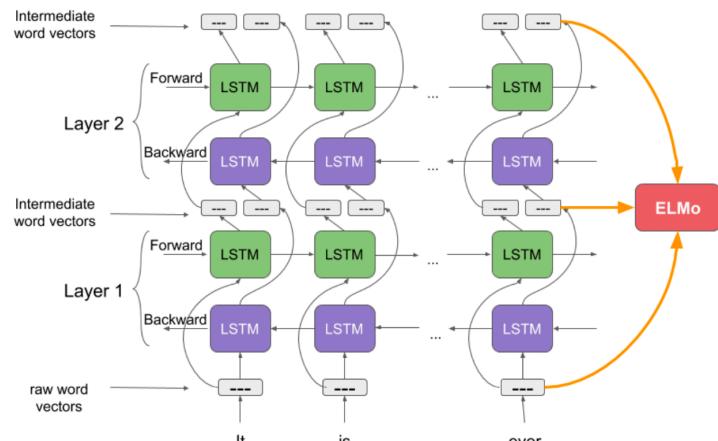
• معیار PPMI

• بازنمایی بر اساس svd

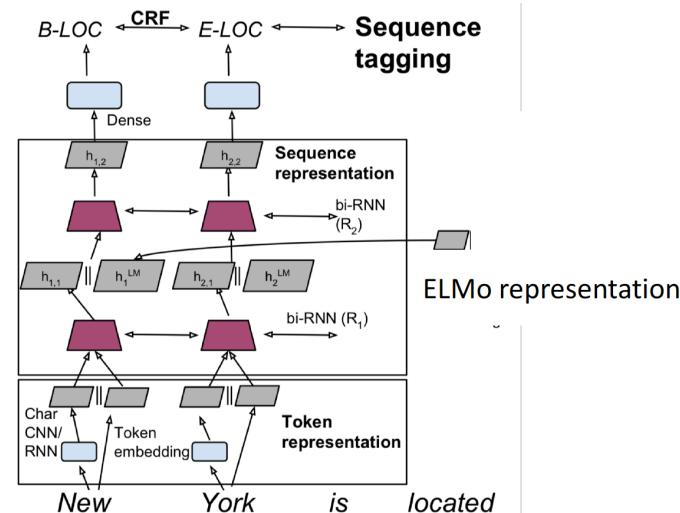
• شبکه عصبی

امبینگ‌های سیاق محور

ELMO: Deep contextualized word representations



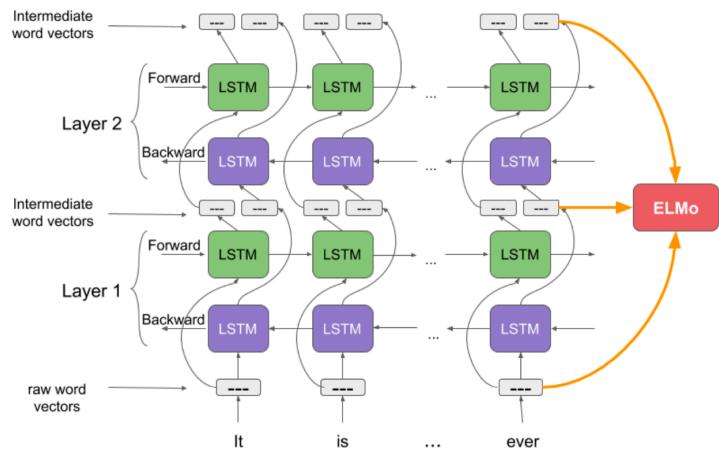
$$\text{ELMo}_k^{task} = E(R_k; \Theta^{task}) = \gamma^{task} \sum_{j=0}^L s_j^{task} \mathbf{h}_{k,j}^{LM}$$



$$\mathbf{h}_{k,1} = [\vec{\mathbf{h}}_{k,1}; \overleftarrow{\mathbf{h}}_{k,1}; \mathbf{h}_k^{LM}]$$

امبدینگ‌های سیاق‌محور

ELMO: Deep contextualized word representations



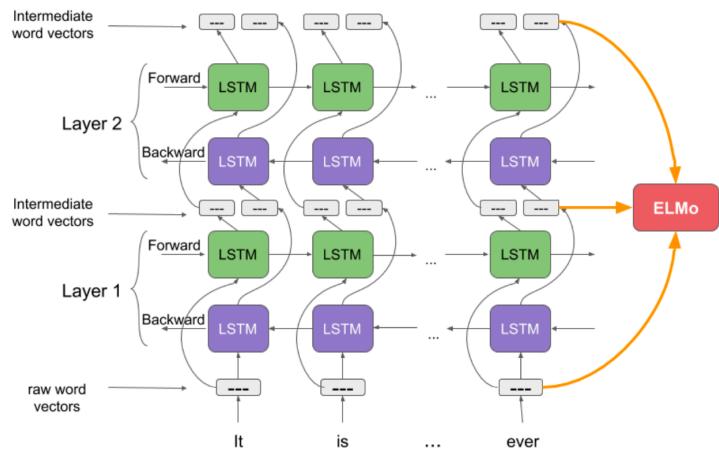
لایه‌های پایین برای مسائل ساده‌تر
Part-of-speech tagging, syntactic dependencies, NER
لایه‌های بالایی برای مسائل پیچیده‌تر
Sentiment, Semantic role labeling, question answering

$$\text{ELMo}_k^{task} = E(R_k; \Theta^{task}) = \gamma^{task} \sum_{j=0}^L s_j^{task} \mathbf{h}_{k,j}^{LM}$$

- مدل زبانی
- انواع مدل زبانی
- مدل‌های زبانی **n-gram**
- مدل‌های زبانی و بازنمایی
- بازنمایی واژگان
- بازنمایی متن **Tf-idf**
- ارزیابی بازنمایی
- نظریه **distributional semantics**
- معیار **PPMI**
- بازنمایی بر اساس **svd**
- شبکه عصبی

امبدینگ‌های سیاق‌محور

ELMO: Deep contextualized word representations



لایه‌های پایین برای مسائل ساده تر
Part-of-speech tagging, syntactic dependencies, NER
لایه‌های بالایی برای مسائل پیچیده‌تر
Sentiment, Semantic role labeling, question answering

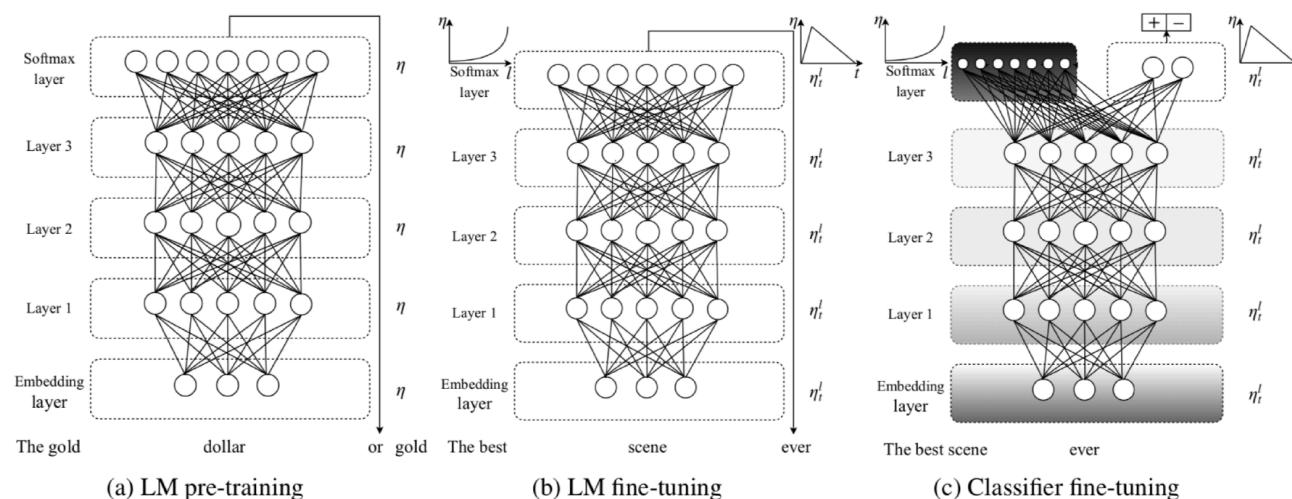
$$\text{ELMo}_k^{task} = E(R_k; \Theta^{task}) = \gamma^{task} \sum_{j=0}^L s_j^{task} \mathbf{h}_{k,j}^{LM}$$

- مدل زبانی
- انواع مدل زبانی
- مدل‌های زبانی **n-gram**
- مدل‌های زبانی و بازنمایی
- بازنمایی واژگان
- بازنمایی متن **Tf-idf**
- ارزیابی بازنمایی
- نظریه **distributional semantics**
- معیار **PPMI**
- بازنمایی بر اساس **svd**
- شبکه عصبی

امبدينگ‌های سیاق محور

ULMFit

- Train LM on big general domain corpus (use biLM)
- Tune LM on target task data
- Fine-tune as classifier on target task



- مدل زبانی
- انواع مدل زبانی
- مدل‌های زبانی
- مدل‌های زبانی و بازنمایی
- بازنمایی واژگان
- بازنمایی متن
- ارزیابی بازنمایی
- نظریه distributional semantics
- معیار PPMI
- بازنمایی بر اساس svd
- شبکه عصبی

آمان
ماه
هر بیش
خورسید
نایمید

How to contextualize the fixed embeddings?



...که دگرنزه عشق خورشید و نه هر ماه دارم



فروزنده ماه و نایمید و هر ...



فروشست از بگار و نقش ماه هر و آبانش

بسم الله الرحمن الرحيم

پردازش زبانهای طبیعی

جلسه ۱۵

احسان الدین عسگری

اردیبهشت ۱۴۰۳

<http://language.ml/>

دانشکده مهندسی کامپیوتر

آزمایشگاه پردازش هوشمند متن و زبان و علوم انسانی محاسباتی

asgari@berkeley.edu



– Self-attention



Attention

Self-Attention Idea

Input embeddings

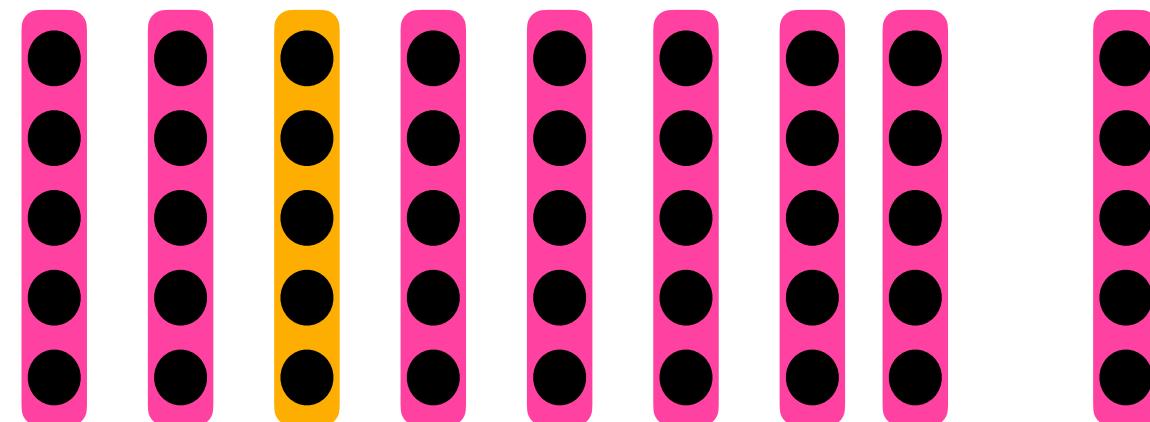
$$x_1, x_2, \dots, x_n$$

Skipgram

Output embeddings

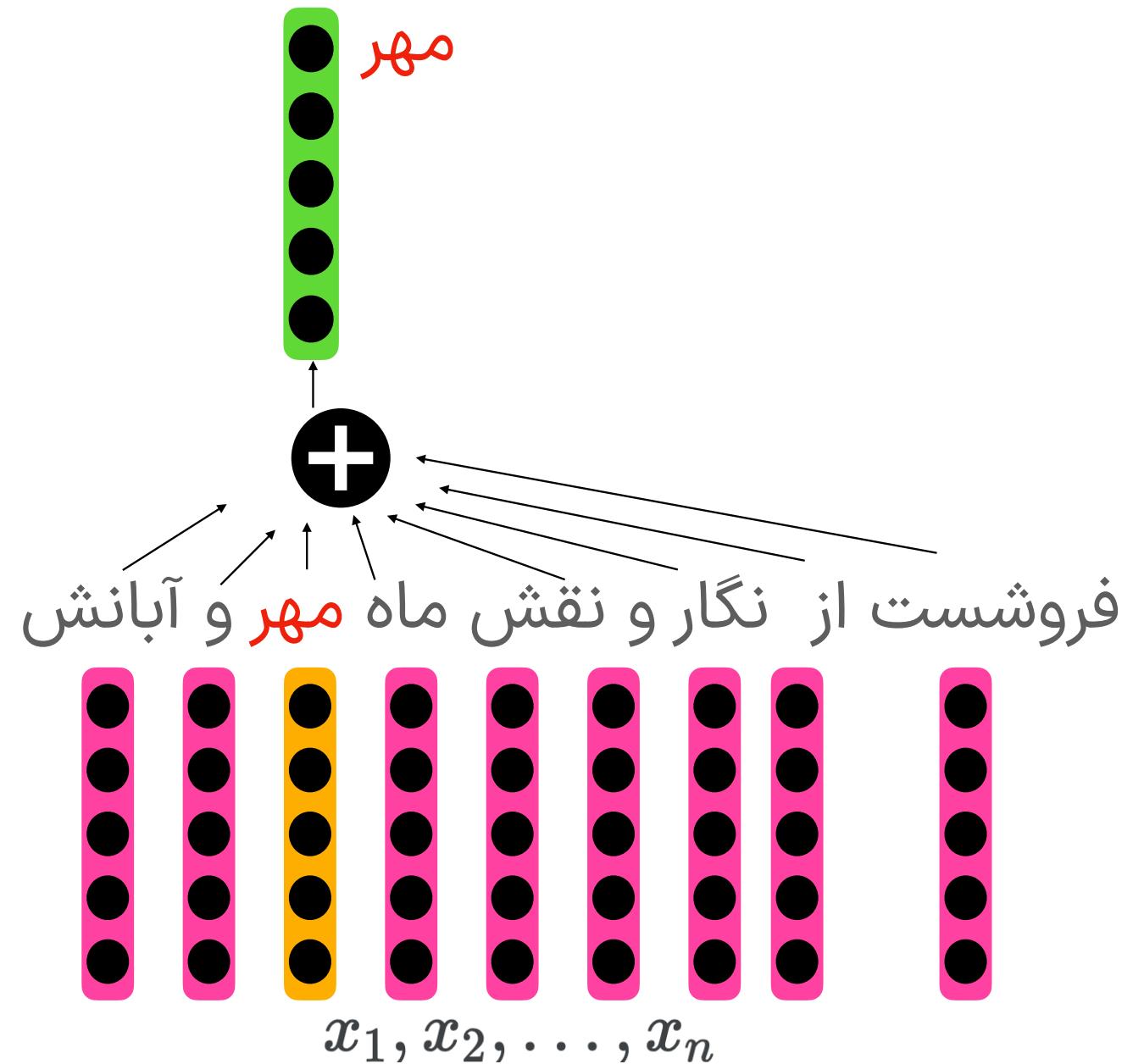
$$y_1, y_2, \dots, y_n$$

فروشست از نگار و نقش ماه **مهر** و آبانش



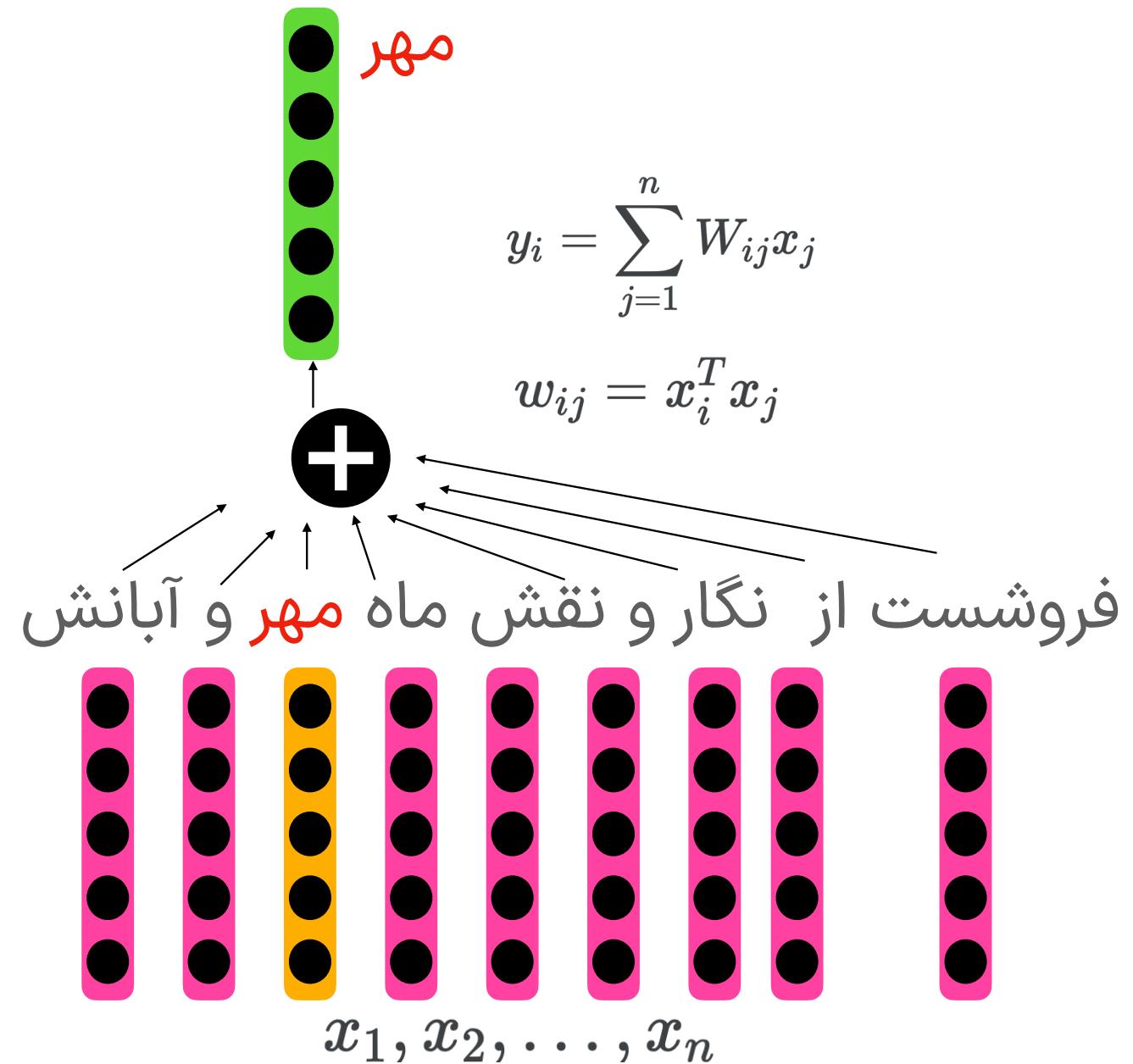
Attention

Self-Attention Idea



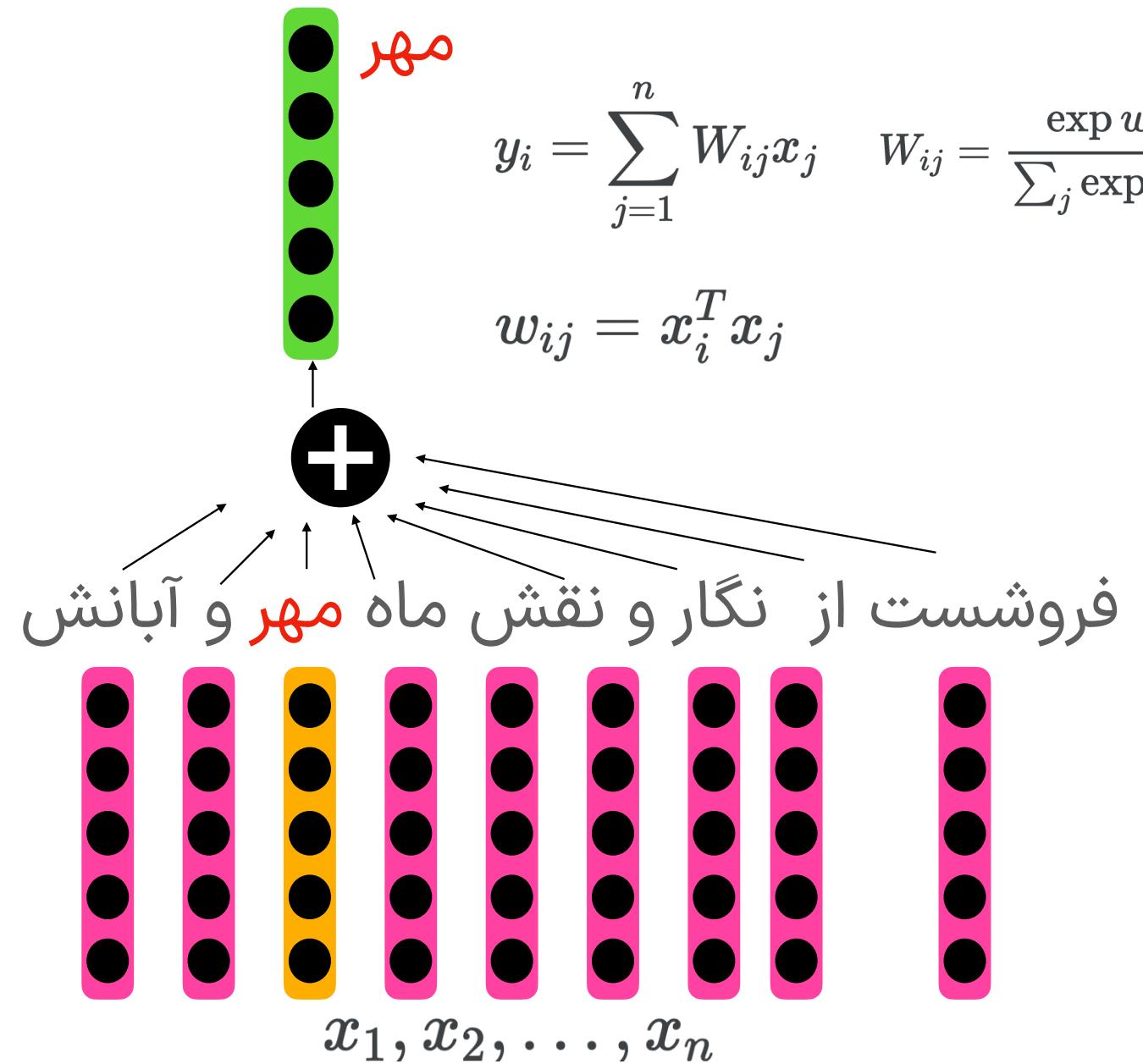
Attention

Self-Attention



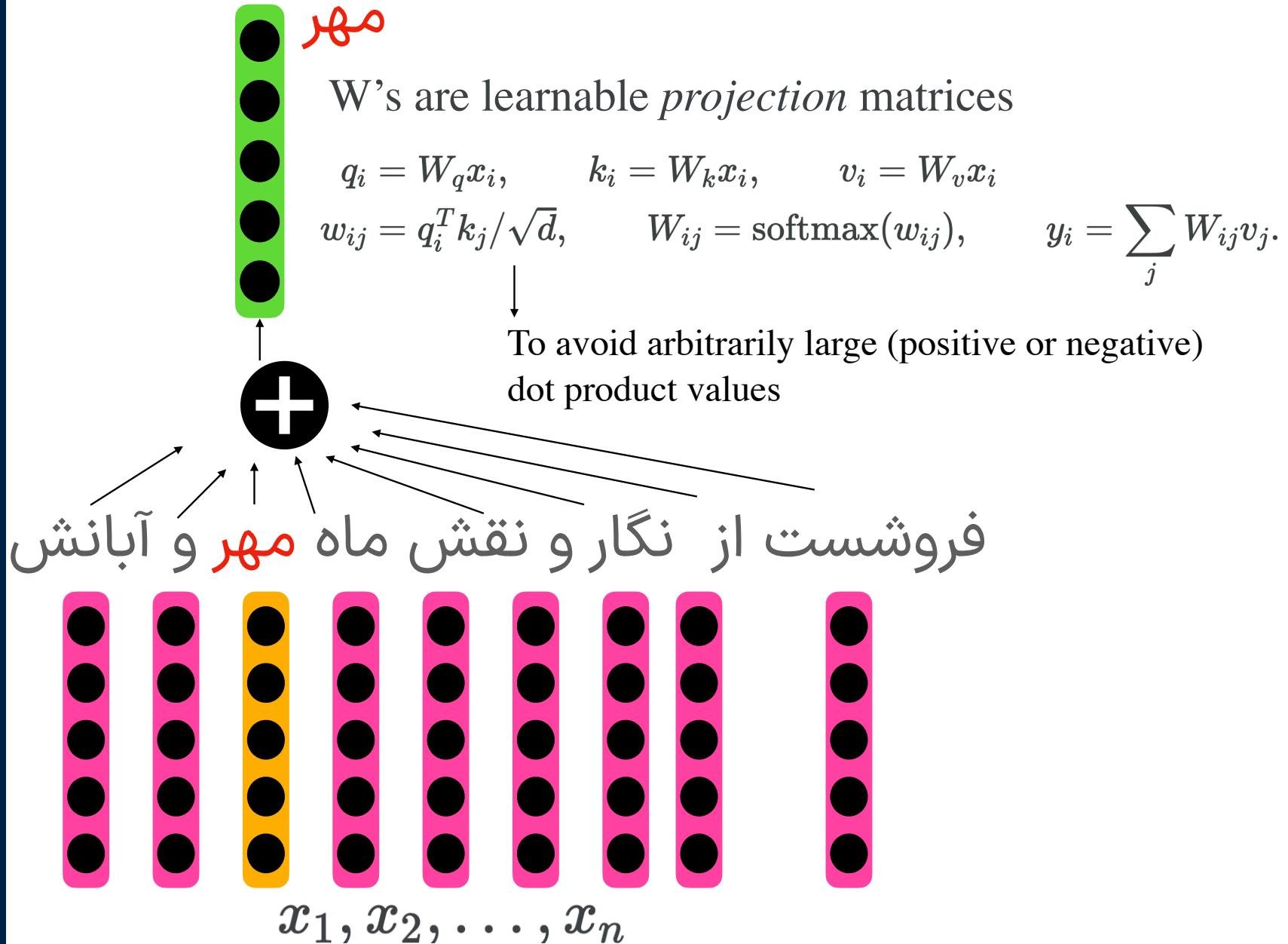
Attention

Self-Attention



Attention

Attention



WHY \sqrt{d} ?

$$q_i = W_q x_i, \quad k_i = W_k x_i, \quad v_i = W_v x_i$$

$$w_{ij} = q_i^T k_j / \sqrt{d}, \quad W_{ij} = \text{softmax}(w_{ij}), \quad y_i = \sum_j W_{ij} v_j.$$

Assume that \mathbf{q} and \mathbf{k} are unit vectors with dimension \mathbf{d} , whose dimensions are independent RV with the following properties:

$$\begin{aligned} E[q_i] &= E[k_i] = 0 \\ \text{var}[q_i] &= \text{var}[k_i] = 1 \end{aligned}$$

$$\begin{aligned} E[q \cdot k] &= E\left[\sum_{i=1}^d q_i k_i\right] & \text{var}[q \cdot k] &= \text{var}\left[\sum_{i=1}^d q_i k_i\right] \\ &= \sum_{i=1}^d E[q_i k_i] & &= \sum_{i=1}^d \text{var}[q_i k_i] \\ &= \sum_{i=1}^d E[q_i] E[k_i] & &= \sum_{i=1}^d \text{var}[q_i] \text{var}[k_i] \\ &= 0 & &= \sum_{i=1}^d 1 \\ & & &= d \end{aligned} \quad \longrightarrow \quad w_{ij} = \frac{q_i^T k_j - \mu}{\sigma} \\ & & &= \frac{q_i^T k_j}{\sqrt{d}}$$

More detailed proof: <https://github.com/BAI-Yeqi/Statistical-Properties-of-Dot-Product/blob/master/proof.pdf>

Attention

Attention

- Inputs: a query q and a set of key-value (k - v) pairs to an output
 - All presented as vectors
 - Output is weighted sum of values
 - Weight of each value: inner product of query and corresponding key

$$A(q, K, V) = \sum_i \frac{e^{q \cdot k_i}}{\sum_j e^{q \cdot k_j}} v_i$$

بسم الله الرحمن الرحيم

پردازش زبانهای طبیعی

جلسه ۱۶

احسان الدین عسگری

اردیبهشت ۱۴۰۳

<http://language.ml/>

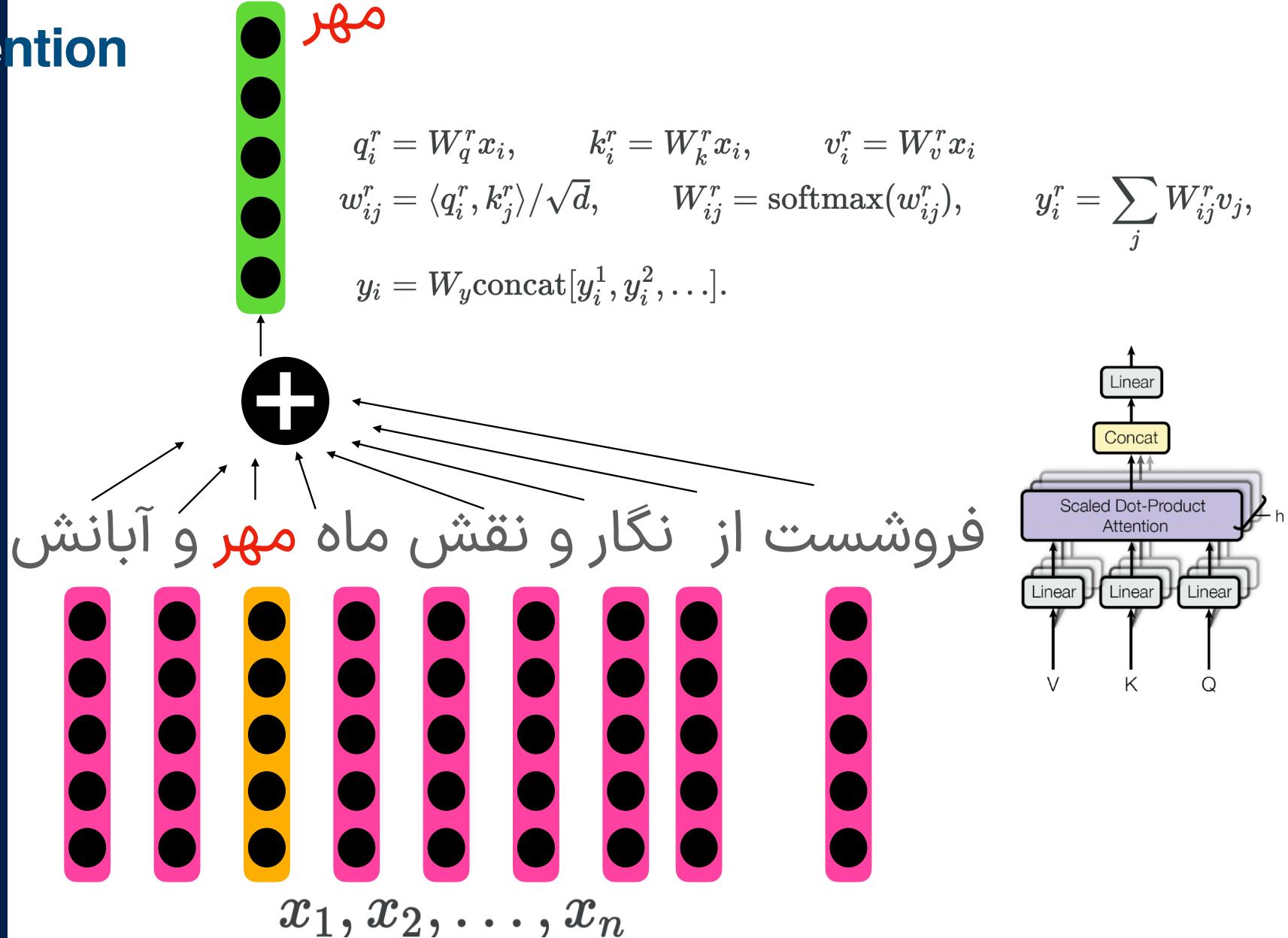
دانشکده مهندسی کامپیوتر

آزمایشگاه پردازش هوشمند متن و زبان و علوم انسانی محاسباتی

asgari@berkeley.edu



Multihead Attention



Attention

Matrix Attention

$$\begin{array}{ccc} \mathbf{X} & \times & \mathbf{W}^Q \\ \begin{matrix} \text{---} \\ \text{---} \\ \text{---} \\ \text{---} \end{matrix} & \times & \begin{matrix} \text{---} \\ \text{---} \\ \text{---} \\ \text{---} \end{matrix} \\ & & = \\ & & \begin{matrix} \text{---} \\ \text{---} \\ \text{---} \\ \text{---} \end{matrix} \end{array}$$
$$\begin{array}{ccc} \mathbf{X} & \times & \mathbf{W}^K \\ \begin{matrix} \text{---} \\ \text{---} \\ \text{---} \\ \text{---} \end{matrix} & \times & \begin{matrix} \text{---} \\ \text{---} \\ \text{---} \\ \text{---} \end{matrix} \\ & & = \\ & & \begin{matrix} \text{---} \\ \text{---} \\ \text{---} \\ \text{---} \end{matrix} \end{array}$$
$$\begin{array}{ccc} \mathbf{X} & \times & \mathbf{W}^V \\ \begin{matrix} \text{---} \\ \text{---} \\ \text{---} \\ \text{---} \end{matrix} & \times & \begin{matrix} \text{---} \\ \text{---} \\ \text{---} \\ \text{---} \end{matrix} \\ & & = \\ & & \begin{matrix} \text{---} \\ \text{---} \\ \text{---} \\ \text{---} \end{matrix} \end{array}$$

Attention

Matrix Attention

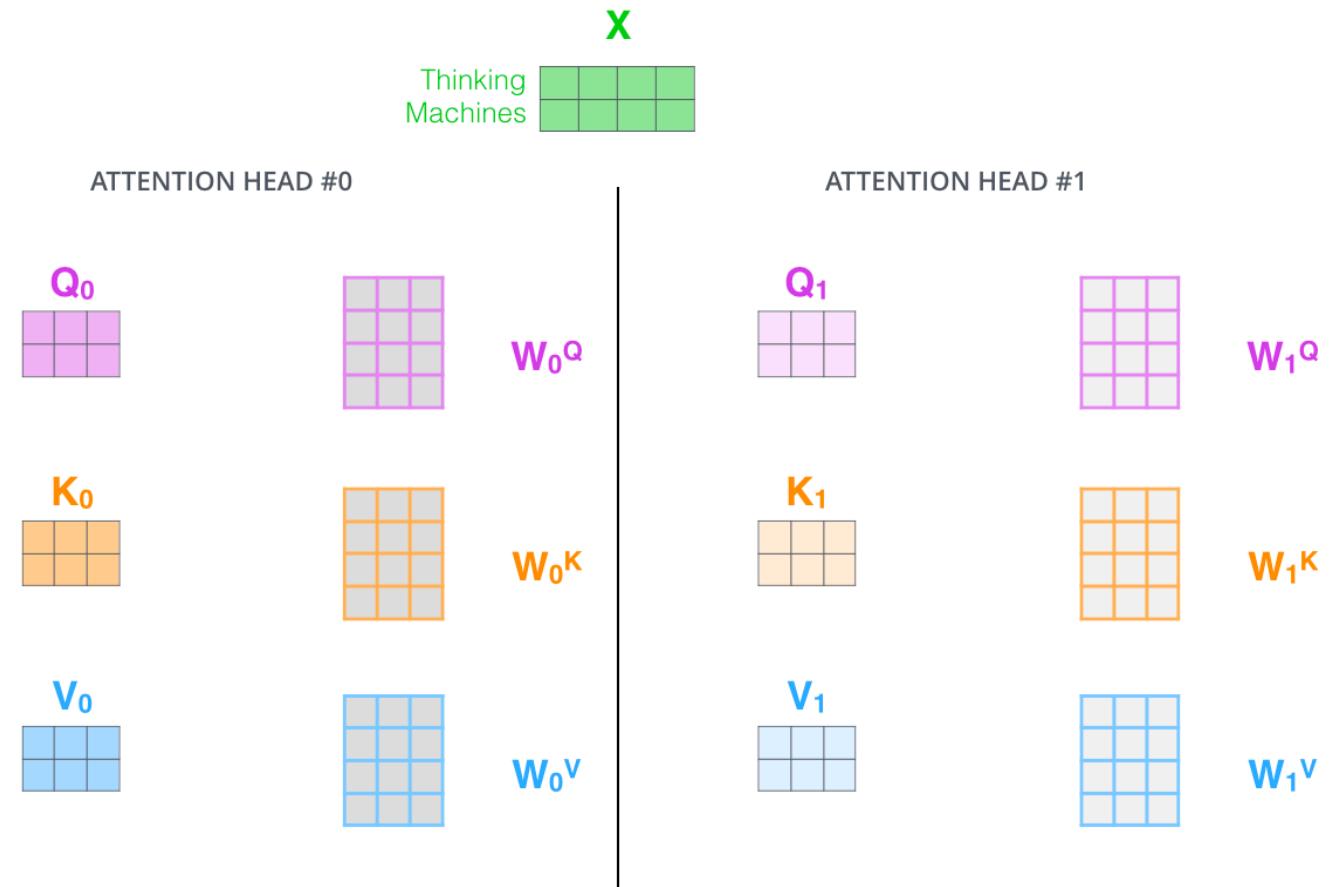
$$\text{softmax}\left(\frac{\mathbf{Q} \times \mathbf{K}^T}{\sqrt{d_k}}\right) \mathbf{V}$$

= \mathbf{Z}

The diagram illustrates the Matrix Attention mechanism. It starts with three matrices: \mathbf{Q} (purple), \mathbf{K}^T (orange), and \mathbf{V} (blue). The \mathbf{Q} matrix is multiplied by the transpose of the \mathbf{K} matrix (\mathbf{K}^T). This product is then divided by the square root of the dimension d_k . The result is passed through a softmax function to produce attention weights, which are then multiplied by the \mathbf{V} matrix to produce the final output \mathbf{Z} .

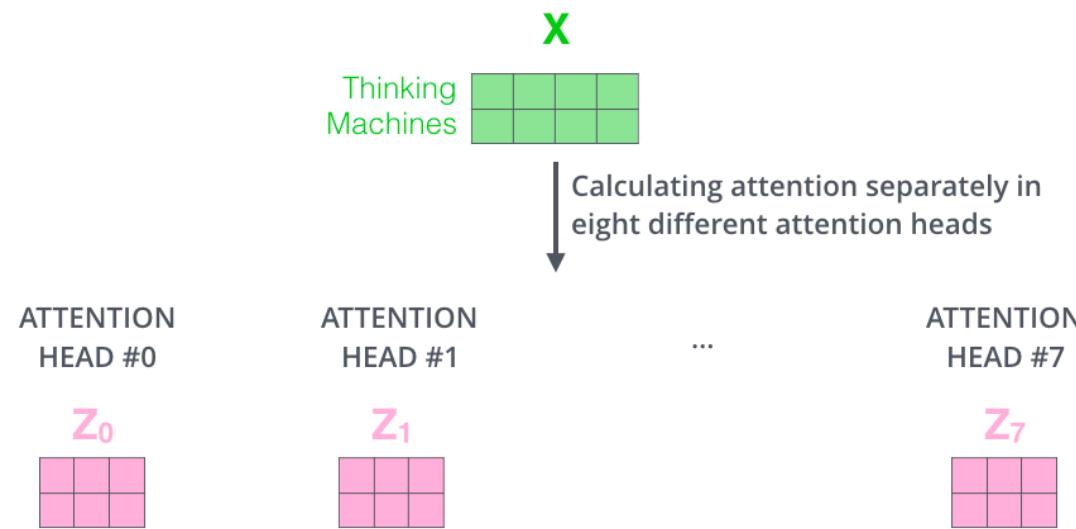
Attention

Matrix Attention



Attention

Matrix Attention



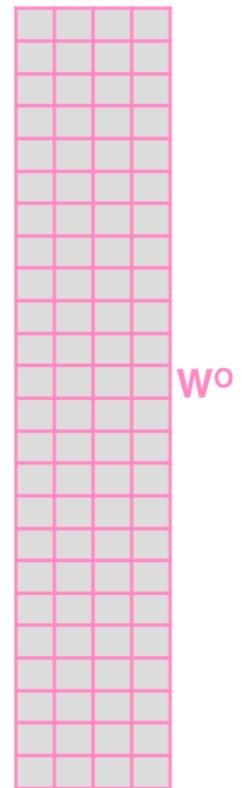
Attention

1) Concatenate all the attention heads



2) Multiply with a weight matrix W^o that was trained jointly with the model

x



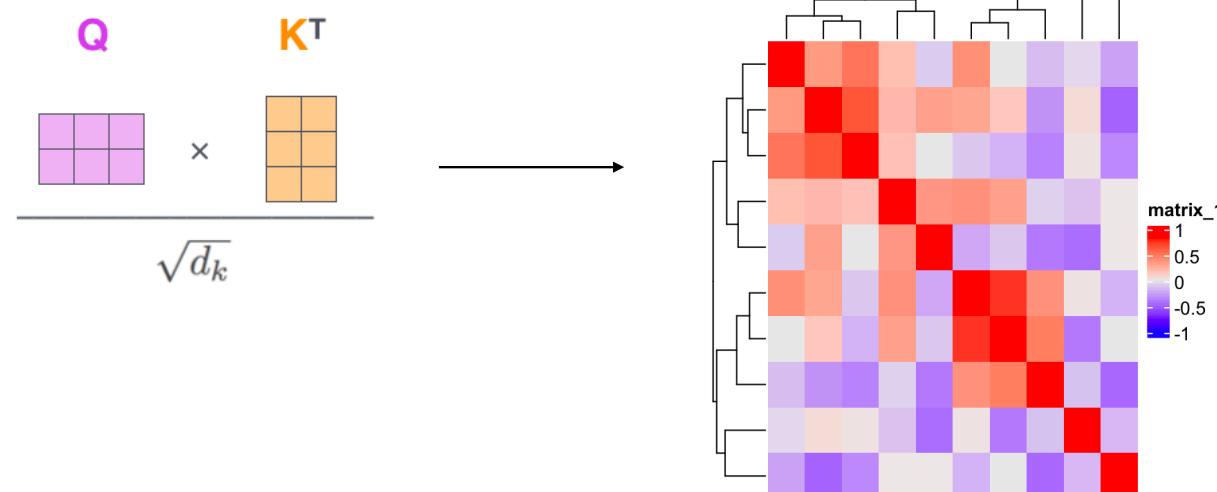
3) The result would be the Z matrix that captures information from all the attention heads. We can send this forward to the FFNN

$$= \begin{matrix} Z \\ \hline \end{matrix}$$

Are W_k and W_q identical? Better not to be identical!

$$q_i = W_q x_i, \quad k_i = W_k x_i, \quad v_i = W_v x_i$$

$$w_{ij} = q_i^T k_j / \sqrt{d}, \quad W_{ij} = \text{softmax}(w_{ij}), \quad y_i = \sum_j W_{ij} v_j.$$



It is in this spirit that a majority of American governments have passed new laws since 2009 making the registration or voting process more difficult.

<EOS> <pad> <pad> <pad> <pad> <pad> <pad> <pad>

– Transformers



Attention Is All You Need

Ashish Vaswani*

Google Brain

avaswani@google.com

Noam Shazeer*

Google Brain

noam@google.com

Niki Parmar*

Google Research

nikip@google.com

Jakob Uszkoreit*

Google Research

usz@google.com

Llion Jones*

Google Research

llion@google.com

Aidan N. Gomez* †

University of Toronto

aidan@cs.toronto.edu

Łukasz Kaiser*

Google Brain

lukaszkaiser@google.com

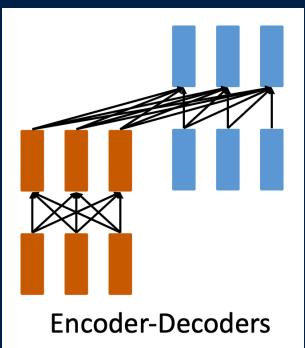
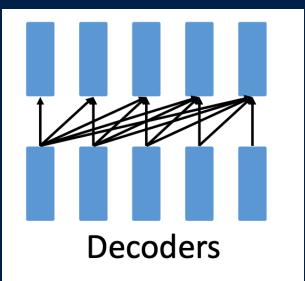
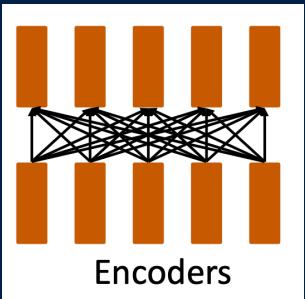
Illia Polosukhin* ‡

illia.polosukhin@gmail.com

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems* (pp. 5998-6008).

<https://arxiv.org/abs/1706.03762>

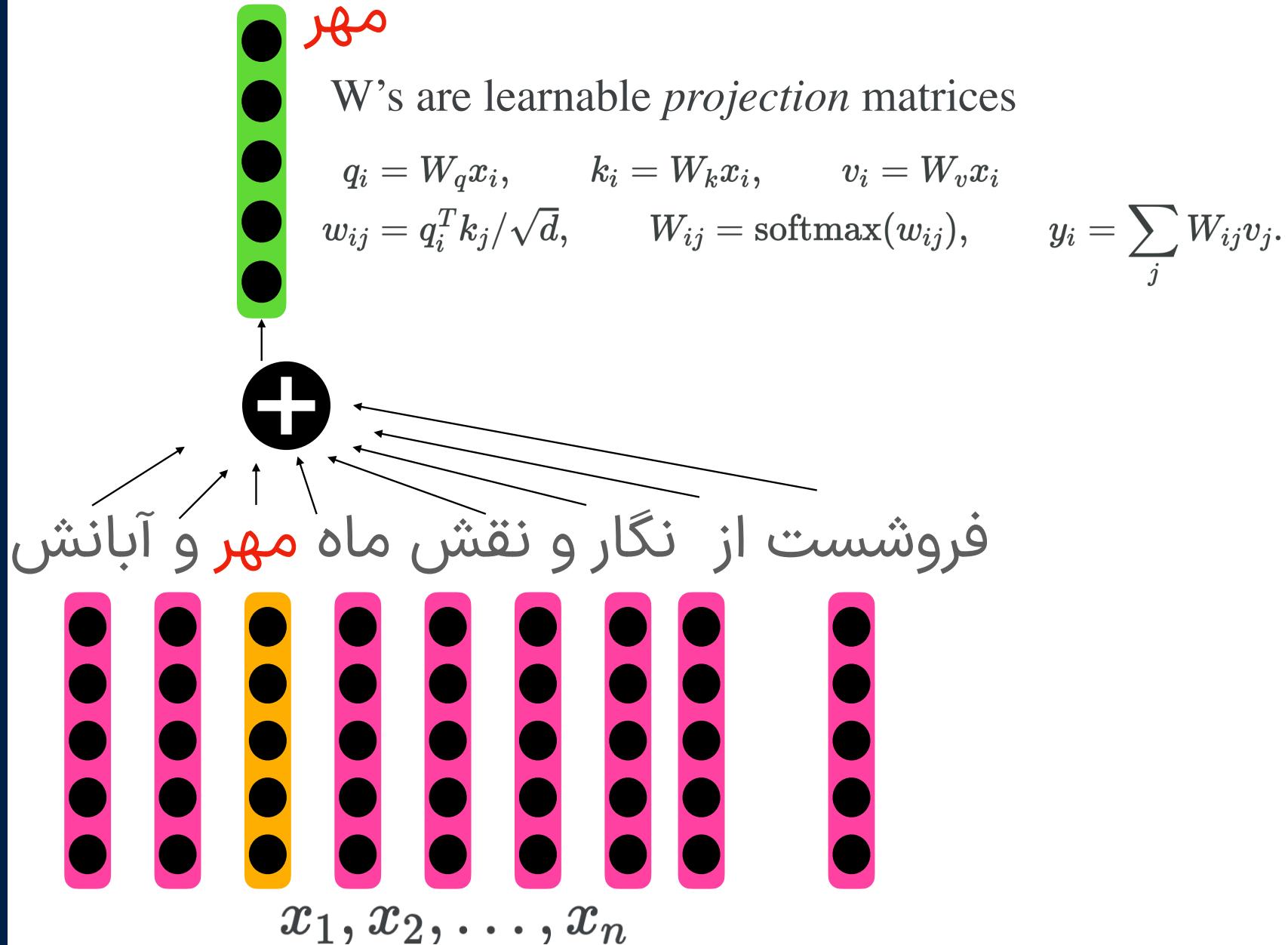
Transformer Architectures



- Encoder-only (e.g., BERT): bidirectional contextual embeddings
- Decoder-only (e.g., GPT-x): unidirectional contextual embeddings, generate one token at a time
- Encoder-decoder (e.g., T5): encode input, decode output

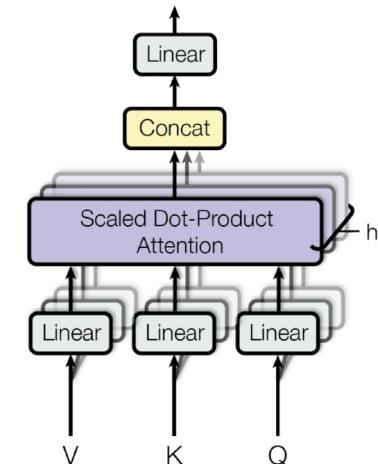
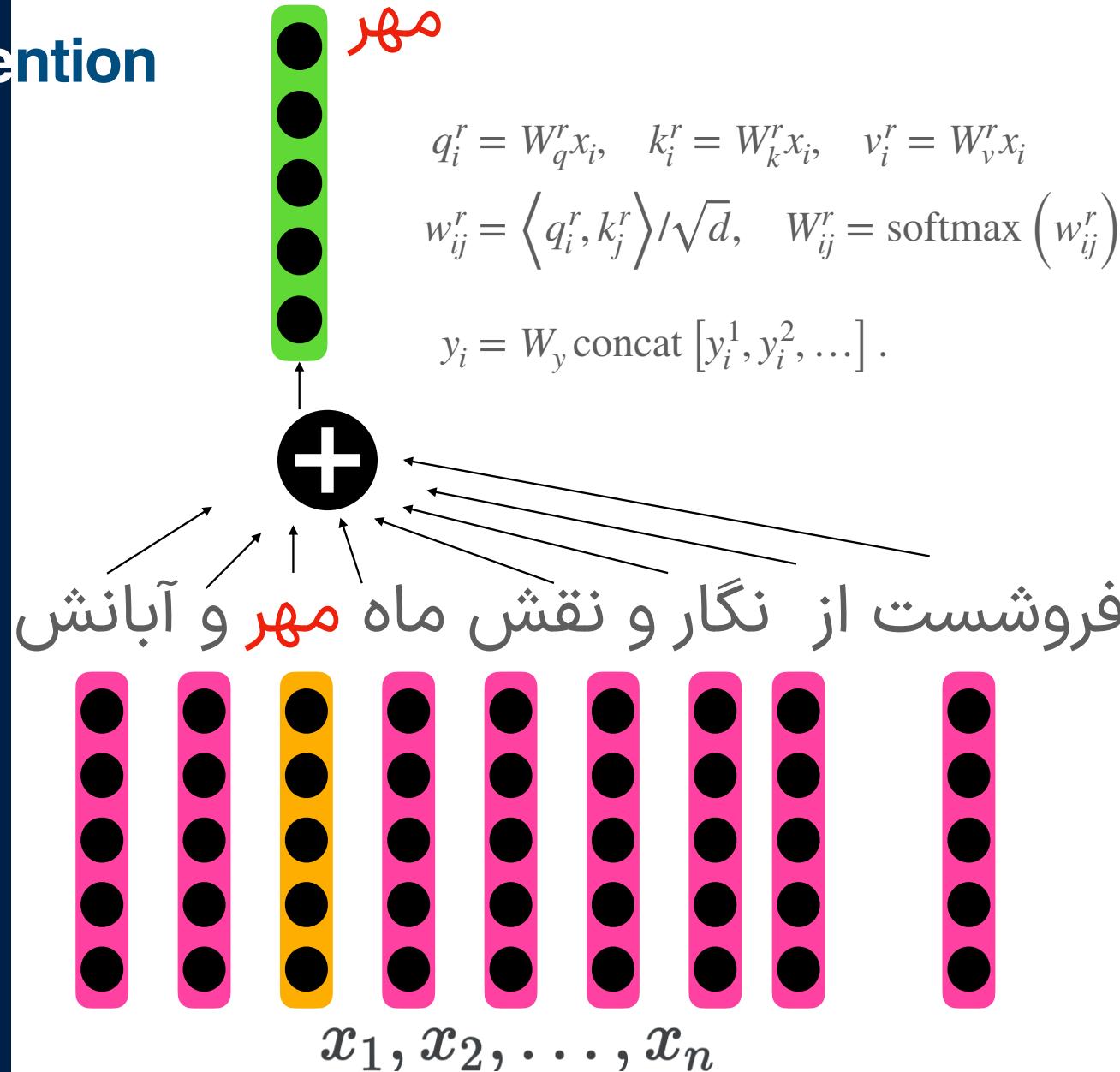
REVIEW

Attention

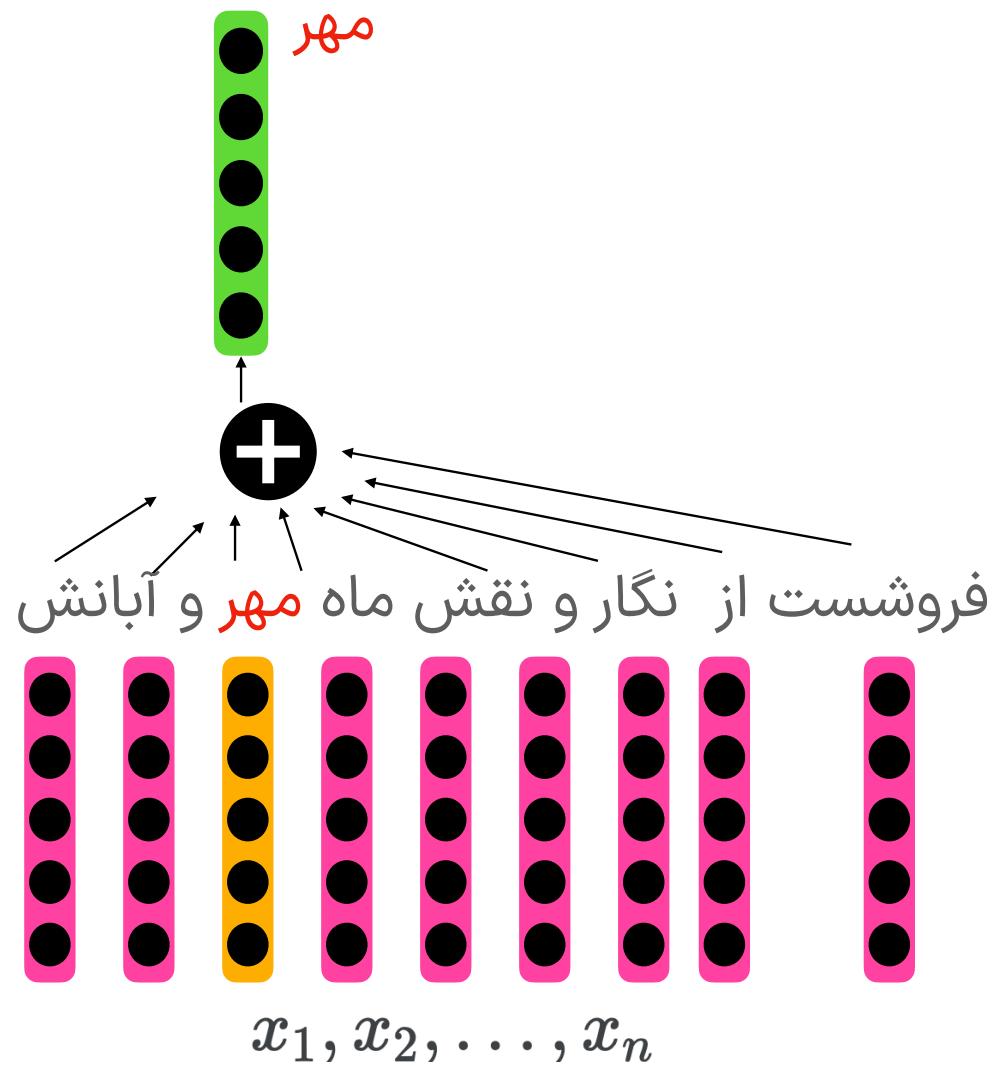


REVIEW

Multihead Attention



Order?

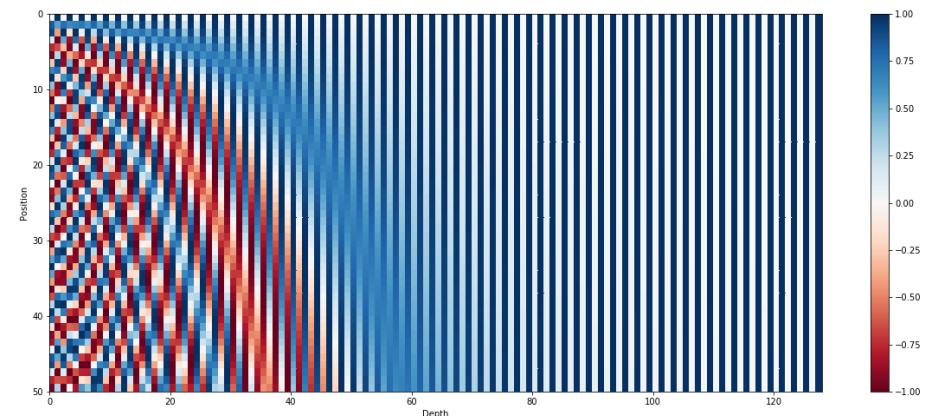
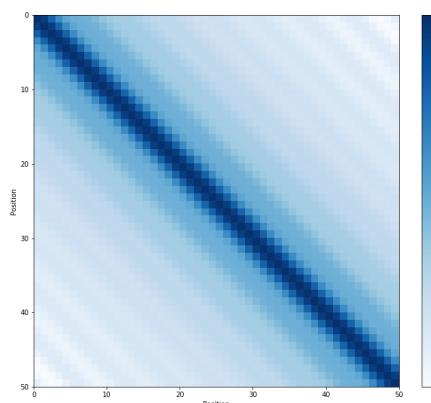


Sinusoidal Positional Embedding

"We chose this function because we hypothesized it would allow the model to easily learn to attend by relative positions, since for any fixed offset k , PE_{pos+k} can be represented as a linear function of PE_{pos} ."

$$\vec{p}_t^{(i)} = f(t)^{(i)} := \begin{cases} \sin(\omega_k \cdot t), & \text{if } i = 2k \\ \cos(\omega_k \cdot t), & \text{if } i = 2k + 1 \end{cases} \quad \omega_k = \frac{1}{10000^{2k/d}}$$

$$\vec{p}_t = \begin{bmatrix} \sin(\omega_1 \cdot t) \\ \cos(\omega_1 \cdot t) \\ \sin(\omega_2 \cdot t) \\ \cos(\omega_2 \cdot t) \\ \vdots \\ \sin(\omega_{d/2} \cdot t) \\ \cos(\omega_{d/2} \cdot t) \end{bmatrix}_{d \times 1}$$



Vaswani, Ashish, et al. "Attention is all you need." Advances in neural information processing systems 30 (2017).

بسم الله الرحمن الرحيم

پردازش زبانهای طبیعی

جلسه ۱۷

احسان الدین عسگری

اردیبهشت ۱۴۰۳

<http://language.ml/>

دانشکده مهندسی کامپیوتر

آزمایشگاه پردازش هوشمند متن و زبان و علوم انسانی محاسباتی

asgari@berkeley.edu



Positional Embedding

- * Assign a number to each time-step within the [0, 1]
 - * Time-step differences are not consistent in different sentences.
- * Assign a natural number to each time-step
 - * Long sentences
 - * Differences in the training and the inference

Positional Embedding

- ★ Unique encoding for each time-step.
- ★ Consistent distance metric between time-steps in varying sentence lengths.
- ★ Easily adapts to longer sentences with bounded values.
- ★ Deterministic output.

Positional

Embedding types?

ABSOLUTE VS. RELATIVE POSITION ENCODING

مهر ماه باران می بارد

معمولاً هر سال مهر ماه باران می بارد

Positional Embedding types?

ABSOLUTE VS. RELATIVE POSITION ENCODING

مهر ماه باران می بارد

معمولاً هر سال مهر ماه باران می بارد

The diagram illustrates the decomposition of a sequence into three components. On the left, the Persian sentence "مهر ماه باران می بارد" is shown. To its right is a dot followed by three matrices: an "Attention Matrix" with entries a_{ij} , an "Absolute Position Bias" with entries $p_{i,j}$, and a "Relative Position Bias" with entries $r_{i,j}$. The matrices are arranged horizontally, with the first two sharing the same column index and the third sharing the same row index.

$$\begin{matrix} \text{مهر} & \text{ماه} & \text{باران} \\ \cdot & \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} & \begin{bmatrix} p_{11} & p_{12} & p_{13} \\ p_{21} & p_{22} & p_{23} \\ p_{31} & p_{32} & p_{33} \end{bmatrix} & \begin{bmatrix} r_0 & r_1 & r_2 \\ r_{-1} & r_0 & r_1 \\ r_{-2} & r_{-1} & r_0 \end{bmatrix} \end{matrix}$$

Attention Matrix Absolute Position Bias Relative Position Bias

Absolute position embeddings are favorable for classification tasks and relative embeddings perform better for span prediction tasks.

Philipp Dufter, Martin Schmitt, and Hinrich Schütze. 2022. Position Information in Transformers: An Overview. *Computational Linguistics*, 48(3):733–763.

Adding Position Embeddings

Input Embedding $U \in \mathbb{R}^{\times d}$

Position Embedding $P \in \mathbb{R}^{\times d}$

$$\tilde{\mathbf{A}} = \sqrt{\frac{1}{d}}(\mathbf{U} + \mathbf{P})\mathbf{W}^{(q)}\mathbf{W}^{(k)\top}(\mathbf{U} + \mathbf{P})^\top$$

$$\tilde{\mathbf{M}} = \text{SoftMax}(\tilde{\mathbf{A}})(\mathbf{U} + \mathbf{P})\mathbf{W}^{(v)}$$

$$\tilde{\mathbf{O}} = \text{LayerNorm}_2(\tilde{\mathbf{M}} + \mathbf{U} + \mathbf{P})$$

$$\tilde{\mathbf{F}} = \text{ReLU}(\tilde{\mathbf{O}}\mathbf{W}^{(f_1)} + \mathbf{b}^{(f_1)})\mathbf{W}^{(f_2)} + \mathbf{b}^{(f_2)}$$

$$\tilde{\mathbf{Z}} = \text{LayerNorm}_1(\tilde{\mathbf{O}} + \tilde{\mathbf{F}})$$

Modifying Attention Matrix

Input Embedding $U \in \mathbb{R}^{d \times d}$

Position Embedding $P \in \mathbb{R}^{d \times d}$

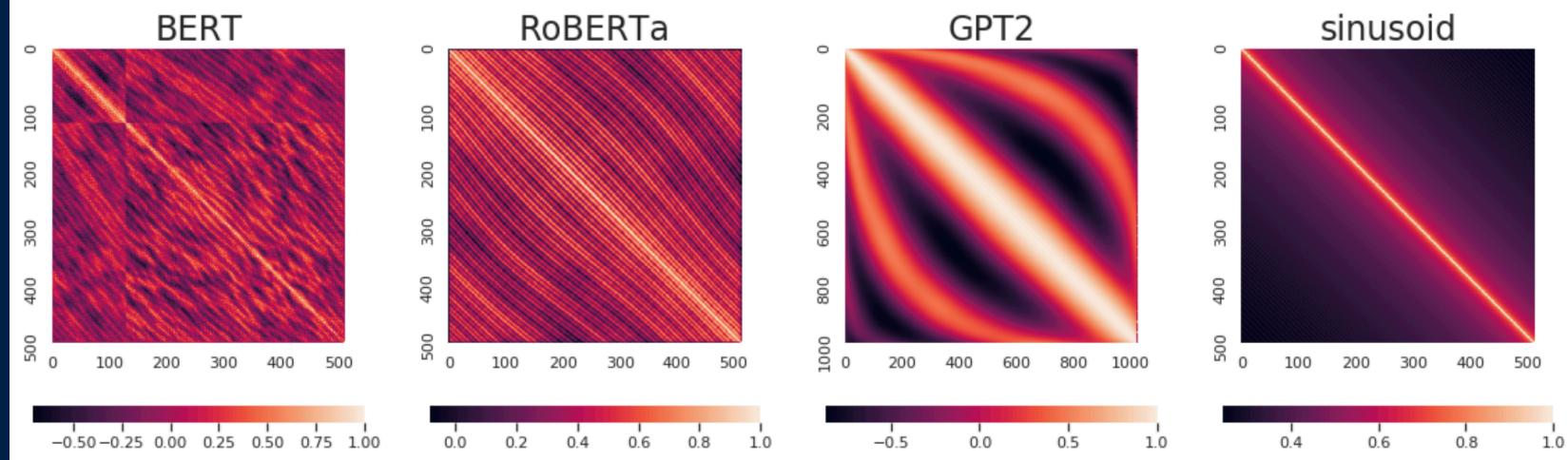
$$\hat{\mathbf{A}} \sim \underbrace{\mathbf{U}\mathbf{W}^{(q)}\mathbf{W}^{(k)\top}\mathbf{U}\top}_{\text{unit-unit } \sim \mathbf{A}} + \underbrace{\mathbf{P}\mathbf{W}^{(q)}\mathbf{W}^{(k)\top}\mathbf{U}\top}_{\text{unit-position}} + \underbrace{\mathbf{U}\mathbf{W}^{(q)}\mathbf{W}^{(k)\top}\mathbf{P}\top}_{\text{position-unit}} + \underbrace{\mathbf{P}\mathbf{W}^{(q)}\mathbf{W}^{(k)\top}\mathbf{P}\top}_{\text{position-position}}$$

Rotary Positional Embedding

$$\mathbf{T}^{(k)} \mathbf{E}_{t,:} = \mathbf{E}_{t+k,:}$$

$$\mathbf{T}^{(k)} = \begin{bmatrix} \Phi_1^{(k)} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \Phi_2^{(k)} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \dots & \Phi_{\frac{d_{\text{model}}}{2}}^{(k)} \end{bmatrix} \quad \Phi_m^{(k)} = \begin{bmatrix} \cos(\lambda_m k) & \sin(\lambda_m k) \\ -\sin(\lambda_m k) & \cos(\lambda_m k) \end{bmatrix}$$
$$\lambda_m = 10000^{\frac{-2m}{d_{\text{model}}}}$$

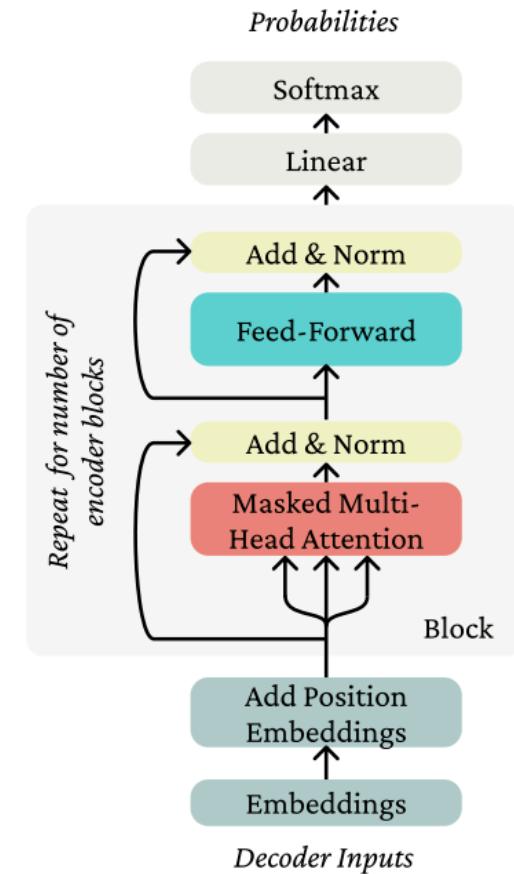
Positional Embedding types?



Yu-An Wang and Yun-Nung Chen. 2020. What Do Position Embeddings Learn? An Empirical Study of Pre-Trained Language Model Positional Encoding. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6840–6849, Online. Association for Computational Linguistics.

Transformer block

- Each block has two “sublayers”
 1. Multihead attention
 2. Feed-forward NNet (with ReLU)
- Residual: $x + \text{Sublayer}(x)$
- Layernorm changes input to have mean 0 and variance 1



Layer normalization

1. **Main Idea:** Batch normalization is advantageous for stability but presents challenges with sequences of varying lengths.
2. **Result:** It provides a more stable input for the next layer.
3. **Solution:** "layer normalization" functions similarly to batch normalization but doesn't normalize across the entire batch.

- Batch norm

$d\text{-dim}$ a_1, a_2, \dots, a_B $d\text{-dimensional vectors}$
for each sample in batch

$$\mu = \frac{1}{B} \sum_{i=1}^B a_i \quad \sigma = \sqrt{\frac{1}{B} \sum_{i=1}^B (a_i - \mu)^2}$$
$$\bar{a}_i = \frac{a_i - \mu}{\sigma}$$

- Layer norm

1-dim different *dimensions* of a

$$\mu = \frac{1}{d} \sum_{i=1}^d a_j \quad \sigma = \sqrt{\frac{1}{d} \sum_{i=1}^d (a_j - \mu)^2}$$
$$\bar{a} = \frac{a - \mu}{\sigma}$$

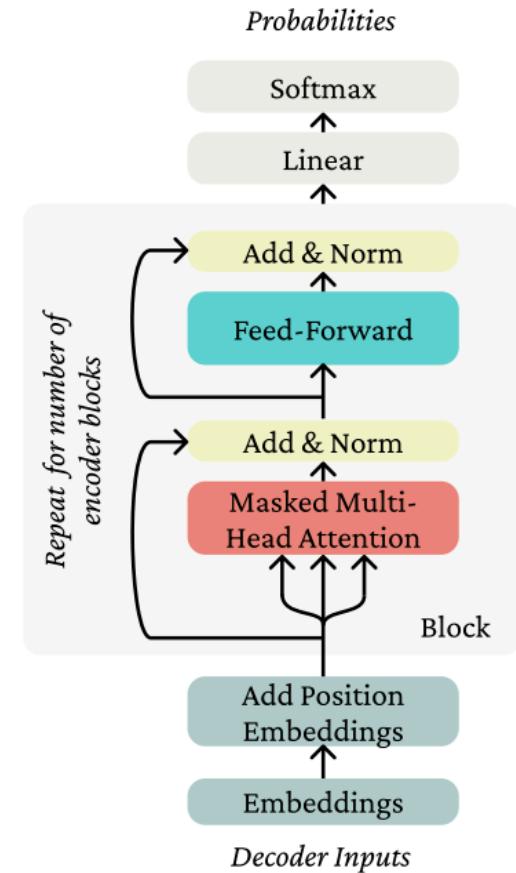
Why transformers?

Pros:

- + Much easier to parallelize
- + Much better long-range connections
- + In practice, can make it much deeper (more layers) than RNN

Cons:

- Attention computations are technically $O(n^2)$
- Somewhat more complex to implement (positional encodings, etc.)



- Encoder Language Model
- BERT LM Architecture



Masked Language Modeling (MLM)

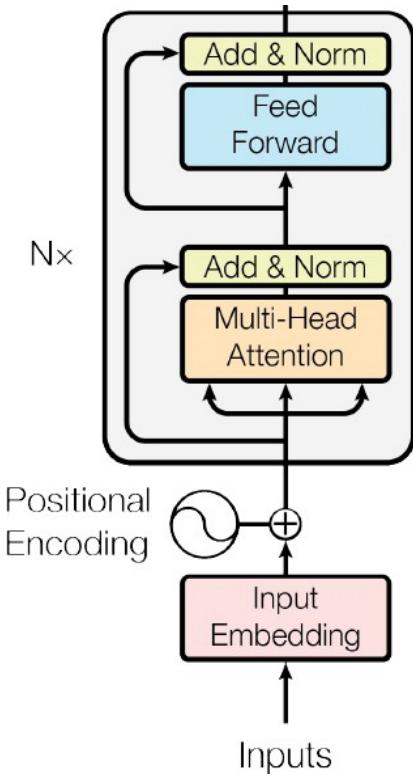
- Q: Why we can't do language modeling with bidirectional models?



- Solution: Mask out k% of the input words, and then predict the masked words

store
gallon
↑
↑
the man went to [MASK] to buy a [MASK] of milk

BERT pre-training: putting together



- BERT-base: 12 layers, 768 hidden size, 12 attention heads, 110M parameters
- BERT-large: 24 layers, 1024 hidden size, 16 attention heads, 340M parameters
- Training corpus: Wikipedia (2.5B) + BooksCorpus (0.8B)
- Max sequence size: 512 word pieces (roughly 256 and 256 for two non-contiguous sequences)
- Trained for 1M steps, batch size 128k

Sentence-level tasks

- Sentence pair classification tasks:

MNLI

Premise: A soccer game with multiple males playing.

Hypothesis: Some men are playing a sport.

{[entailment](#), contradiction, neutral}

Q1: Where can I learn to invest in stocks?

QQP

Q2: How can I learn more about stocks?

{[duplicate](#), not duplicate}

- Single sentence classification tasks:

SST2

rich veins of funny stuff in this movie

{[positive](#), negative}

Token-level tasks

- Extractive question answering e.g., SQuAD (Rajpurkar et al., 2016)

SQuAD

Question: The New York Giants and the New York Jets play at which stadium in NYC ?

Context: The city is represented in the National Football League by the New York Giants and the New York Jets , although both teams play their home games at MetLife Stadium in nearby East Rutherford , New Jersey , which hosted Super Bowl XLVIII in 2014 .

(Training example 29,883)

MetLife Stadium

CoNLL 2003 NER

John Smith lives in New York

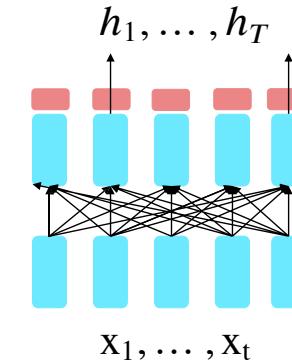
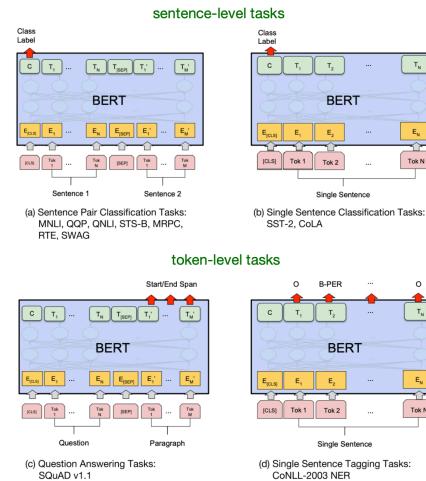
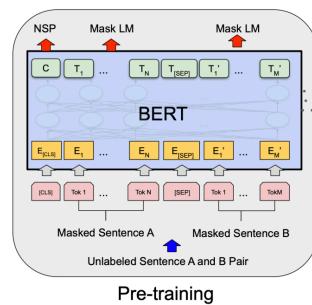
B-PER I-PER O O B-LOC I-LOC

Encoder LM

- BERT
- Variations

Encoder Language Model

$$P(x) = \prod_{i=1}^n P(x_i | x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n)$$



$$h_1, \dots, h_t = \text{Decoder}(x_1, \dots, x_t)$$

$$x_{\text{mask}} \sim Ah_{\text{masked}} + b$$



BERT: key contributions

- It is a **fine-tuning approach** based on a deep **Transformer encoder**
- The key: learn representations based on **bidirectional context**

Why? Because both left and right contexts are important to understand the meaning of words.

Example #1: we went to the river bank.

Example #2: I need to go to bank to make a deposit.

- **Pre-training objectives:** masked language modeling + next sentence prediction
- State-of-the-art performance on a large set of **sentence-level** and **token-level** tasks

MLM:masking rate and strategy

- Q: What is the value of k ?
 - They always use $k = 15\%$.
 - Too little masking: computationally expensive (we need to increase # of epochs)
 - Too much masking: not enough context
 - See (Wettig et al., 2022) for more discussion of masking rates:
 - Masking 40% outperforms 15% for BERT-large size models on GLUE and SQuAD
 - High masking rate of 80% can still preserve 95% fine-tuning performance
- Q: How are masked tokens selected?
 - 15% tokens are uniformly sampled
 - Is it optimal? See span masking (Joshi et al., 2020) and PMI masking (Levine et al., 2021)

Example: He [MASK] from Kuala [MASK] , Malaysia.

Next Sentence Prediction (NSP)

- Motivation: many NLP downstream tasks require understanding the relationship between two sentences (natural language inference, paraphrase detection, QA)
- NSP is designed to reduce the gap between pre-training and fine-tuning

[CLS]: a special token
always at the beginning

[SEP]: a special token used
to separate two segments

Input = [CLS] the man went to [MASK] store [SEP]

he bought a gallon [MASK] milk [SEP]

Label = IsNext

Input = [CLS] the man [MASK] to the store [SEP]

penguin [MASK] are flight ##less birds [SEP]

Label = NotNext

They sample two contiguous segments for 50% of the time and another random segment from the corpus for 50% of the time

بسم الله الرحمن الرحيم

پردازش زبانهای طبیعی

جلسه ۱۸

احسان الدین عسگری

اردیبهشت ۱۴۰۳

<http://language.ml/>

دانشکده مهندسی کامپیوتر

آزمایشگاه پردازش هوشمند متن و زبان و علوم انسانی محاسباتی

asgari@berkeley.edu



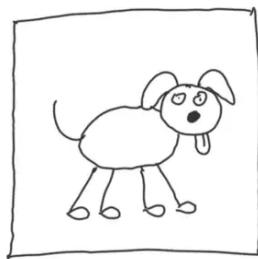
BERT Training

Dataset. Let \mathcal{D} be a set of examples $(x_{1:L}, c)$ constructed as follows:

- Let A be a sentence from the corpus.
- With probability 0.5, let B be the next sentence.
- With probability 0.5, let B be a random sentence from the corpus.
- Let $x_{1:L} = [[\text{CLS}], A, [\text{SEP}], B]$.
- Let c denote whether B is the next sentence or not.

Objective. Then the BERT objective is:

$$\mathcal{O}(\theta) = \sum_{(x_{1:L}, c) \in \mathcal{D}} \underbrace{\mathbb{E}_{I, \tilde{x}_{1:L} \sim A(\cdot | x_{1:L}, I)} \left[\sum_{i \in I} -\log p_\theta(\tilde{x}_i | x_{1:L}) \right]}_{\text{masked language modeling}} + \underbrace{-\log p(c | \phi(x_{1:L})_1)}_{\text{next sentence prediction}}$$



MODEL

0.5 → DOG PROBABILITY
0.3 → CAT PROBABILITY
0.2 → PANDA PROBABILITY

TARGET

1
0
0

Loss for class X = $- \underbrace{p(x)}_{\text{probability of class } X \text{ in TARGET}} \cdot \log \underbrace{q(x)}_{\text{probability of class } X \text{ in PREDICTION}}$

Regression

$$MSE = \frac{1}{n} \sum (y - \hat{y})^2$$

Multi-class

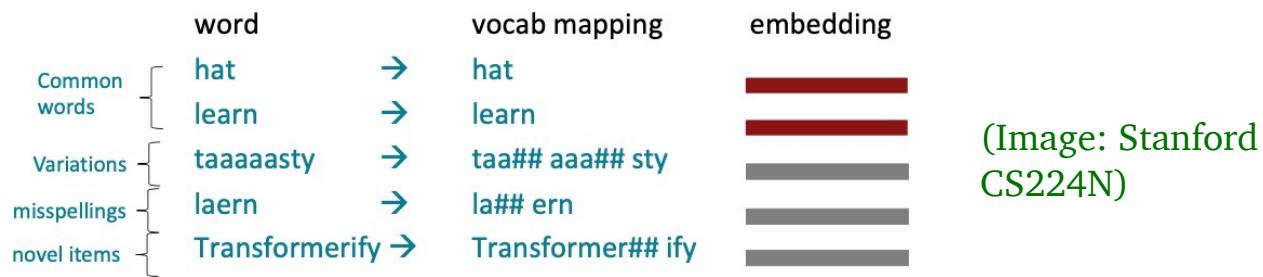
$$L_{\text{cross-entropy}}(\hat{\mathbf{y}}, \mathbf{y}) = - \sum_i y_i \log(\hat{y}_i)$$

Binary / Multi-label

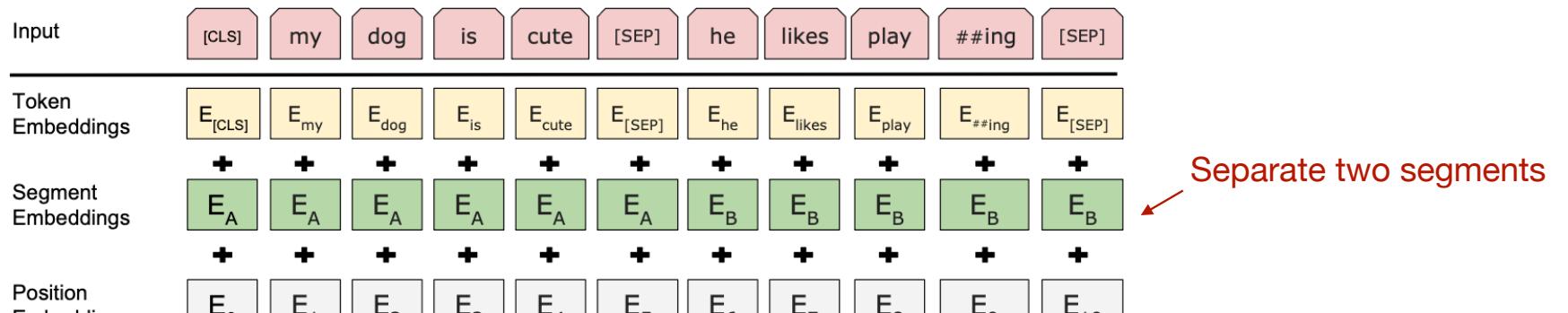
$$\text{Loss} = - \frac{1}{\text{output size}} \sum_{i=1}^{\text{output size}} y_i \cdot \log \hat{y}_i + (1 - y_i) \cdot \log (1 - \hat{y}_i)$$

BERT pre-training: putting together

- Vocabulary size: 30,000 wordpieces (common sub-word units) (Wu et al., 2016)



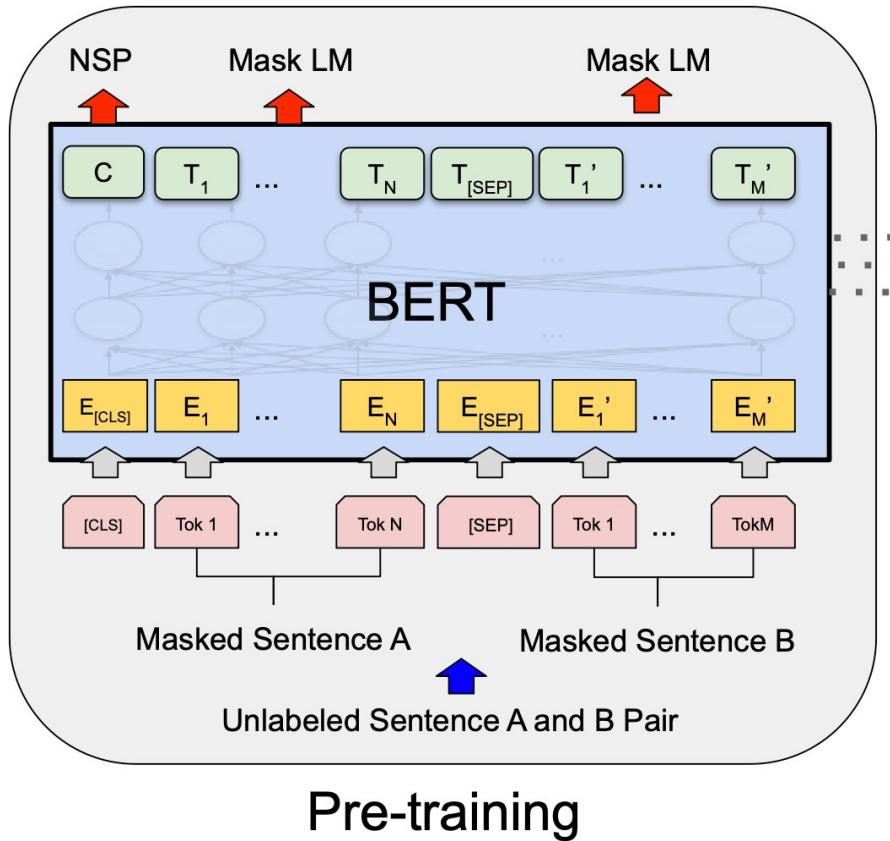
- Input embeddings:



- Just two possible "segment embeddings": E_A and E_B .

- Positional embeddings are learned vectors for every possible position between 0 and 512-1.

BERT pre-training: putting together

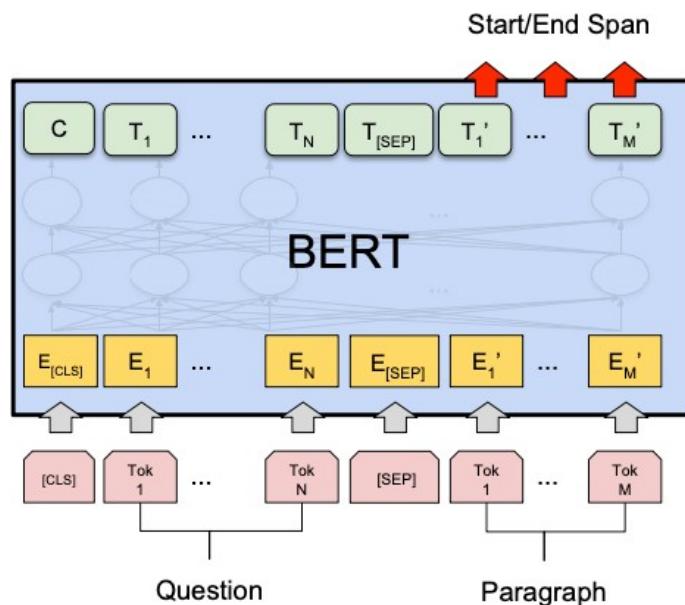


- MLM and NSP are trained together
- [CLS] is pre-trained for NSP
- Other token representations are trained for MLM

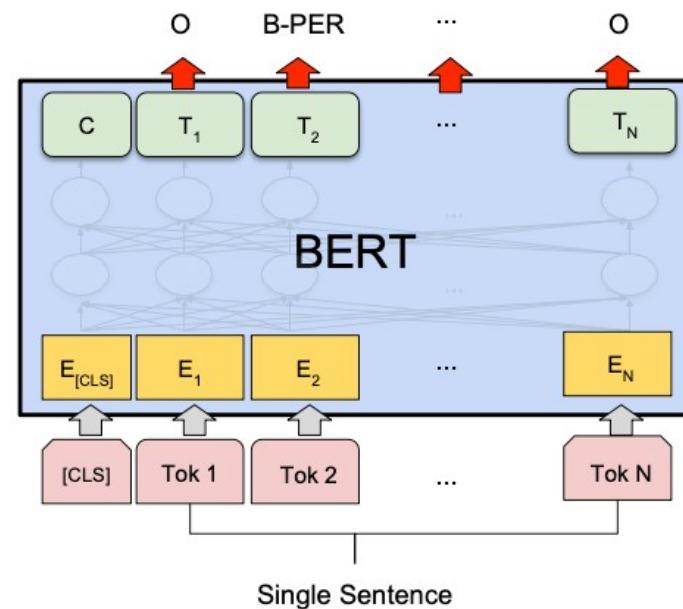
Fine-tuning BERT

“Pretrain once, finetune many times.”

token-level tasks



(c) Question Answering Tasks:
SQuAD v1.1

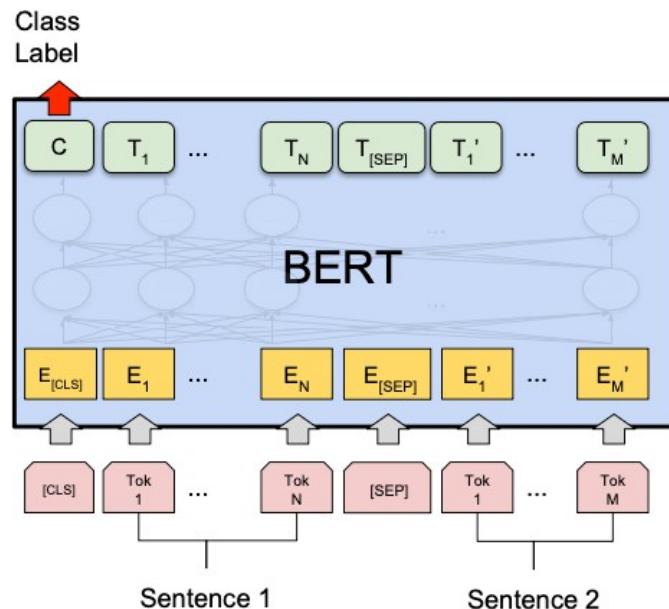


(d) Single Sentence Tagging Tasks:
CoNLL-2003 NER

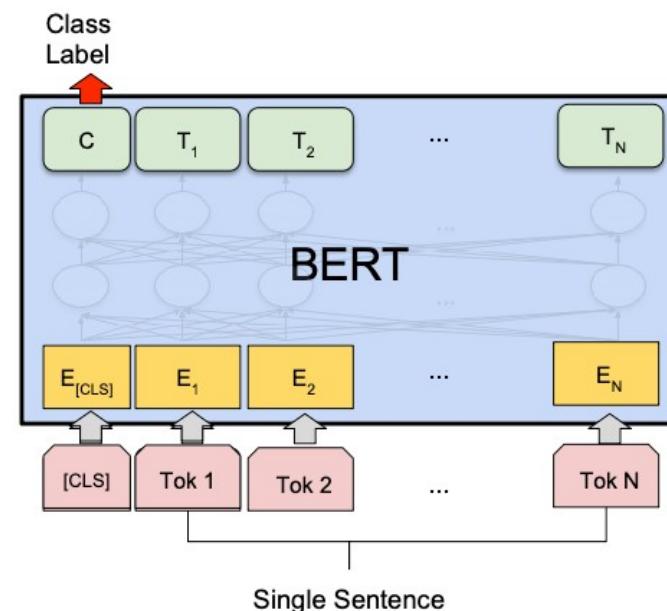
Fine-tuning BERT

“Pretrain once, finetune many times.”

sentence-level tasks

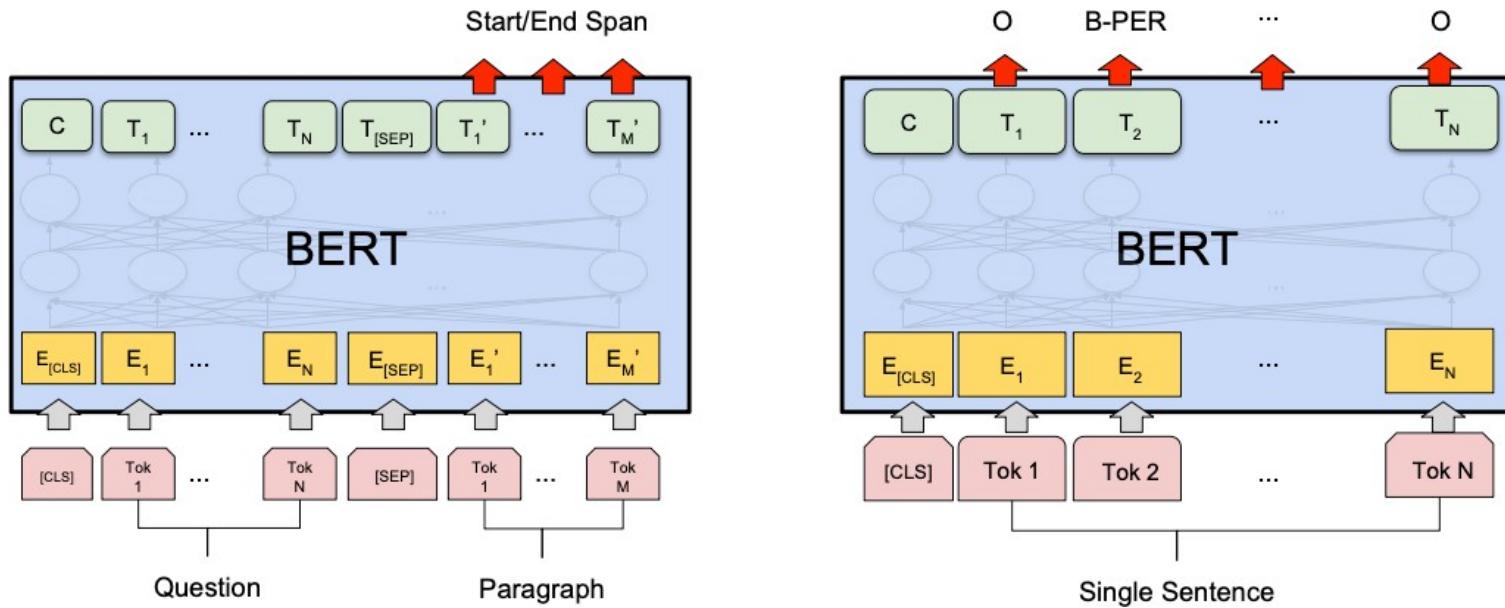


(a) Sentence Pair Classification Tasks:
MNLI, QQP, QNLI, STS-B, MRPC,
RTE, SWAG



(b) Single Sentence Classification Tasks:
SST-2, CoLA

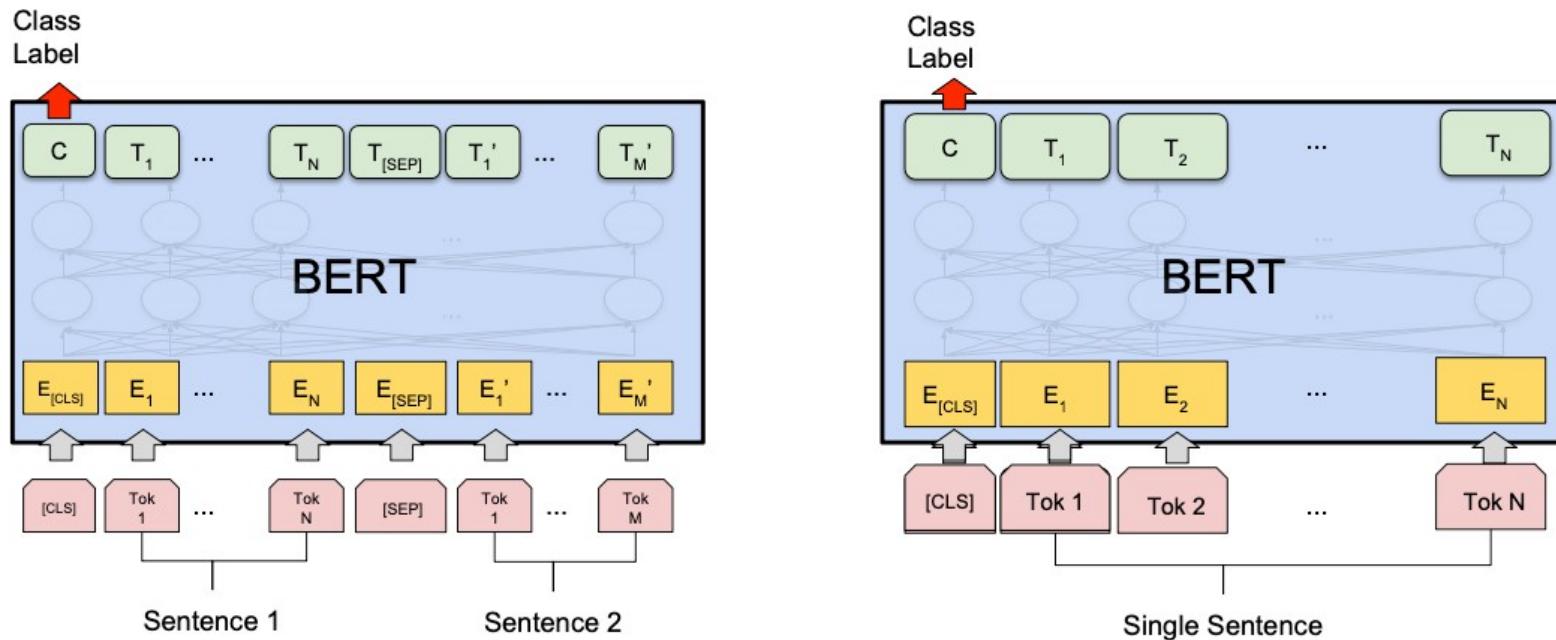
Fine-tuning BERT



- For token-level prediction tasks, add linear classifier on top of hidden representations

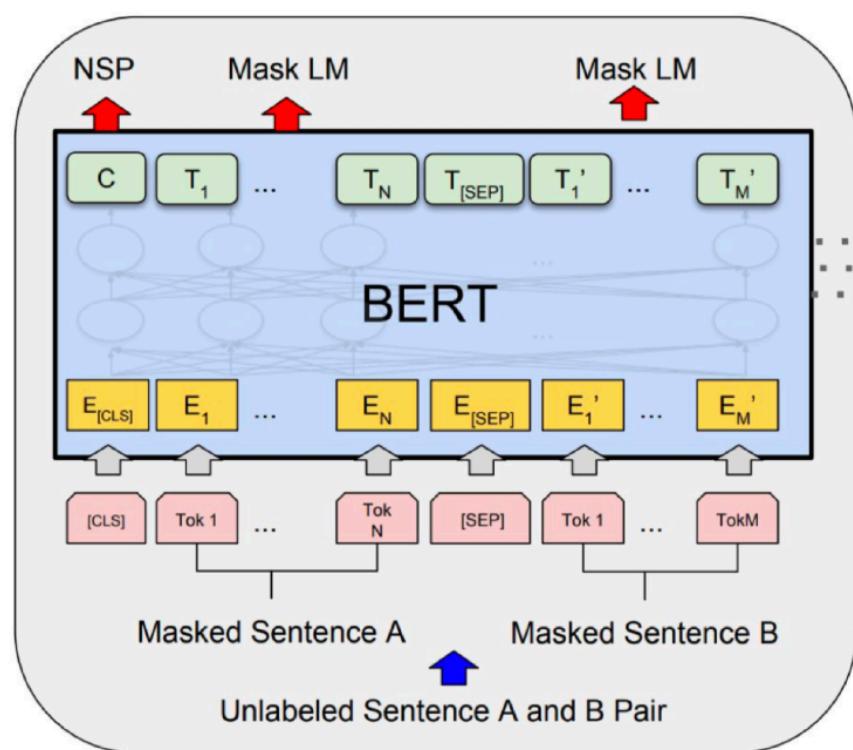
Q: How many new parameters?

Fine-tuning BERT

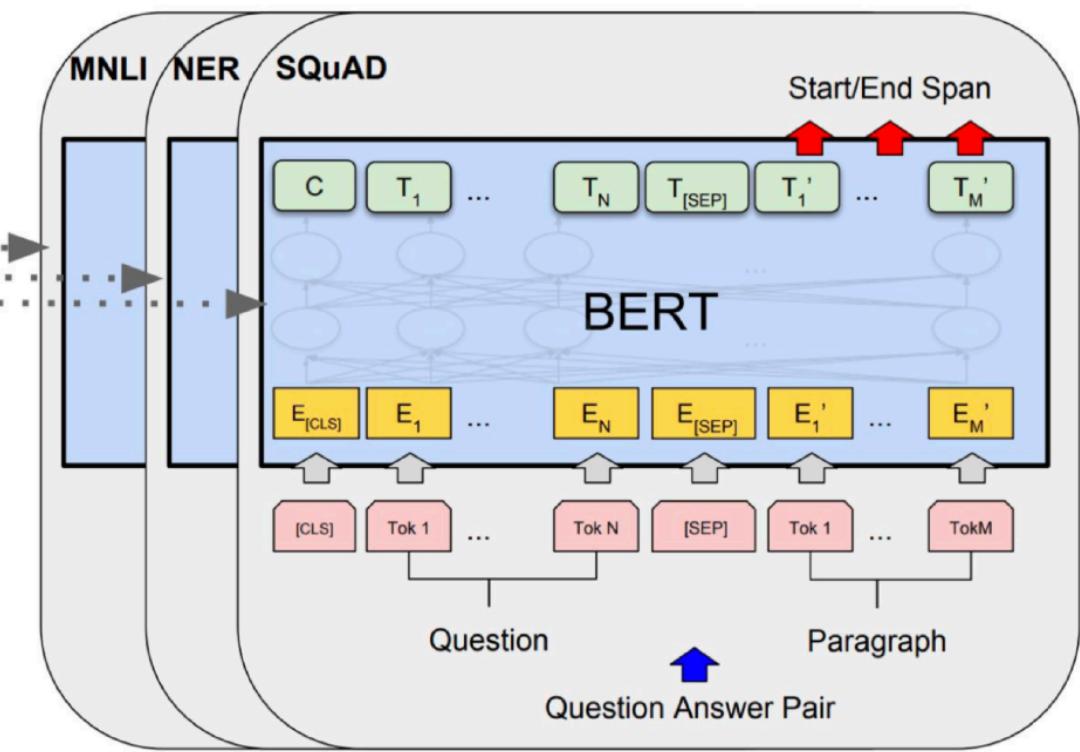


- For sentence pair tasks, use $[SEP]$ to separate the two segments with segment embeddings
- Add a linear classifier on top of $[CLS]$ representation

Finetuning Paradigm in NLP



Pre-training



Fine-Tuning

Encoder LM

- **BERT**
- **Variations**

BERT Extensions

- Models that handle long contexts ($\gg 512$ tokens)
 - Longformer, Big Bird, ...
- Multilingual BERT
 - Trained single model on 104 languages from Wikipedia. Shared 110k WordPiece vocabulary
- BERT extended to different domains
 - SciBERT, BioBERT, FinBERT, ClinicalBERT, ...
- Making BERT smaller to use
 - DistillBERT, TinyBERT, ...

Encoder LM

- **BERT**
- **Variations**

BERT Extensions

- RoBERTa (Liu et al., 2019)
 - Trained on 10x data & longer, no NSP
 - Much stronger performance than BERT (e.g., 94.6 vs 90.9 on SQuAD)
 - Still one of the most popular models to date
- ALBERT (Lan et al., 2020)
 - Increasing model sizes by sharing model parameters across layers
 - Less storage, much stronger performance but runs slower..

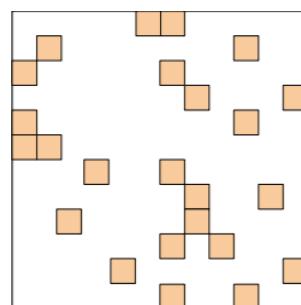
Encoder LM

- BERT
- Variations

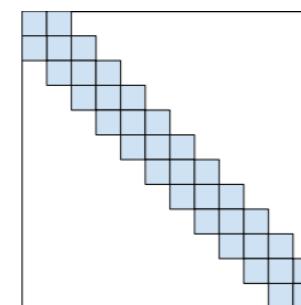
Reducing Attention Cost

- BigBird [\[Zaheer et al., 2021\]](#)

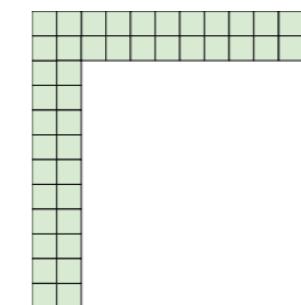
Key idea: replace all-pairs interactions with a family of other interactions, like local windows, looking at everything, and random interactions.



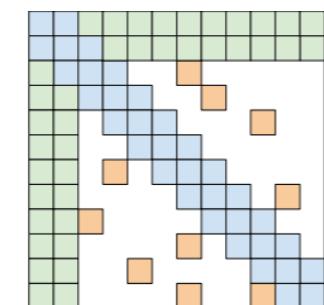
(a) Random attention



(b) Window attention



(c) Global Attention



(d) BIGBIRD

- Decoder Language Model
 - GPT LM Architecture



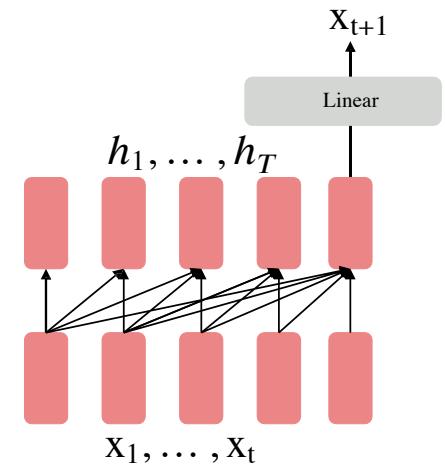
Decoder

- GPT-models

Decoder Language Model

Autoregressive (AR) models use decoder stacks in generation, aiming to maximize log-likelihood via forward autoregressive factorization:

$$\max_{\theta} \log p_{\theta}(x_1, \dots, x_T) \approx \sum_{t=1}^T \log p_{\theta}(x_t | x_1, \dots, x_{t-1})$$



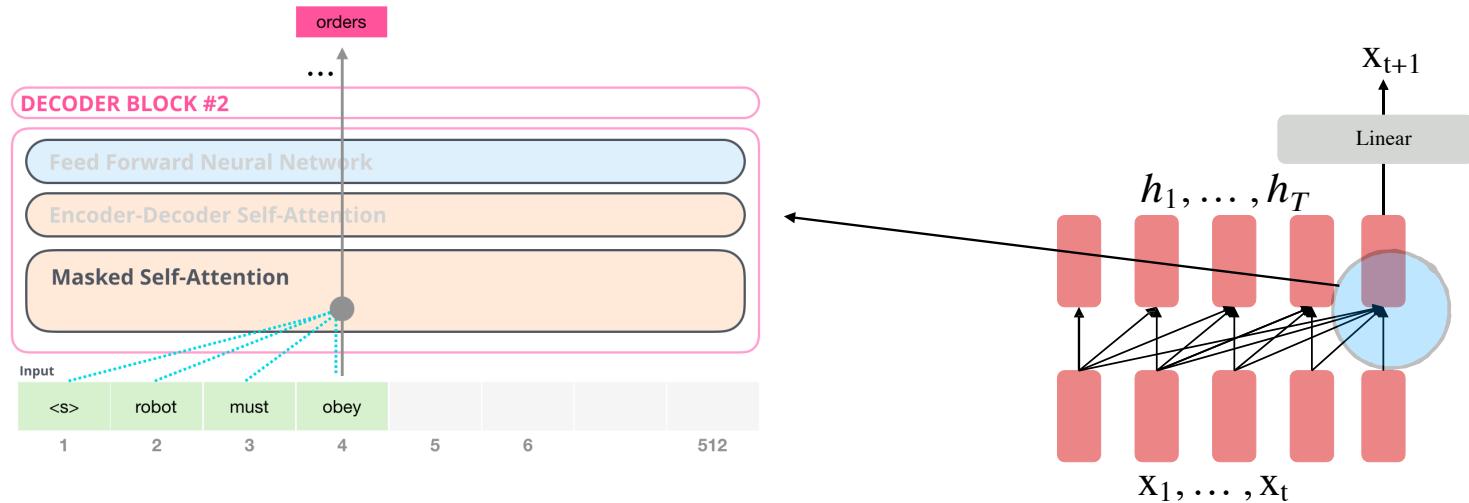
$$h_1, \dots, h_t = \text{Decoder}(x_1, \dots, x_t)$$
$$x_{t+1} \sim Ah_t + b$$

Decoder

- GPT-models

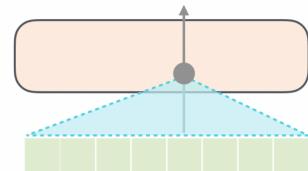
Decoder Language Model

$$\max_{\theta} \log p_{\theta}(x_1, \dots, x_T) \approx \sum_{t=1}^T \log p_{\theta}(x_t | x_1, \dots, x_{t-1})$$

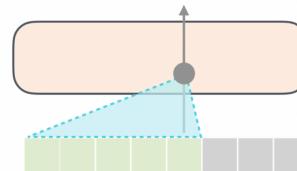


BERT vs. GPT

Self-Attention



Masked Self-Attention



$$h_1, \dots, h_t = \text{Decoder}(x_1, \dots, x_t)$$
$$x_{t+1} \sim Ah_t + b$$

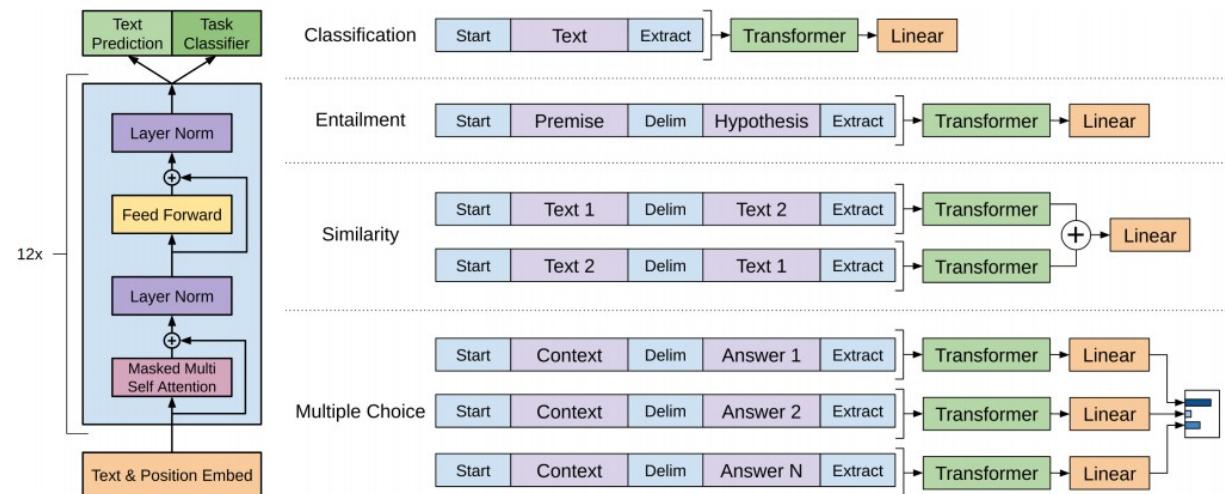
<http://jalammar.github.io/illustrated-gpt2/>

Decoder

- GPT-models

Generative Pre-Trained Transformer (GPT)

- Transformer decoder with 12 layers.
- Byte-pair encoding with 40,000 merges
- Trained on BooksCorpus: over 7000 unique books.
 - Contains long spans of contiguous text, for learning long-distance dependencies.

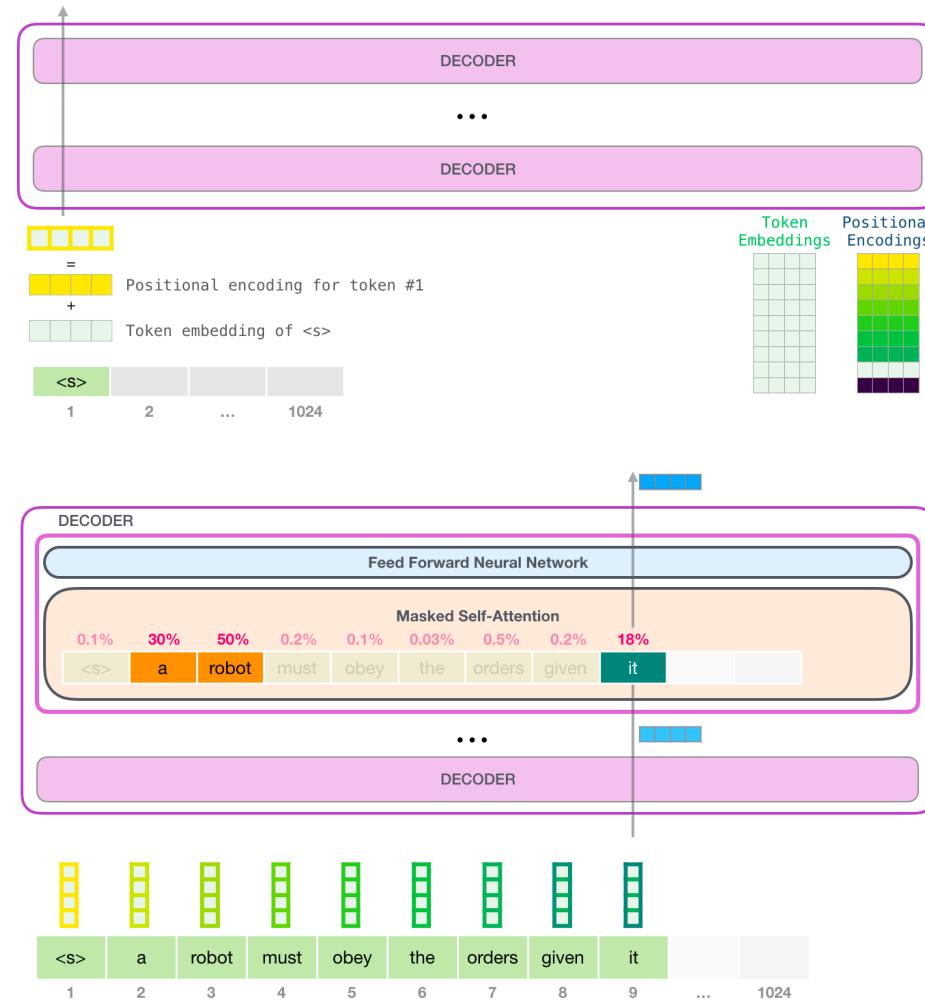


[Radford et al., 2018](https://www.cs.ubc.ca/~amuhamed/LING530/papers/radford2018improving.pdf) <https://www.cs.ubc.ca/~amuhamed/LING530/papers/radford2018improving.pdf>

Decoder

- GPT-models

Generative Pre-Trained Transformer (GPT)



Decoder

- GPT-models

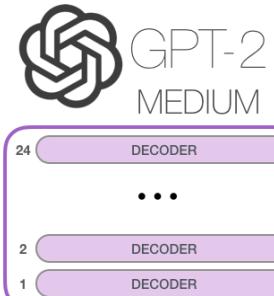
GPT released June 2018

GPT-2 released Nov. 2019 with 1.5B parameters

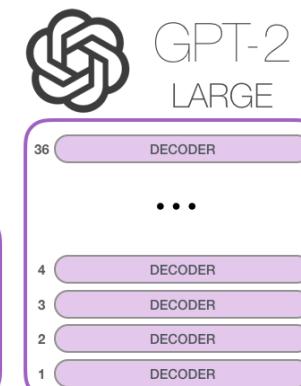
GPT-3: 175B parameters trained on 45TB texts



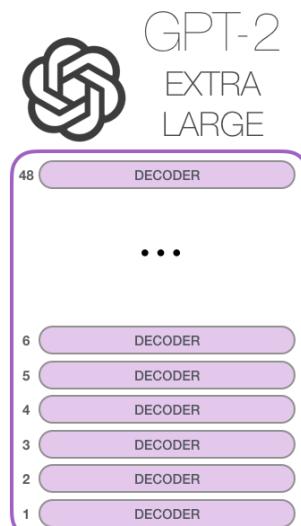
Model Dimensionality: 768



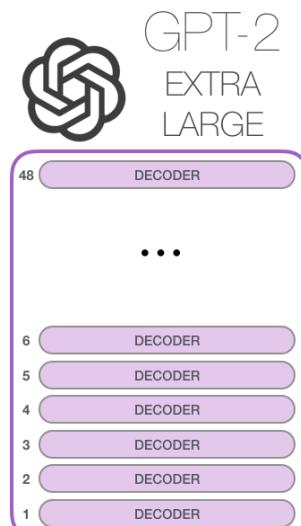
Model Dimensionality: 1024



Model Dimensionality: 1280



Model Dimensionality: 1600



Model Dimensionality: 1600

Decoder

- GPT-models

	Model	Data
GPT-2 (Radford et al. 2019)	Context size: 1024 tokens 117M-1.5B parameters	WebText (45 million outbound links from Reddit with 3+ karma); 8 million documents (40GB)
GPT-3 (Brown et al. 2020)	Context size: 2048 tokens 125M-175B parameters	Common crawl + WebText + “two internet-based books corpora” + Wikipedia (400B tokens, 570GB)

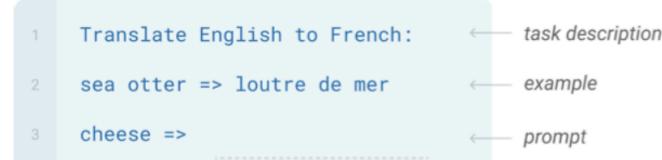
Zero-shot

The model predicts the answer given only a natural language description of the task. No gradient updates are performed.



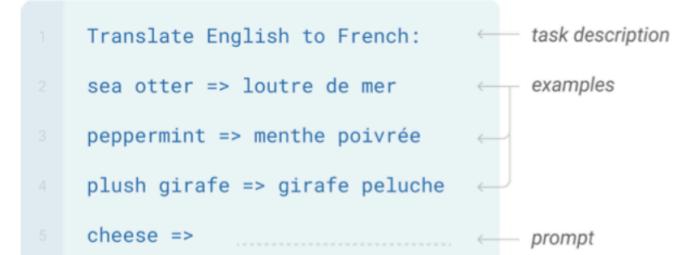
One-shot

In addition to the task description, the model sees a single example of the task. No gradient updates are performed.



Few-shot

In addition to the task description, the model sees a few examples of the task. No gradient updates are performed.



Brown et al. (2020, “Language Models are Few-Shot Learners”
<https://arxiv.org/pdf/2005.14165.pdf>

- Enc-Dec Language Model
- Attention is all you need, T5, BART

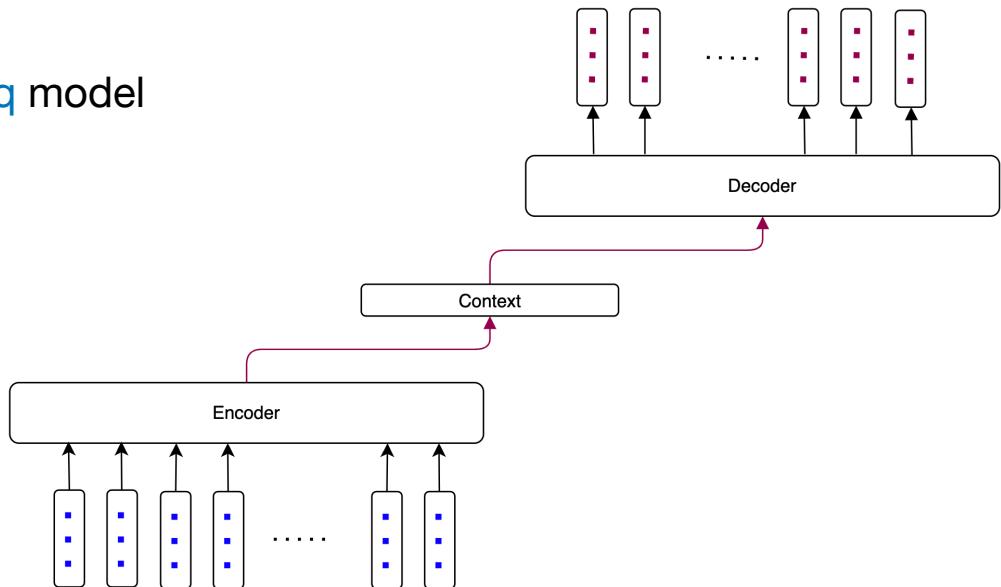


Encoder-Decoder

- Earlier Models
- Model T5
- Model BART
- Beam Search

Basic Idea of encoder-decoder

- The **encoder** encodes the input into a **context vector**
- The **decoder** produces task-specific **output** given the context
 - * **Output:** contextually relevant, variable-length
- Also known as **seq-to-seq** model

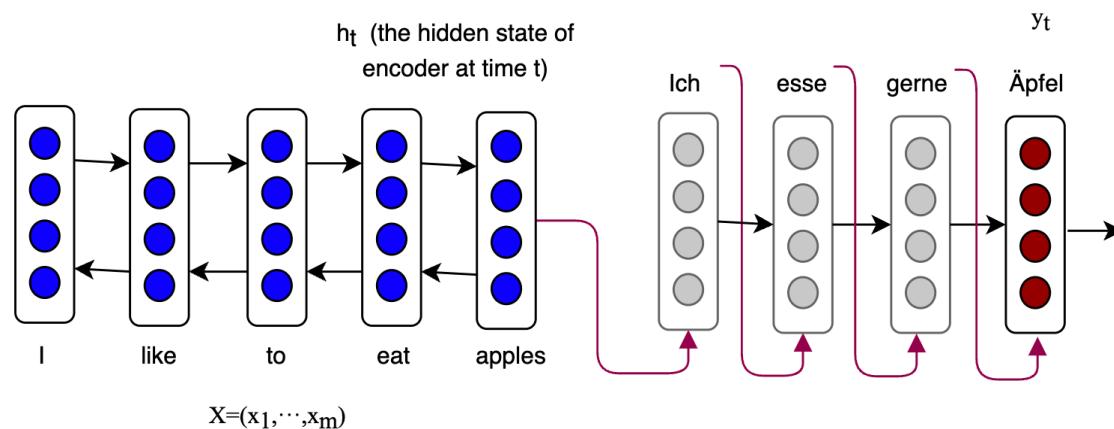


Encoder-Decoder

- **Earlier Models**
- Model T5
- Model BART
- Beam Search

Earliest works

- Machine Translation using RNNs



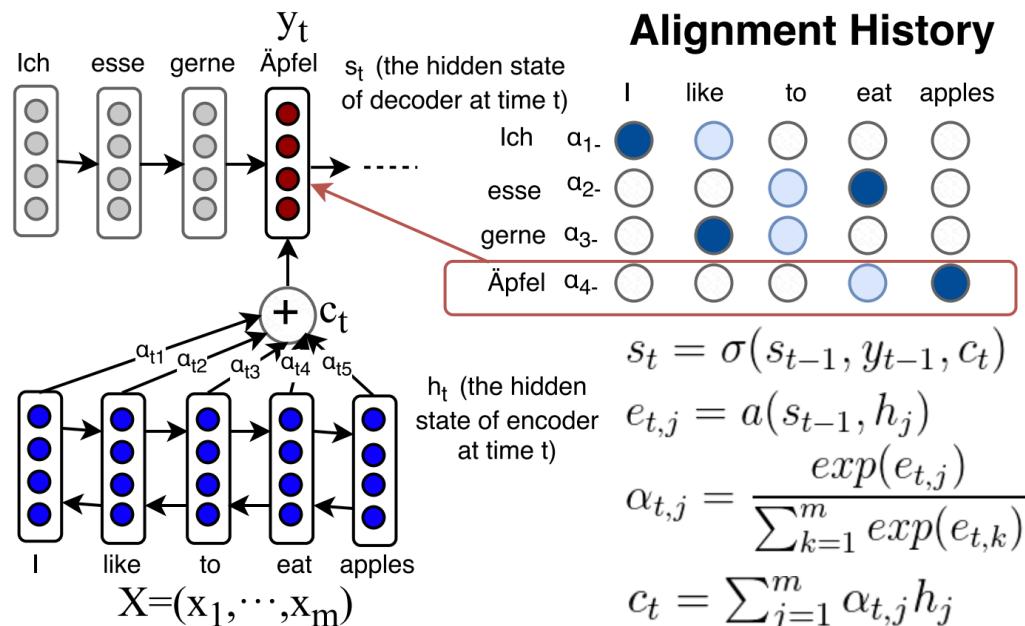
Ilya Sutskever, Oriol Vinyals, Quoc V. Le,
Sequence to Sequence Learning with Neural Networks. NIPS 2014: 3104-3112

Encoder-Decoder

- **Earlier Models**
- Model T5
- Model BART
- Beam Search

Earliest works

- Machine Translation using RNNs with **attention** mechanism



Dzmitry Bahdanau, Kyunghyun Cho, Yoshua Bengio,
Neural Machine Translation by Jointly Learning to Align and Translate. ICLR 2015

Encoder-Decoder

- **Earlier Models**
- **Model T5**
- **Model BART**
- **Beam Search**

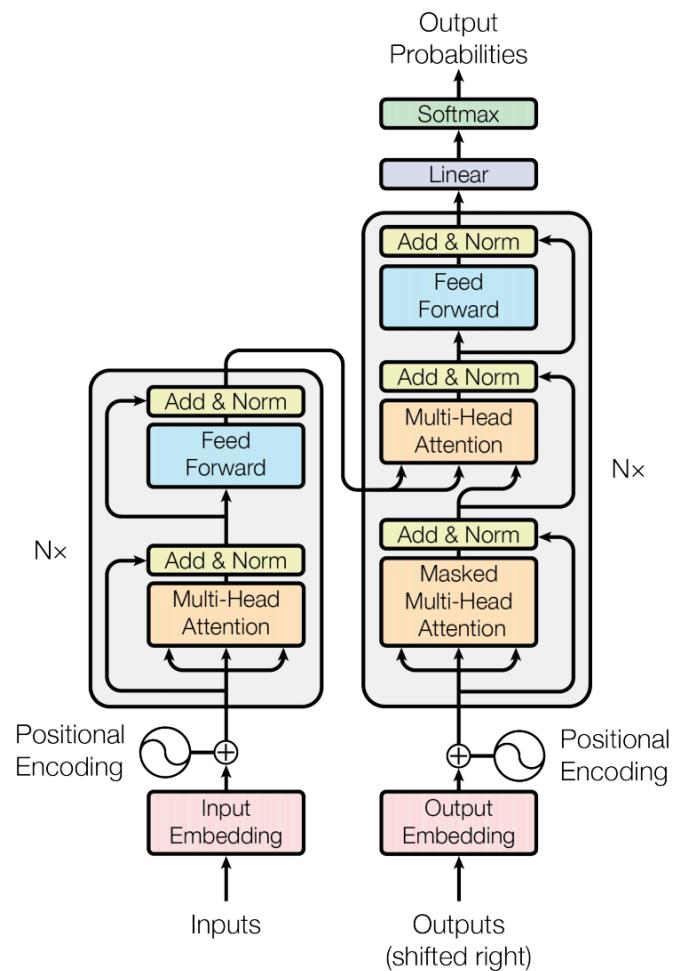
Earliest works

- Introducing transformers
 - Multihead attention
 - For machine translation

$$\Pr(y_1, \dots, y_n | \mathbf{x}) = \prod_i^n \Pr(y_i | y_{i-1}, \dots, y_1, \mathbf{x})$$

- Target seq $\mathbf{y} = (y_1, \dots, y_{T_y})$
- Source seq $\mathbf{x} = (x_1, \dots, x_{T_x})$

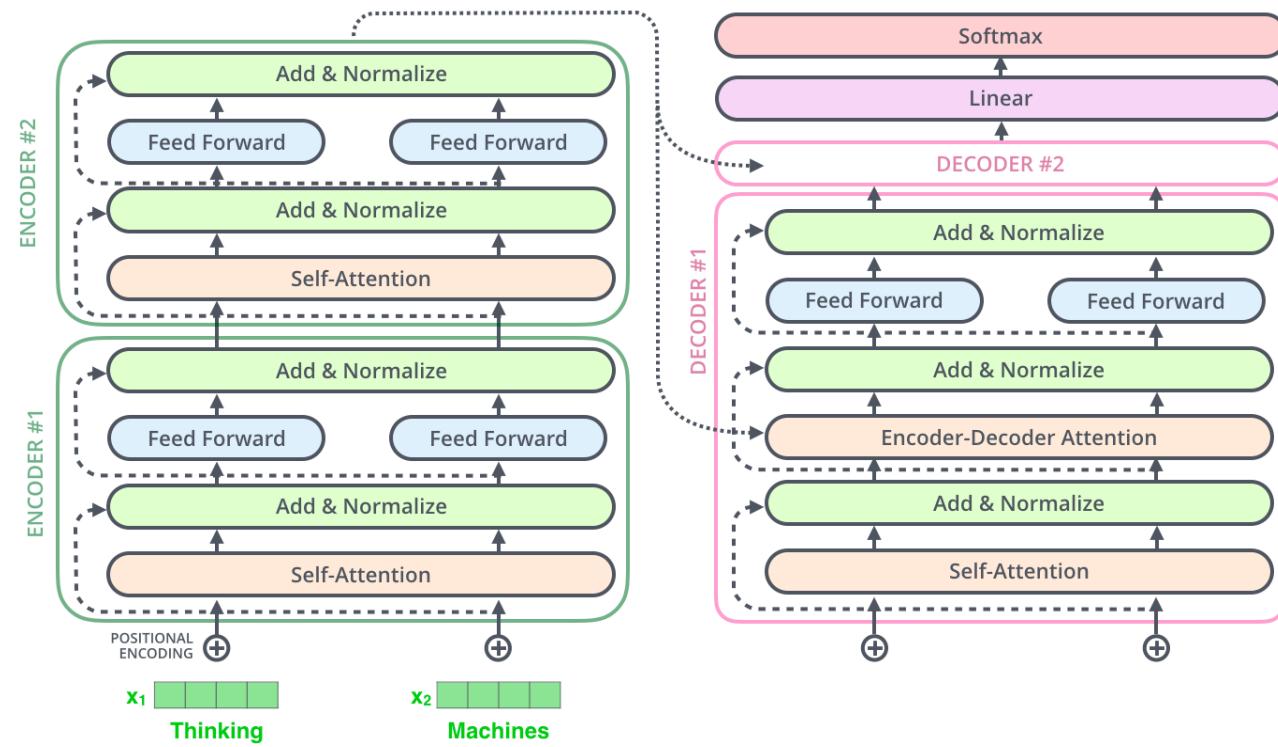
Ashish Vaswani *et al.*
Attention is All you Need. NIPS 2017: 5998-6008.



Encoder-Decoder

- Earlier Models
- Model T5
- Model BART
- Beam Search

Translation Model

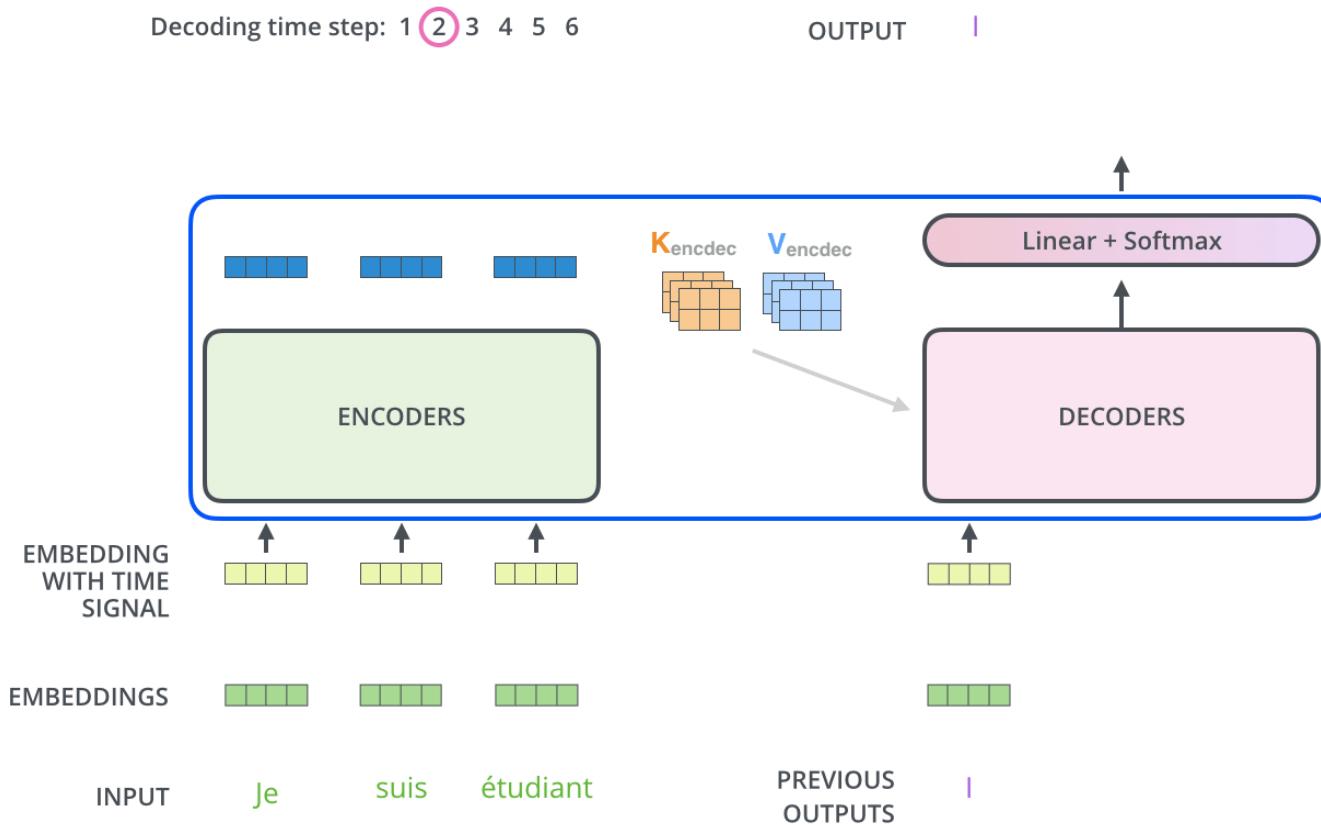


Reference: <https://jalammar.github.io/>

Encoder-Decoder

- Earlier Models
- Model T5
- Model BART
- Beam Search

Animation of the Translation Model



بسم الله الرحمن الرحيم

پردازش زبانهای طبیعی

جلسه ۱۹

احسان الدین عسگری

اردیبهشت ۱۴۰۳

<http://language.ml/>

دانشکده مهندسی کامپیوتر

آزمایشگاه پردازش هوشمند متن و زبان و علوم انسانی محاسباتی

asgari@berkeley.edu

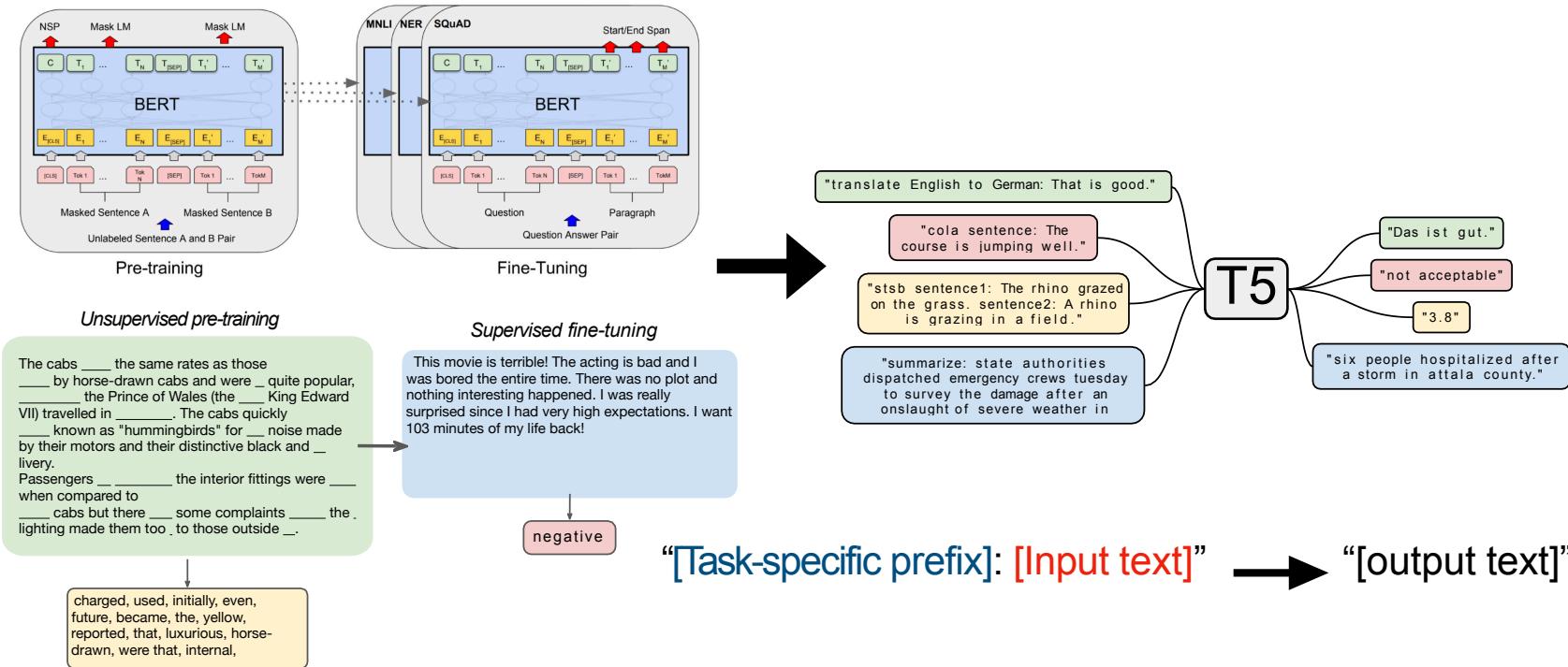


Encoder-Decoder

- Earlier Models
- Model T5
- Model BART
- Beam Search

Basic Idea of T5

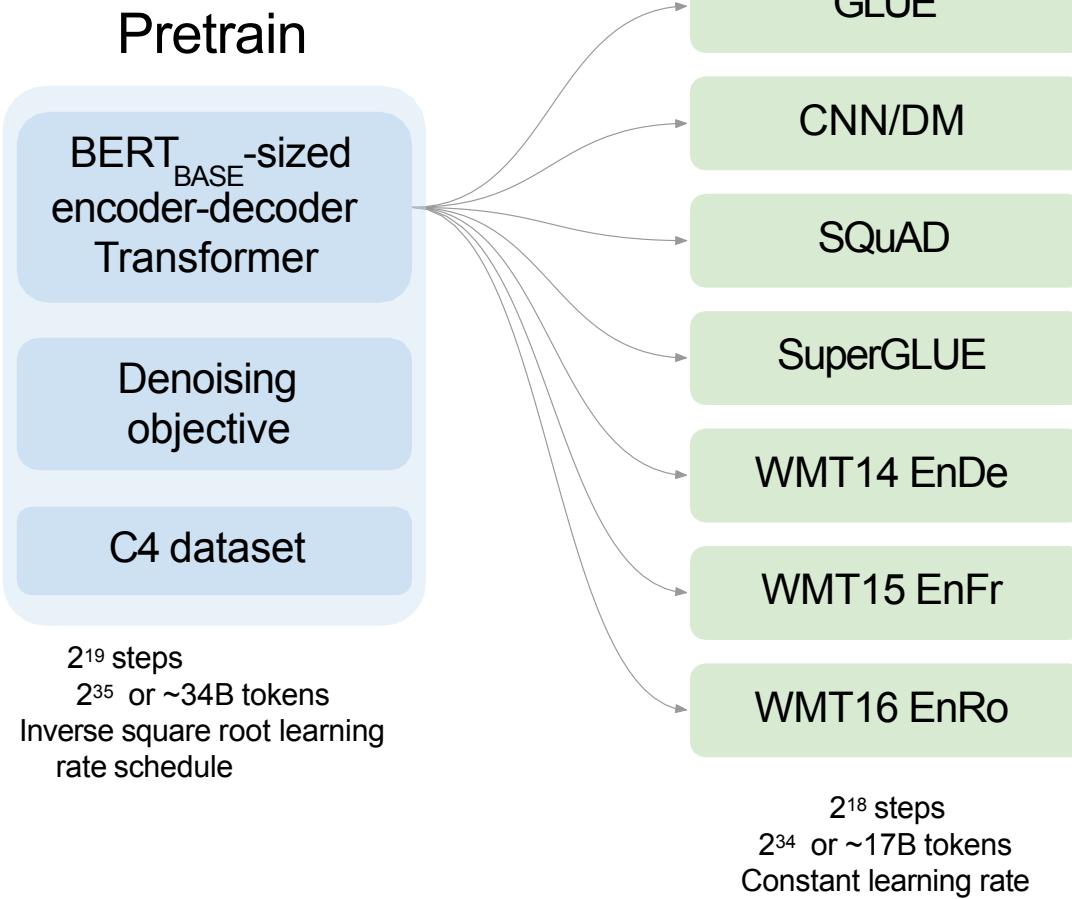
- Text-to-Text Transfer Transformer
- Moving from task-specific fine-tuning of language models → Single model for all



Encoder-Decoder

- Earlier Models
- Model T5
- Model BART
- Beam Search

Objective



Encoder-Decoder

- Earlier Models
- Model T5
- Model BART
- Beam Search

Denoising Objective

Original text

Thank you for inviting me to your party last week.

Inputs

Thank you <X> me to your party <Y> week.

Targets

<X> for inviting <Y> last <Z>

Encoder-Decoder

- Earlier Models
- Model T5
- Model BART
- Beam Search

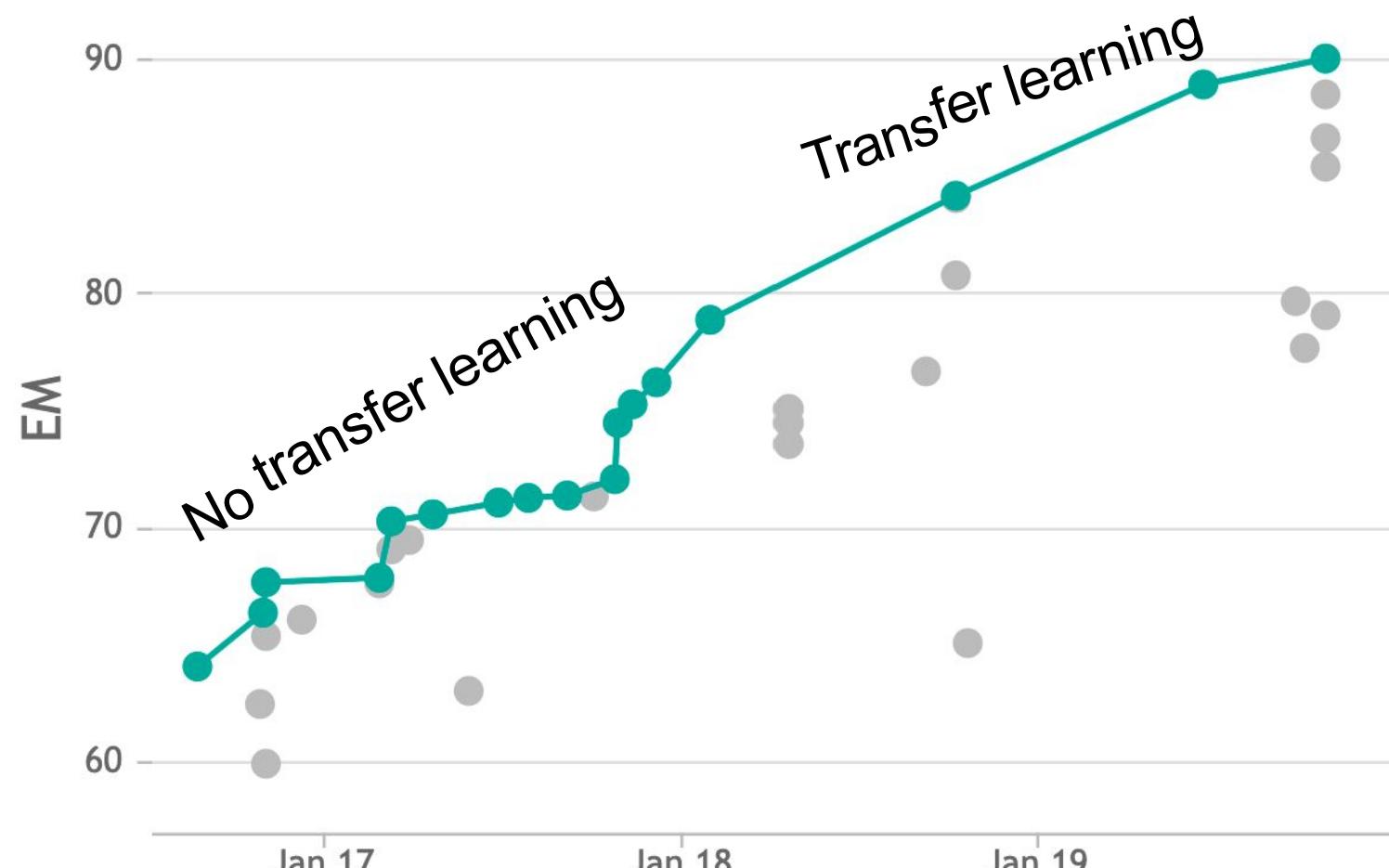
Finetuning Examples

- CoLA(GLUE): Sentence acceptability
 - **Input:** sentence, **output:** labels “acceptable” or “not acceptable”
 - Ex: “The course is jumping well.” -> not acceptable
- STSB (GLUE): Sentence similarity
 - **Input:** pair of sentences, **output:** similarity score [1,5]
 - Ex: “sentence1: The rhino grazed. sentence2: Arhino is grazing.” -> 3.8
- COPA(SuperGLUE): Causal reasoning
 - **Input:** premise and 2 alternatives, **output:** alternative1 or alternative2
 - Ex: “Premise: I tipped the bottle. What happened as a RESULT? Alternative 1: The liquid in the bottle froze. Alternative 2: The liquid in the bottle poured out.” -> alternative2
- EnDe (Translation):
“translate English to German: **That is good**” -> “Das ist gut”
- CNNDM (Summarization):
“**summarize: state authorities dispatched...**” -> “six people hospitalized after storm”

Encoder-Decoder

- Earlier Models
- Model T5
- Model BART
- Beam Search

Every task is Question-Answering

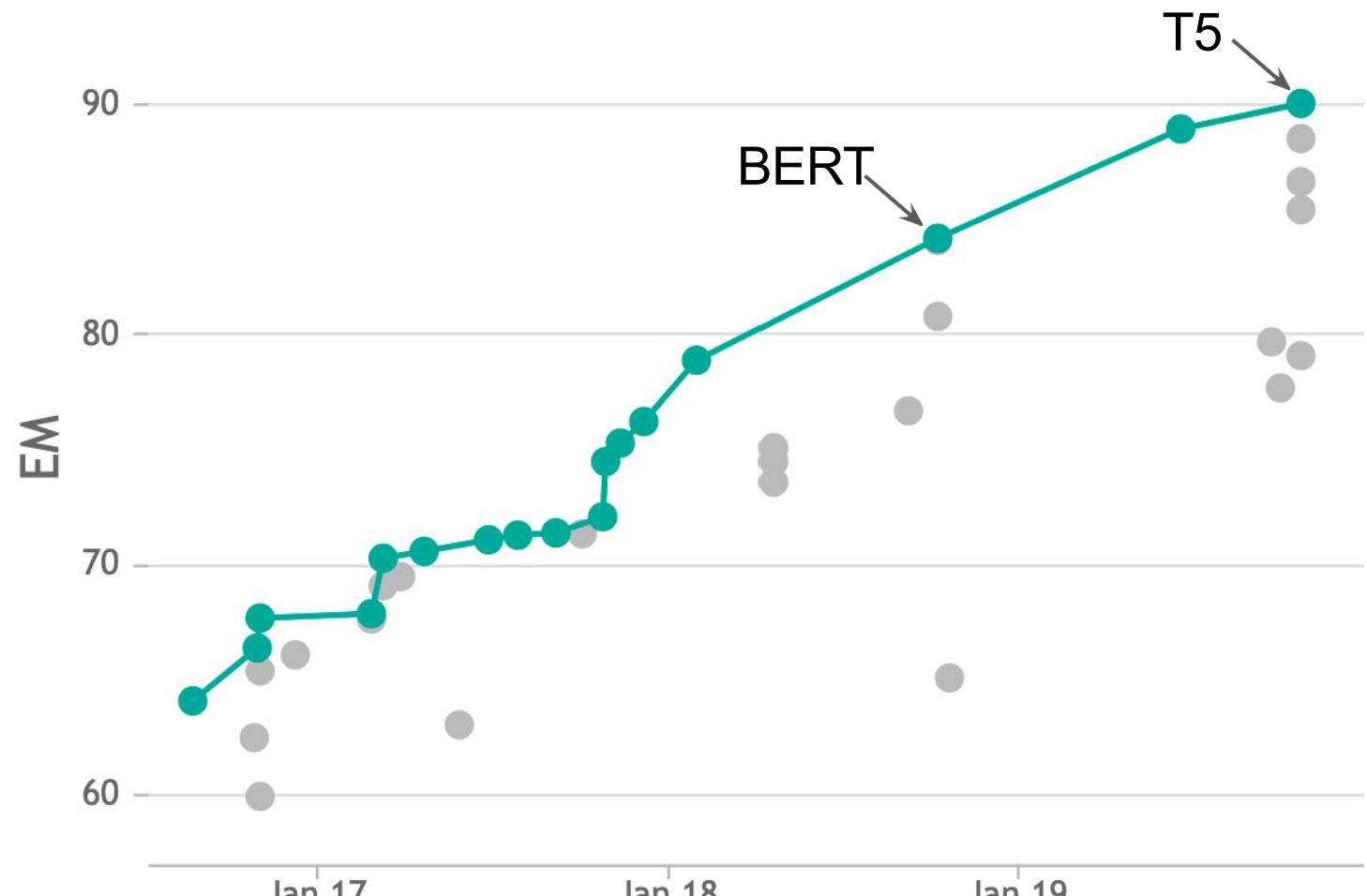


Source: <https://paperswithcode.com/sota/question-answering-on-squad11-dev>

Encoder-Decoder

- Earlier Models
- Model T5
- Model BART
- Beam Search

Every task is Question-Answering

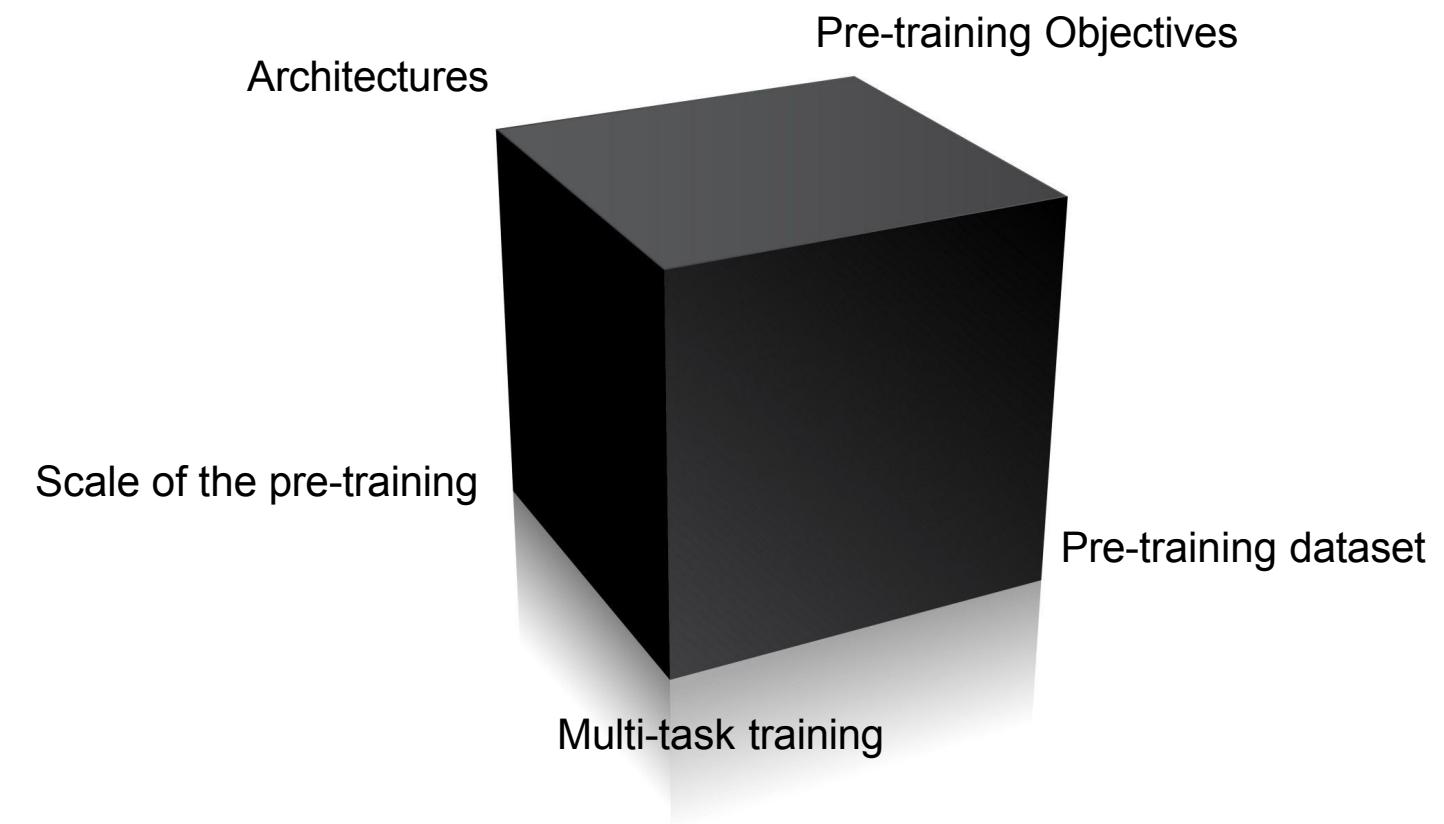


Source: <https://paperswithcode.com/sota/question-answering-on-squad11-dev>

Encoder-Decoder

- Earlier Models
- Model T5
- Model BART
- Beam Search

Trying different decisions for Pre-training and Fine-tuning



Encoder-Decoder

- Earlier Models
- Model T5
- Model BART
- Beam Search

Training Dataset

- **C4 Dataset: Colossal Clean Crawled Corpus**
- Web-extracted text
- English language only (langdetect)
- Extreme cleaning and filtering: 20TB → 750GB

Menu

The lemon, Citrus Limon (L.) Osbeck, is a species of small evergreen tree in the flowering plant family Rutaceae. The tree's ellipsoidal yellow fruit is used for culinary and non-culinary purposes throughout the world, primarily for its juice, which has both culinary and cleaning uses. The juice of the lemon is about 5% to 6% citric acid, with a pH of around 2.2, giving it a sour taste.

Article

The origin of the lemon is unknown, though lemons are thought to have first grown in Assam (a region in northeast India), northern Burma or China. A genomic study of the lemon indicated it was a hybrid between bitter orange (sour orange) and citron.

Please enable JavaScript to use our site.

Dried Lemons, \$3.59/pound

Organic dried lemons from our farm in California. Lemons are harvested and sun-dried for maximum flavor. Good in soups and on popcorn.

The lemon, Citrus Limon (L.) Osbeck, is a species of small evergreen tree in the flowering plant family Rutaceae. The tree's ellipsoidal yellow fruit is used for culinary and non-culinary purposes throughout the world, primarily for its juice, which has both culinary and cleaning uses. The juice of the lemon is about 5% to 6% citric acid, with a pH of around 2.2, giving it a sour taste.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Curabitur in tempus quam. In mollis et ante at consectetur. Aliquam erat volutpat. Donec at lacinia est. Duis semper, magna tempor interdum suscipit, ante elit molestie urna, eget efficitur risus nunc ac elit. Fusce quis blandit lectus. Mauris at mauris a turpis tristique lacinia at nec ante. Aenean in scelerisque tellus, a efficitur ipsum. Integer justo enim, ornare vitae sem non, mollis fermentum lectus. Mauris ultrices nisl at libero porta sodales in ac orci.

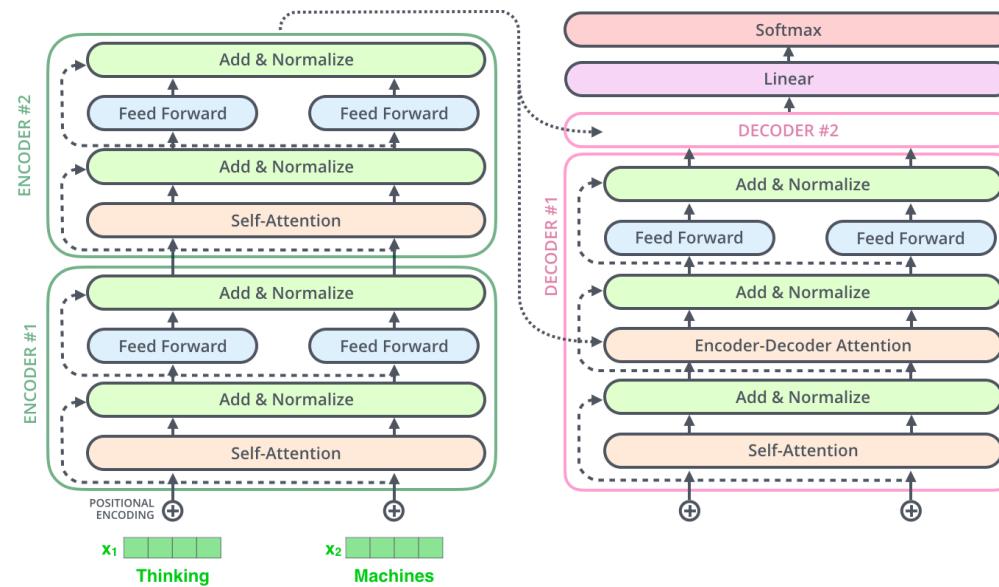
```
function Ball(r)
{this.radius = r;
```

Encoder-Decoder

- Earlier Models
- Model T5
- Model BART
- Beam Search

Vocab

- 32,000 wordpieces shared across input and output
- Pre-training is English, but fine-tuning includes German, French, and Romanian

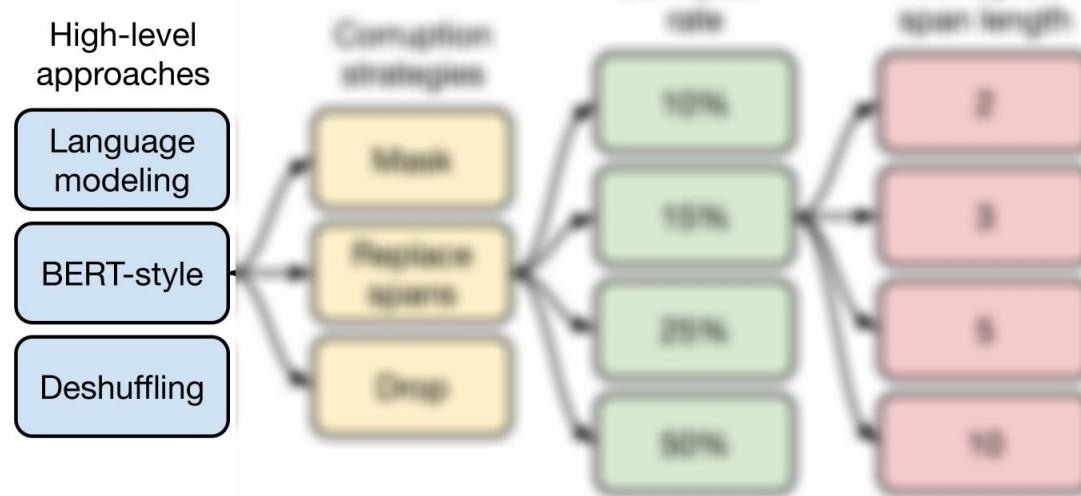


Encoder-Decoder

- Earlier Models
- Model T5
- Model BART
- Beam Search

Pretraining Objective

1. BERT-style objective performs best.
2. Prefix LM works well on translation tasks.
3. Deshuffling objective is significantly worse.



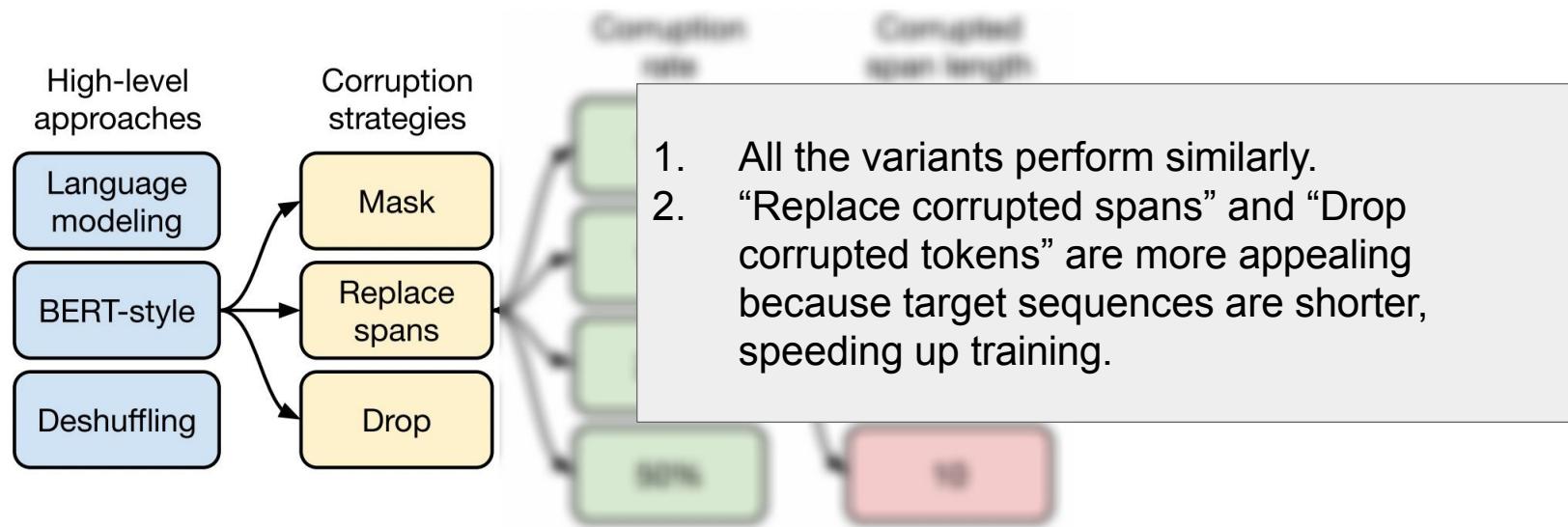
Objective	GLUE	CNNDM	SQuAD	SGLUE	EnDe	EnFr	EnRo
Prefix language modeling	80.69	18.94	77.99	65.27	26.86	39.73	27.49
BERT-style (Devlin et al., 2018)	82.96	19.17	80.65	69.85	26.78	40.03	27.41
Deshuffling	73.17	18.59	67.61	58.47	26.11	39.30	25.62

Objective	Inputs	Targets
Prefix language modeling	Thank you for inviting	me to your party last week .
BERT-style Devlin et al. (2018)	Thank you <M> <M> me to your party apple week .	(original text)
Deshuffling	party me for your to . last fun you inviting week Thank	(original text)

Encoder-Decoder

- Earlier Models
- Model T5
- Model BART
- Beam Search

Pretraining Objective

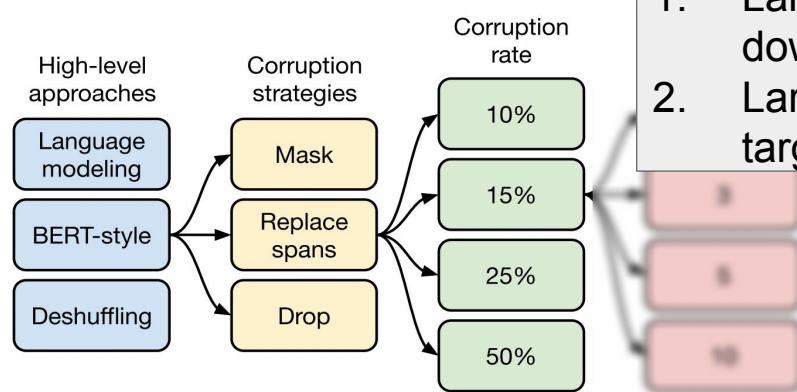


Objective	GLUE	CNNDM	SQuAD	SGLUE	EnDe	EnFr	EnRo
BERT-style (Devlin et al., 2018)	82.96	19.17	80.65	69.85	26.78	40.03	27.41
MASS-style (Song et al., 2019)	82.32	19.16	80.10	69.28	26.79	39.89	27.55
★ Replace corrupted spans	83.28	19.24	80.88	71.36	26.98	39.82	27.65
Drop corrupted tokens	84.44	19.31	80.52	68.67	27.07	39.76	27.82

Encoder-Decoder

- Earlier Models
- Model T5
- Model BART
- Beam Search

Different Corruption Rates



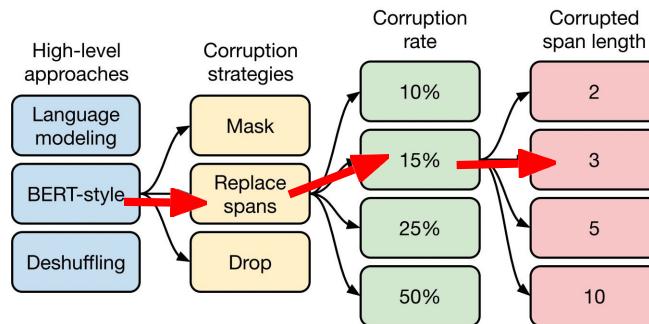
1. Larger corruption rate leads to downstream performance degradation.
2. Larger corruption rate also leads to longer targets, slowing down training.

Corruption rate	GLUE	CNNDM	SQuAD	SGLUE	EnDe	EnFr	EnRo
10%	82.82	19.00	80.38	69.55	26.87	39.28	27.44
★ 15%	83.28	19.24	80.88	71.36	26.98	39.82	27.65
25%	83.00	19.54	80.96	70.48	27.04	39.83	27.47
50%	81.27	19.32	79.80	70.33	27.01	39.90	27.49

Encoder-Decoder

- Earlier Models
- Model T5
- Model BART
- Beam Search

Span-corruption rate



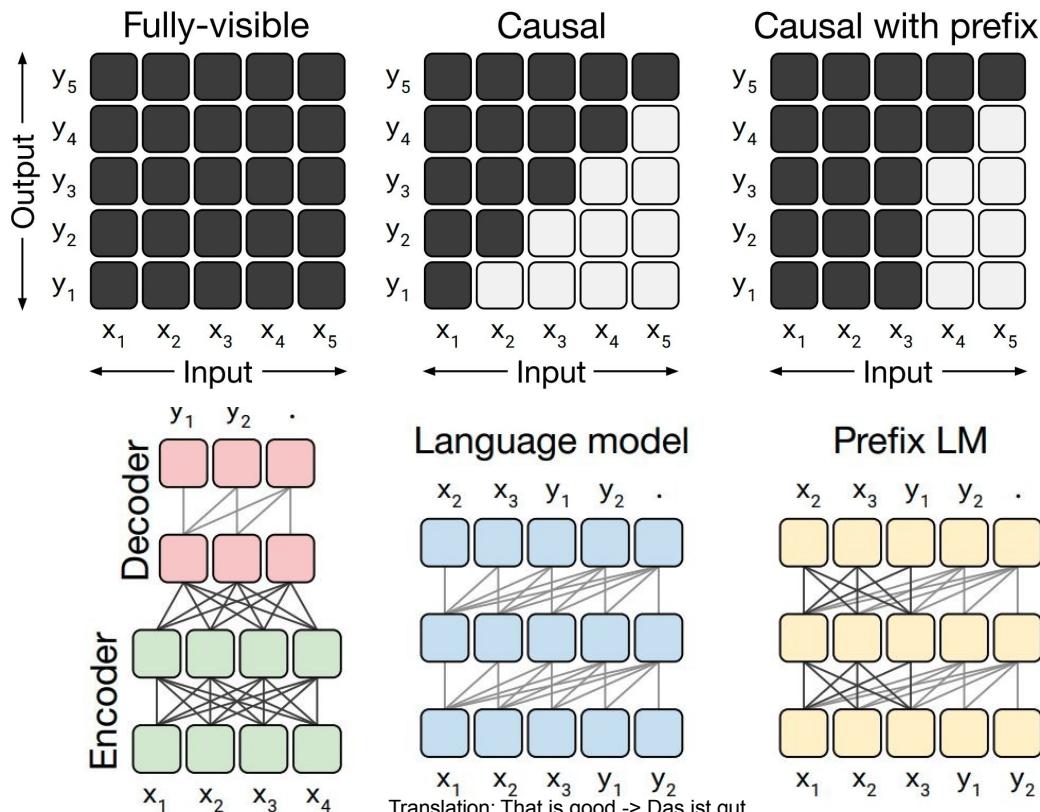
1. Average span length of 3 works well on most non-translation tasks.
2. Span corruption produces shorter target sequences and leads to speedup in training.

Span length	GLUE	CNNDM	SQuAD	SGLUE	EnDe	EnFr	EnRo
★ Baseline (i.i.d.)	83.28	19.24	80.88	71.36	26.98	39.82	27.65
2	83.54	19.39	82.09	72.20	26.76	39.99	27.63
3	83.49	19.62	81.84	72.53	26.86	39.65	27.62
5	83.40	19.24	82.05	72.23	26.88	39.40	27.53
10	82.85	19.33	81.84	70.44	26.79	39.49	27.69

Encoder-Decoder

- Earlier Models
- Model T5
- Model BART
- Beam Search

Architecture - Attention Mask



Translate English to German: That is good. Target: Das ist gut.

Translate English to German: That is good. Target: Das ist gut.
 “Good” representation can only look at “Translate English to German: That is”.

Translate English to German: That is good. Target: Das ist gut.
 “Good” representation can look at “Translate English to German: That is. Target:”.

Architecture	Objective	Params	Cost	GLUE	CNNDM	SQuAD	SGLUE	EnDe	EnFr	EnRo
★ Encoder-decoder	Denoising	$2P$	M	83.28	19.24	80.88	71.36	26.98	39.82	27.65
Enc-dec, shared	Denoising	P	M	82.81	18.78	80.63	70.73	26.72	39.03	27.46
Enc-dec, 6 layers	Denoising	P	$M/2$	80.88	18.97	77.59	68.42	26.38	38.40	26.95
Language model	Denoising	P	M	74.70	17.93	61.14	55.02	25.09	35.28	25.86
Prefix LM	Denoising	P	M	81.82	18.61	78.94	68.11	26.43	37.98	27.39

Encoder-Decoder

- Earlier Models
- Model T5
- Model BART
- Beam Search

Multitasking

Training strategy	GLUE	CNNDM	SQuAD	SGLUE	EnDe	EnFr	EnRo
★ Unsupervised pre-training + fine-tuning	83.28	19.24	80.88	71.36	26.98	39.82	27.65
Multi-task training	81.42	19.24	79.78	67.30	25.21	36.30	27.76
Multi-task pre-training + fine-tuning	83.11	19.12	80.26	71.03	27.08	39.80	28.07
Leave-one-out multi-task training	81.98	19.05	79.97	71.68	26.93	39.79	27.87
Supervised multi-task pre-training	79.93	18.96	77.38	65.36	26.81	40.13	28.04

1. Multi-task pre-training + fine-tuning works as well as unsupervised pre-training + fine-tuning.
2. Practical benefit of Multi-task pre-training + fine-tuning is to monitor downstream performance during pre-training.

Encoder-Decoder

- Earlier Models
- Model T5
- Model BART
- Beam Search

Architecture	Objective	Params	Cost	GLUE	CNNDM	SQuAD	SGLUE	EnDe	EnFr	EnRo
★ Encoder-decoder	Denoising	<i>2P</i>	<i>M</i>	83.28	19.24	80.88	71.36	26.98	39.82	27.65
Enc-dec, shared	Denoising	<i>P</i>	<i>M</i>	82.81	18.78	80.63	70.73	26.72	39.03	27.46
Enc-dec, 6 layers	Denoising	<i>P</i>	<i>M/2</i>	80.88	18.97	77.59	68.42	26.38	38.40	26.95
Language model	Denoising	<i>P</i>	<i>M</i>	74.70	17.93	61.14	55.02	25.09	35.28	25.86
Prefix LM	Denoising	<i>P</i>	<i>M</i>	81.82	18.61	78.94	68.11	26.43	37.98	27.39
Span length	GLUE	CNNDM	SQuAD	SGLUE	EnDe	EnFr	EnRo			
★ Baseline (i.i.d.)	83.28	19.24	80.88	71.36	26.98	39.82	27.65			
2	83.54	19.39	82.09	72.20	26.76	39.99	27.63			
3	83.49	19.62	81.84	72.53	26.86	39.65	27.62			
5	83.40	19.24	82.05	72.23	26.88	39.40	27.53			
10	82.85	19.33	81.84	70.44	26.79	39.49	27.69			
Data set	Size	GLUE	CNNDM	SQuAD	SGLUE	EnDe	EnFr	EnRo		
★ C4	745GB	83.28	19.24	80.88	71.36	26.98	39.82	27.65		
C4, unfiltered	6.1TB	81.46	19.14	78.78	68.04	26.55	39.34	27.21		
RealNews-like	35GB	83.83	19.23	80.39	72.38	26.75	39.90	27.48		
WebText-like	17GB	84.03	19.31	81.42	71.40	26.80	39.74	27.59		
Wikipedia	16GB	81.85	19.31	81.29	68.01	26.94	39.69	27.67		
Wikipedia + TBC	20GB	83.65	19.28	82.08	73.24	26.77	39.63	27.57		
Training strategy		GLUE	CNNDM	SQuAD	SGLUE	EnDe	EnFr	EnRo		
★ Unsupervised pre-training + fine-tuning		83.28	19.24	80.88	71.36	26.98	39.82	27.65		
Multi-task training		81.42	19.24	79.78	67.30	25.21	36.30	27.76		
Multi-task pre-training + fine-tuning		83.11	19.12	80.26	71.03	27.08	39.80	28.07		
Leave-one-out multi-task training		81.98	19.05	79.97	71.68	26.93	39.79	27.87		
Supervised multi-task pre-training		79.93	18.96	77.38	65.36	26.81	40.13	28.04		

Scaling strategy	GLUE	CNNDM	SQuAD	SGLUE	EnDe	EnFr	EnRo
★ Baseline	83.28	19.24	80.88	71.36	26.98	39.82	27.65
1× size, 4× training steps	85.33	19.33	82.45	74.72	27.08	40.66	27.93
1× size, 4× batch size	84.60	19.42	82.52	74.64	27.07	40.60	27.84
2× size, 2× training steps	86.18	19.66	84.18	77.18	27.52	41.03	28.19
4× size, 1× training steps	85.91	19.73	83.86	78.04	27.47	40.71	28.10
4× ensembled	84.77	20.10	83.09	71.74	28.05	40.53	28.57
4× ensembled, fine-tune only	84.05	19.57	82.36	71.55	27.55	40.22	28.09

Encoder-Decoder

- Earlier Models
- Model T5
- Model BART
- Beam Search

Model Size

Model	Parameters	No. of layers	d_{model}	d_{ff}	d_{kv}	No. of heads
Small	60M	6	512	2048	64	8
Base	220M	12	768	3072	64	12
Large	770M	24	1024	4096	64	16
3B	3B	24	1024	16384	128	32
11B	11B	24	1024	65536	128	128

Model	GLUE	CNNDM	SQuAD	SGLUE	EnDe	EnFr	EnRo
Previous best	89.4	20.30	95.5	84.6	33.8	43.8	38.5
T5-Small	77.4	19.56	87.24	63.3	26.7	36.0	26.8
T5-Base	82.7	20.34	92.08	76.2	30.9	41.2	28.0
T5-Large	86.4	20.68	93.79	82.3	32.0	41.5	28.1
T5-3B	88.5	21.02	94.95	86.4	31.8	42.6	28.2
T5-11B	89.7	21.55	95.64	88.9	32.1	43.4	28.1

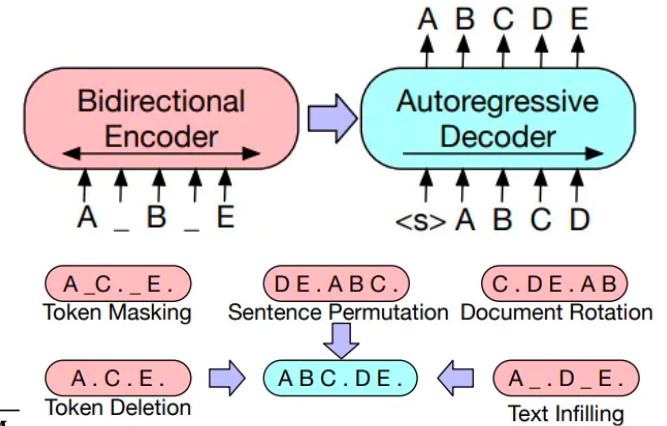
Encoder-Decoder

- Earlier Models
- Model T5
- Model BART
- Beam Search

BART

Bidirectional and Auto-Regressive Transformers (BART)

- A bidirectional encoder and an autoregressive decoder.
- BART achieves the state of the art results in the summarization task.



Model	SQuAD 1.1	MNLI	ELI5	XSum	ConvAI2	CNN/DM
	F1	Acc	PPL	PPL	PPL	PPL
BERT Base (Devlin et al., 2019)	88.5	84.3	-	-	-	-
Masked Language Model	90.0	83.5	24.77	7.87	12.59	7.06
Masked Seq2seq	87.0	82.1	23.40	6.80	11.43	6.19
Language Model	76.7	80.1	21.40	7.00	11.51	6.56
Permuted Language Model	89.1	83.7	24.03	7.69	12.23	6.96
Multitask Masked Language Model	89.2	82.4	23.73	7.50	12.39	6.74
BART Base						
w/ Token Masking	90.4	84.1	25.05	7.08	11.73	6.10
w/ Token Deletion	90.4	84.1	24.61	6.90	11.46	5.87
w/ Text Infilling	90.8	84.0	24.26	6.61	11.05	5.83
w/ Document Rotation	77.2	75.3	53.69	17.14	19.87	10.59
w/ Sentence Shuffling	85.4	81.5	41.87	10.93	16.67	7.89
w/ Text Infilling + Sentence Shuffling	90.8	83.8	24.17	6.62	11.12	5.41

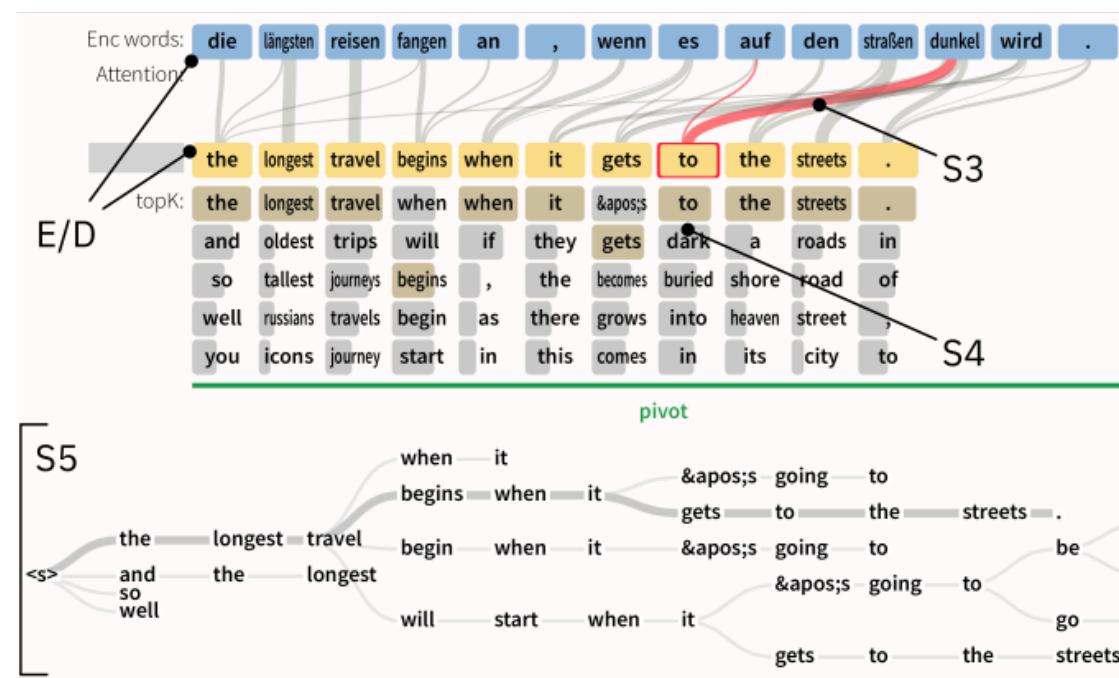
Lewis, Mike, et al.

"Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension."
ACL 2020.

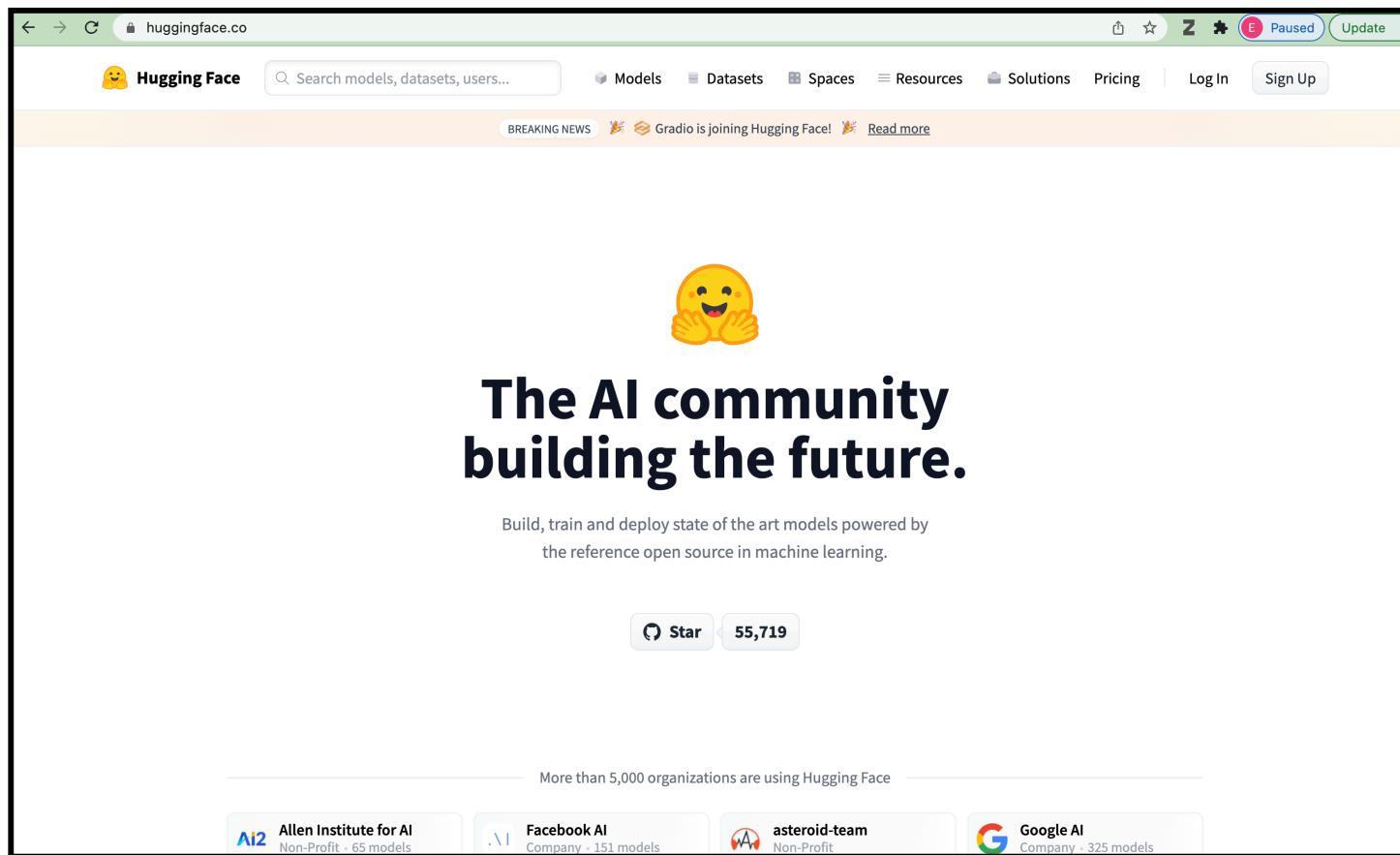
Encoder-Decoder

- Earlier Models
- Model T5
- Model BART
- Beam Search

Beam Search for Decoding



HuggingFace



The screenshot shows the homepage of the Hugging Face website. At the top, there is a navigation bar with links for Models, Datasets, Spaces, Resources, Solutions, Pricing, Log In, and Sign Up. A search bar is also present. Below the navigation bar, there is a banner with the text "BREAKING NEWS" and "Gradio is joining Hugging Face!" followed by a "Read more" link. The main feature of the page is a large yellow emoji of a smiling face with hands clasped together. Below the emoji, the text "The AI community building the future." is displayed in a large, bold, black font. Underneath this text, there is a smaller description: "Build, train and deploy state of the art models powered by the reference open source in machine learning." To the right of this description is a button labeled "Star" with the number "55,719" next to it. At the bottom of the page, there is a section titled "More than 5,000 organizations are using Hugging Face" with logos for Allen Institute for AI, Facebook AI, asteroid-team, and Google AI.

huggingface.co

Hugging Face

Models Datasets Spaces Resources Solutions Pricing Log In Sign Up

BREAKING NEWS Gradio is joining Hugging Face! Read more

The AI community building the future.

Build, train and deploy state of the art models powered by the reference open source in machine learning.

Star 55,719

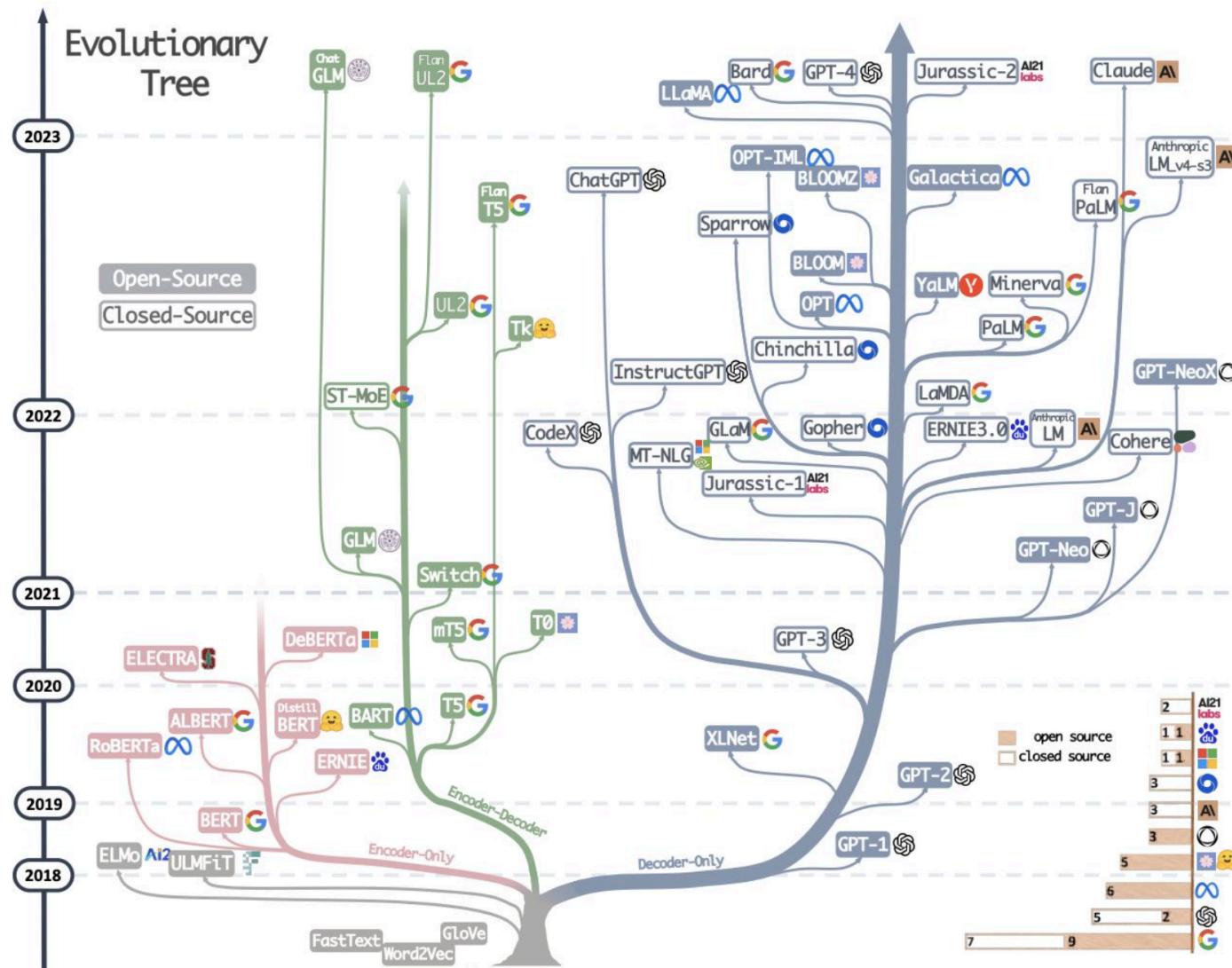
More than 5,000 organizations are using Hugging Face

Allen Institute for AI Non-Profit - 65 models

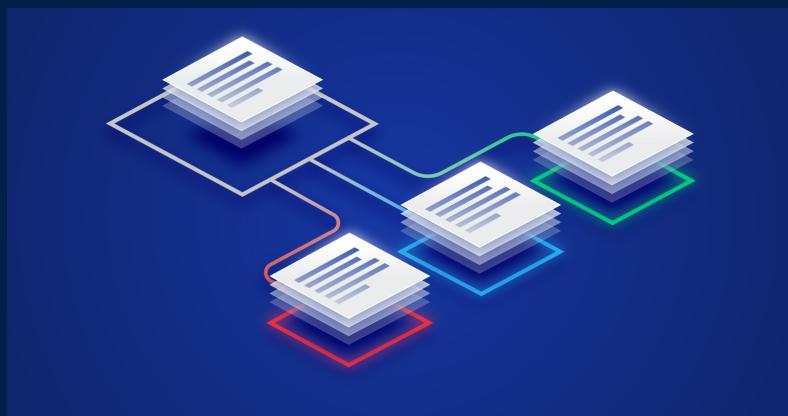
Facebook AI Company - 151 models

asteroid-team Non-Profit

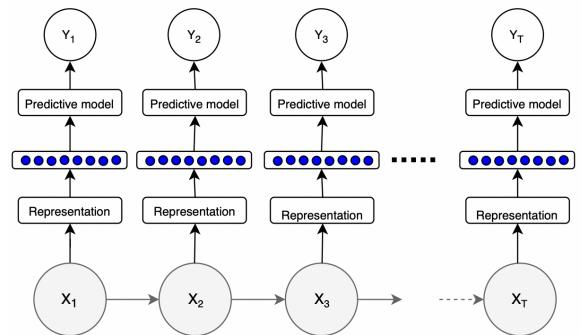
Google AI Company - 325 models



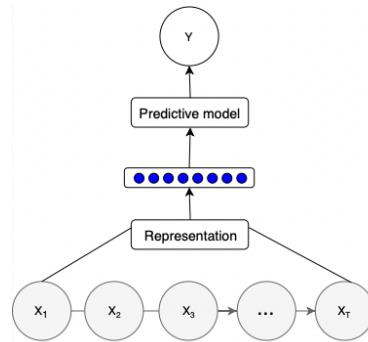
طبقه‌بندی متن یا classification



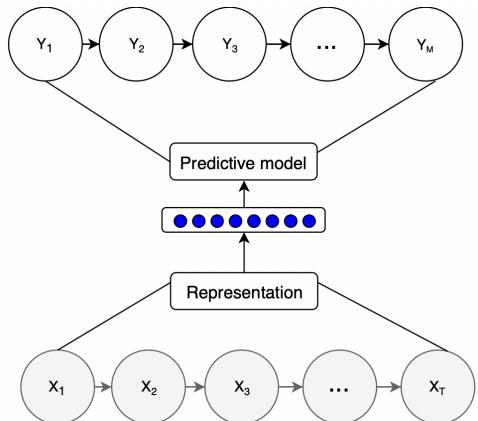
مسائل پردازش زبان



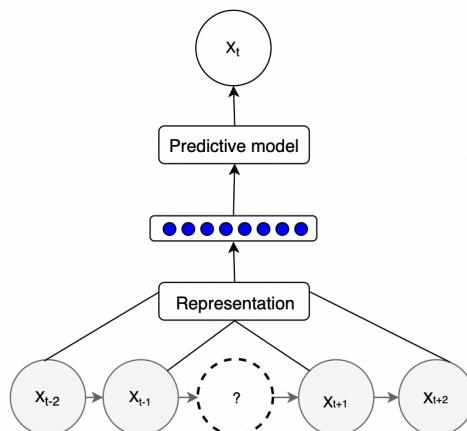
برچسب زنی



طبقه بندی



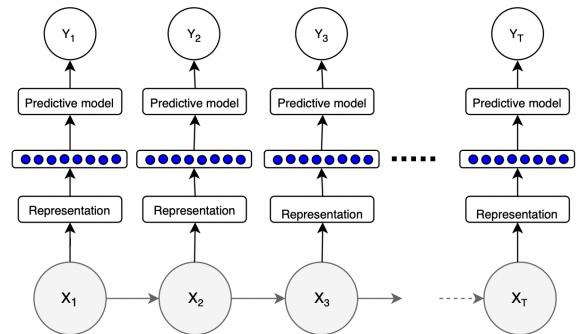
تولید متن



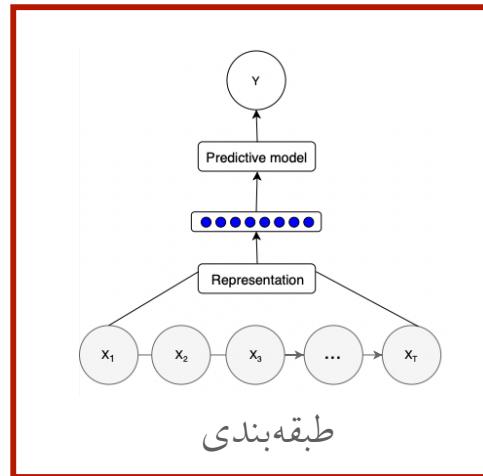
مدل زبانی

- معرفی طبقه بندی
- ارزیابی و یادگیری طبقه بندی
- **naive bayes** مدل
- **logistic regression** مدل
- بازنمایی مناسب برای تسک
- شبکه های عصبی دیگر

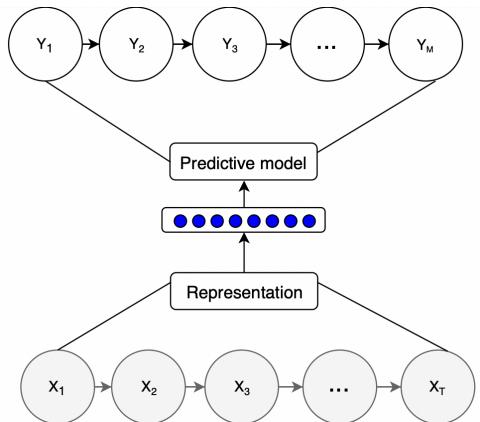
مسائل پردازش زبان



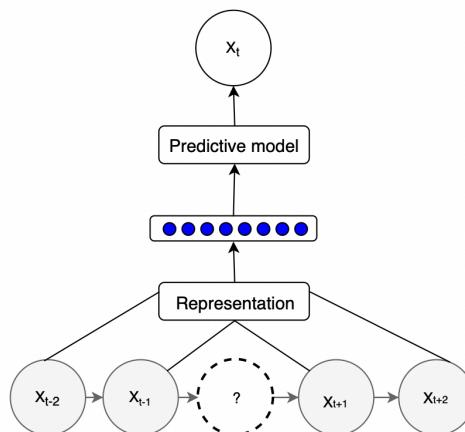
برچسبزنی



طبقه‌بندی



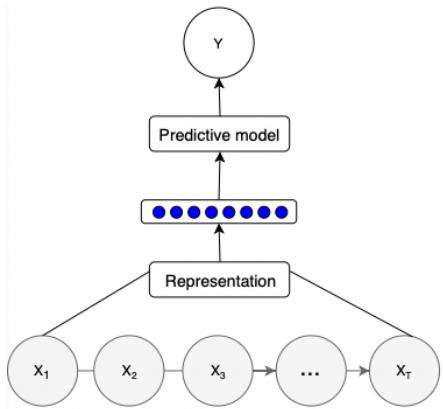
تولید متن



مدل زبانی

- معرفی طبقه‌بندی
- ارزیابی و یادگیری طبقه‌بندی
- **naive bayes** مدل
- **logistic regression** مدل
- بازنمایی مناسب برای تسک
- شبکه‌های عصبی دیگر

طبقه‌بندی



- یک مجموعه از برچسبها را می‌پذیرد.
- ورودی یک رشته متنی است.
- یادگیری با ناظر

task	x	y
language ID	text	{english, mandarin, greek, ...}
spam classification	email	{spam, not spam}
authorship attribution	text	{jk rowling, james joyce, ...}
genre classification	novel	{detective, romance, gothic, ...}
sentiment analysis	text	{positive, negative, neutral, mixed}

- معرفی طبقه‌بندی
- ارزیابی و یادگیری طبقه‌بندی
- **naive bayes**
- **logistic regression**
- مدل
- بازنمایی مناسب برای تسک
- شبکه‌های عصبی دیگر

بسم الله الرحمن الرحيم

پردازش زبانهای طبیعی

جلسه ۱۸

احسان الدین عسگری

آذر ۱۴۰۲

<http://language.ml/>

دانشکده مهندسی کامپیوتر

آزمایشگاه پردازش هوشمند متن و زبان و علوم انسانی محاسباتی

asgari@berkeley.edu



طبقه‌بندی

- داده یادگیری

$$\mathcal{D} = \{(x^1, y^1), \dots, (x^M, y^M)\} \subseteq \mathcal{X} \times \mathcal{Y}.$$

- داده تست

$x \in \mathcal{X}$ (test example)

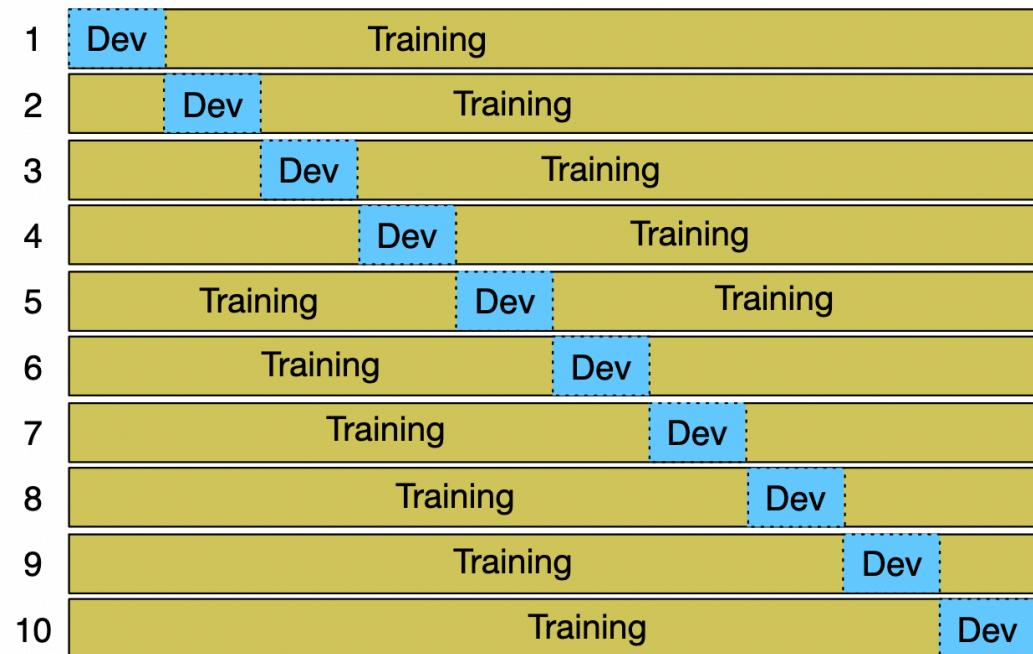
- مدل یادگیری

$$\hat{y} = h(x)$$

- معرفی طبقه‌بندی
- ارزیابی و یادگیری طبقه‌بندی
- مدل **naive bayes**
- مدل **logistic regression**
- بازنمایی مناسب برای تلسکوپ
- شبکه‌های عصبی دیگر

Cross-validation

Training Iterations



Testing

Test Set

- معرفی طبقه‌بندی
- ارزیابی و یادگیری طبقه‌بندی
- **naive bayes** مدل
- **logistic regression** مدل
- بازنمایی مناسب برای تسک
- شبکه‌های عصبی دیگر

ارزیابی طبقه‌بندی

		<i>gold standard labels</i>		
		gold positive	gold negative	
<i>system output labels</i>	system positive	true positive	false positive	$\text{precision} = \frac{\text{tp}}{\text{tp} + \text{fp}}$
	system negative	false negative	true negative	
		$\text{recall} = \frac{\text{tp}}{\text{tp} + \text{fn}}$		$\text{accuracy} = \frac{\text{tp} + \text{tn}}{\text{tp} + \text{fp} + \text{tn} + \text{fn}}$

ارزیابی طبقه‌بندی

		gold labels			
		urgent	normal	spam	
system output	urgent	8	10	1	$\text{precision}_u = \frac{8}{8+10+1}$
	normal	5	60	50	$\text{precision}_n = \frac{60}{5+60+50}$
	spam	3	30	200	$\text{precision}_s = \frac{200}{3+30+200}$
		$\text{recall}_u = \frac{8}{8+5+3}$	$\text{recall}_n = \frac{60}{10+60+30}$	$\text{recall}_s = \frac{200}{1+50+200}$	

- معرفی طبقه‌بندی

- ارزیابی و یادگیری طبقه‌بندی

- مدل **naive bayes**

- مدل **logistic regression**

- بازنمایی مناسب برای تسک

- شبکه‌های عصبی دیگر

بسم الله الرحمن الرحيم

پردازش زبانهای طبیعی

جلسه ۲۰

احسان الدین عسگری

اردیبهشت ۱۴۰۳

<http://language.ml/>

دانشکده مهندسی کامپیوتر

آزمایشگاه پردازش هوشمند متن و زبان و علوم انسانی محاسباتی

asgari@berkeley.edu



- معرفی طبقه‌بندی

- ارزیابی و یادگیری طبقه‌بندی

- **naive bayes**

- **logistic regression**

- بازنمایی مناسب برای تسک

- شبکه‌های عصبی دیگر

Generative Classifiers: Naive Bayes

$$\hat{y} = \arg \max_{y \in \mathcal{Y}} P(y|x)$$

$$P(Y = \oplus | X = \dots) \text{ از این ناراحتم که ..} = 0.088$$

$$P(Y = \ominus | X = \dots) \text{ از این ناراحتم که ..} = 0.912$$

$$= p(x|y) P(y)/P(x)$$

$$\sim P(x|y)P(y)$$

Unigram lang model

$$= \operatorname{argmax} P(y) \prod P(f_i|y)$$

$$= \operatorname{argmax} \log p(y) + \sum \log P(f_i|y)$$

- معرفی طبقه‌بندی
- ارزیابی و یادگیری طبقه‌بندی
- مدل **naive bayes**
- مدل **logistic regression**
- بازنمایی مناسب برای تسک
- شبکه‌های عصبی دیگر

Generative Classifiers: Naive Bayes - Inference

$$\hat{y} = \arg \max_{y \in \mathcal{Y}} \hat{P}(y) \hat{P}(x|y) \quad \text{maximum likelihood estimation}$$

- معرفی طبقه‌بندی
 - ارزیابی و یادگیری طبقه‌بندی
 - مدل **naive bayes**
 - مدل **logistic regression**
 - بازنمایی مناسب برای تسک
 - شبکه‌های عصبی دیگر

Generative Classifiers: Naive Bayes - Inference

$$\hat{y} = \arg \max_{y \in \mathcal{Y}} \hat{P}(y) \hat{P}(x|y) \quad \text{maximum likelihood estimation}$$

- معرفی طبقه‌بندی
- ارزیابی و یادگیری طبقه‌بندی
- **naive bayes** مدل
- **logistic regression** مدل
- بازنمایی مناسب برای تسک
- شبکه‌های عصبی دیگر

Generative Classifiers: Naive Bayes - Inference

$$\hat{y} = \arg \max_{y \in \mathcal{Y}} \hat{P}(y) \hat{P}(x|y) \quad \text{maximum likelihood estimation}$$

	Positive				Negative					
	x ₁	x ₂	x ₃	x ₄	x ₅	x ₆	x ₇	x ₈	p _{MLE,P}	p _{MLE,N}
f ₁	1	0	0	0	1	1	0	0	1/7	2/7
f ₂	0	0	0	0	0	0	1	0	0/7	1/7
f ₃	1	1	1	1	1	0	0	1	4/7	2/7
f ₄	1	0	0	1	1	0	0	1	2/7	2/7
f ₅	0	0	0	0	0	0	0	0	0/7	0/7

Log 0.5 Log 0.5

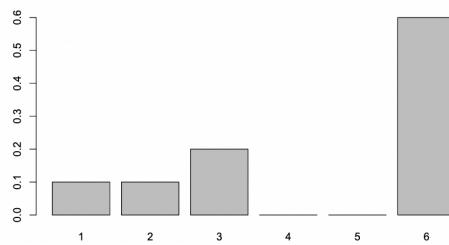
$$\hat{\theta}_{p,f_i} = \frac{N_{p,f_i}}{N_p}$$

- معرفی طبقه‌بندی
- ارزیابی و یادگیری طبقه‌بندی
- **naive bayes**
- **logistic regression**
- مدل
- بازنمایی مناسب برای تسک
- شبکه‌های عصبی دیگر

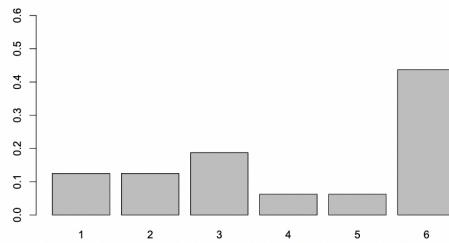
Generative Classifiers: Naive Bayes - Smoothing

$$\hat{y} = \arg \max_{y \in \mathcal{Y}} \hat{P}(y) \hat{P}(x|y) \quad \text{maximum likelihood estimation}$$

MLE



smoothing with $\alpha = 1$

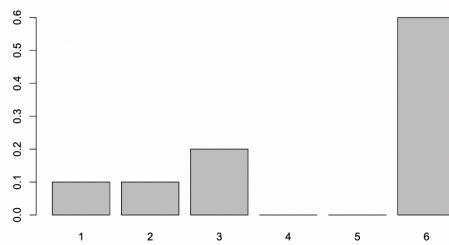


- معرفی طبقه‌بندی
- ارزیابی و یادگیری طبقه‌بندی
- **naive bayes**
- **logistic regression**
- مدل
- بازنمایی مناسب برای تسک
- شبکه‌های عصبی دیگر

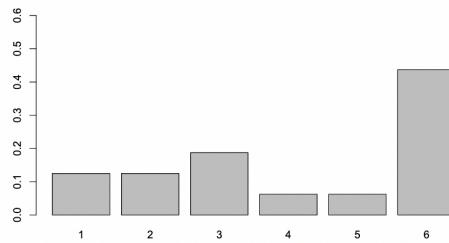
Generative Classifiers: Naive Bayes - Smoothing

$$\hat{y} = \arg \max_{y \in \mathcal{Y}} \hat{P}(y) \hat{P}(x|y) \quad \text{maximum likelihood estimation}$$

MLE



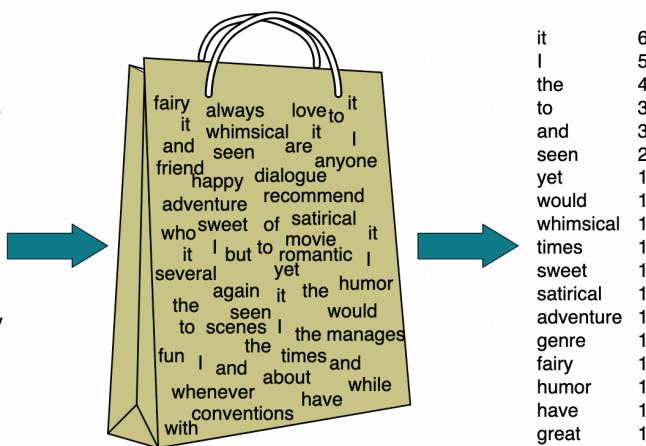
smoothing with $\alpha = 1$



- معرفی طبقه‌بندی
- ارزیابی و یادگیری طبقه‌بندی
- **naive bayes**
- **logistic regression**
- مدل تابعی مناسب برای تسخیح
- شبکه‌های عصبی دیگر

معاپب Naive Bayes

I love this movie! It's sweet, but with satirical humor. The dialogue is great and the adventure scenes are fun.... It manages to be whimsical and romantic while laughing at the conventions of the fairy tale genre. I would recommend it to just about anyone. I've seen it several times, and I'm always happy to see it again whenever I have a friend who hasn't seen it yet!



Discriminative Classifier - logistic regression

$$\hat{y} = \arg \max_{y \in \mathcal{Y}} P(y|x)$$

- معرفی طبقه‌بندی
- ارزیابی و یادگیری طبقه‌بندی
- مدل **naive bayes**
- مدل **logistic regression**
- بازنمایی مناسب برای تسک
- شبکه‌های عصبی دیگر

- معرفی طبقه‌بندی

- ارزیابی و یادگیری طبقه‌بندی

- مدل **naive bayes**

- مدل **logistic regression**

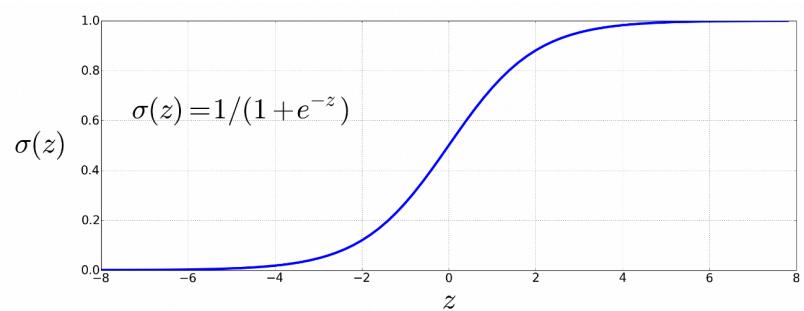
- بازنمایی مناسب برای تسک

- شبکه‌های عصبی دیگر

Discriminative Classifier - logistic regression

$$\arg \max_{y \in \mathcal{Y}} P(y|x)$$

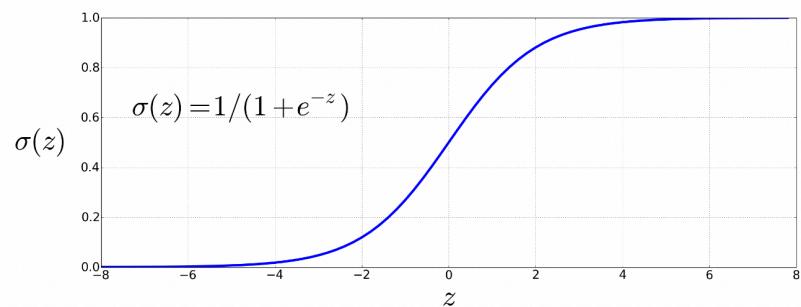
$$z = \left(\sum_{i=1}^n w_i x_i \right) + b = w \cdot x + b$$



Discriminative Classifier - logistic regression

$$\arg \max_{y \in \mathcal{Y}} P(y|x)$$

$$z = \left(\sum_{i=1}^n w_i x_i \right) + b = w \cdot x + b$$



$$P(y=1) = \sigma(w \cdot x + b)$$

$$P(y=0) = 1 - \sigma(w \cdot x + b)$$

$$= \sigma(-(w \cdot x + b))$$

- معرفی طبقه‌بندی
- ارزیابی و یادگیری طبقه‌بندی
- **naive bayes**
- **logistic regression**
- مدل
- بازنمایی مناسب برای تسک
- شبکه‌های عصبی دیگر

- معرفی طبقه‌بندی

- ارزیابی و یادگیری طبقه‌بندی

- مدل **naive bayes**

- مدل **logistic regression**

- بازنمایی مناسب برای تسک

- شبکه‌های عصبی دیگر

Discriminative Classifier - logistic regression

$$\hat{y} = \arg \max_{y \in \mathcal{Y}} P(y | x) \quad \left| \begin{array}{l} P(y=1) = \sigma(w \cdot x + b) \\ P(y=0) = \sigma(-(w \cdot x + b)) \end{array} \right.$$

- معرفی طبقه‌بندی
- ارزیابی و یادگیری طبقه‌بندی
- **naive bayes** مدل
- **logistic regression** مدل
- بازنمایی مناسب برای تسک
- شبکه‌های عصبی دیگر

Discriminative Classifier - logistic regression

$$\hat{y} = \arg \max_{y \in \{0, 1\}} P(y | x) \quad \left| \begin{array}{l} P(y=1) = \sigma(w \cdot x + b) \\ P(y=0) = \sigma(-(w \cdot x + b)) \end{array} \right.$$

\hat{y}
 $\sigma(w \cdot x + b)$
 $1 - \hat{y}$

$$P(y|x) = \begin{cases} y=1 \rightarrow \hat{y} & \rightarrow \hat{y}^y (1-\hat{y})^{1-y} \\ y=0 \rightarrow 1-\hat{y} & \end{cases}$$

$$-\log P(y|x) = -[y \log \hat{y} + (1-y) \log (1-\hat{y})]$$

logistic regression - cross-entropy loss

$$L_{\text{CE}}(\hat{y}, y) = -[y \log \sigma(w \cdot x + b) + (1 - y) \log (1 - \sigma(w \cdot x + b))]$$

- Laplace smoothing

$$\hat{\theta} = \operatorname{argmin}_{\theta} \frac{1}{m} \sum_{i=1}^m L_{\text{CE}}(f(x^{(i)}; \theta), y^{(i)})$$

$$\theta_{t+1} = \theta_t - \eta \nabla L(f(x; \theta), y)$$

logistic regression - cross-entropy loss

$$L_{\text{CE}}(\hat{y}, y) = -[y \log \sigma(w \cdot x + b) + (1 - y) \log (1 - \sigma(w \cdot x + b))]$$

- Laplace smoothing

$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}} \frac{1}{m} \sum_{i=1}^m L_{\text{CE}}(f(x^{(i)}; \theta), y^{(i)})$$

$$\theta_{t+1} = \theta_t - \eta \nabla L(f(x; \theta), y)$$

$$\frac{\partial L_{\text{CE}}(\hat{y}, y)}{\partial w_j} = [\sigma(w \cdot x + b) - y] x_j$$

```

function STOCHASTIC GRADIENT DESCENT( $L()$ ,  $f()$ ,  $x$ ,  $y$ ) returns  $\theta$ 
    # where: L is the loss function
    # f is a function parameterized by  $\theta$ 
    # x is the set of training inputs  $x^{(1)}$ ,  $x^{(2)}$ , ...,  $x^{(m)}$ 
    # y is the set of training outputs (labels)  $y^{(1)}$ ,  $y^{(2)}$ , ...,  $y^{(m)}$ 

     $\theta \leftarrow 0$ 
    repeat til done  # see caption
        For each training tuple  $(x^{(i)}, y^{(i)})$  (in random order)
            1. Optional (for reporting):      # How are we doing on this tuple?
                Compute  $\hat{y}^{(i)} = f(x^{(i)}; \theta)$   # What is our estimated output  $\hat{y}$ ?
                Compute the loss  $L(\hat{y}^{(i)}, y^{(i)})$  # How far off is  $\hat{y}^{(i)}$  from the true output  $y^{(i)}$ ?
            2.  $g \leftarrow \nabla_{\theta} L(f(x^{(i)}; \theta), y^{(i)})$       # How should we move  $\theta$  to maximize loss?
            3.  $\theta \leftarrow \theta - \eta g$                       # Go the other way instead
    return  $\theta$ 

```

Figure 5.5 The stochastic gradient descent algorithm. Step 1 (computing the loss) is used to report how well we are doing on the current tuple. The algorithm can terminate when it converges (or when the gradient norm $< \epsilon$), or when progress halts (for example when the loss starts going up on a held-out set).

- معرفی طبقه‌بندی

- ارزیابی و یادگیری طبقه‌بندی

- مدل **naive bayes**

- مدل **logistic regression**

- بازنمایی مناسب برای تسک

- شبکه‌های عصبی دیگر

logistic regression - regularisation

$$\hat{\theta} = \underset{\theta}{\operatorname{argmax}} \sum_{i=1}^m \log P(y^{(i)} | x^{(i)}) - \alpha R(\theta)$$

$$R(\theta) = \|\theta\|_2^2 = \sum_{j=1}^n \theta_j^2$$

$$R(\theta) = \|\theta\|_1 = \sum_{i=1}^n |\theta_i|$$

بسم الله الرحمن الرحيم

پردازش زبانهای طبیعی

جلسه ۲۱

احسان الدین عسگری

خرداد ۱۴۰۳

<http://language.ml/>

دانشکده مهندسی کامپیوتر

آزمایشگاه پردازش هوشمند متن و زبان و علوم انسانی محاسباتی

asgari@berkeley.edu



sentiment تحليل

Sentiment / Aspect

Annotation

Doc

Token

Neg_خوب

- معرفی طبقه‌بندی
- ارزیابی و یادگیری طبقه‌بندی
- مدل **naive bayes**
- مدل **logistic regression**
- بازنمایی مناسب برای تسک
- شبکه‌های عصبی دیگر

تحلیل sentiment

- Only positive/negative words in MPQA
- Only words in isolation (**bag of words**)
- Conjunctions of words (sequential, skip ngrams, other non-linear combinations)
- Higher-order linguistic structure (e.g., syntax)

- معرفی طبقه‌بندی

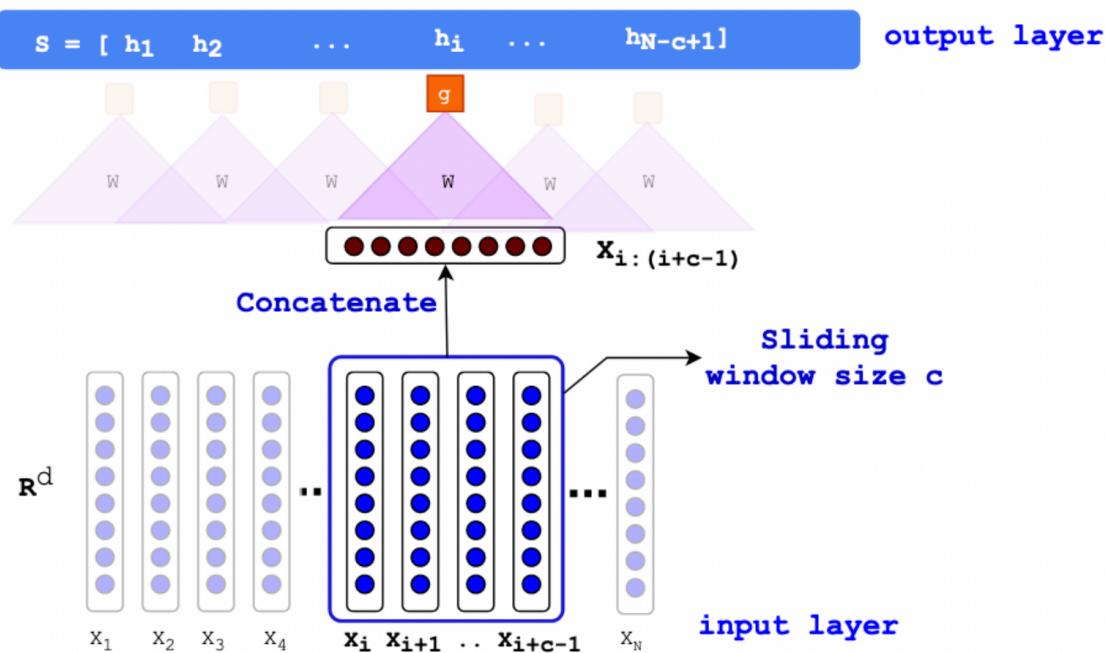
- ارزیابی و یادگیری طبقه‌بندی

- **naive bayes** مدل

- **logistic regression** مدل

- بازنمایی مناسب برای تسک

- شبکه‌های عصبی دیگر

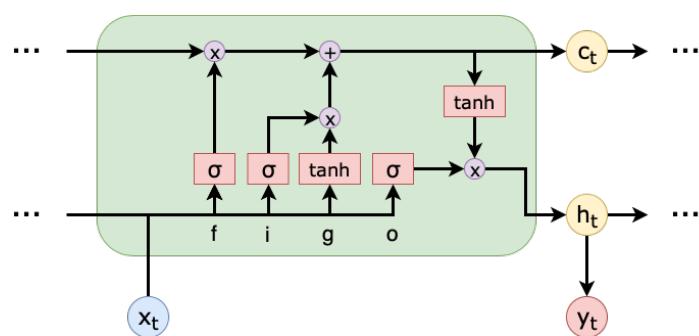


CNN-Classification

- معرفی طبقه‌بندی
- ارزیابی و یادگیری طبقه‌بندی
- **naive bayes**
- **logistic regression**
- مدل
- بازنمایی مناسب برای تسک
- شبکه‌های عصبی دیگر

LSTM-based – Classification

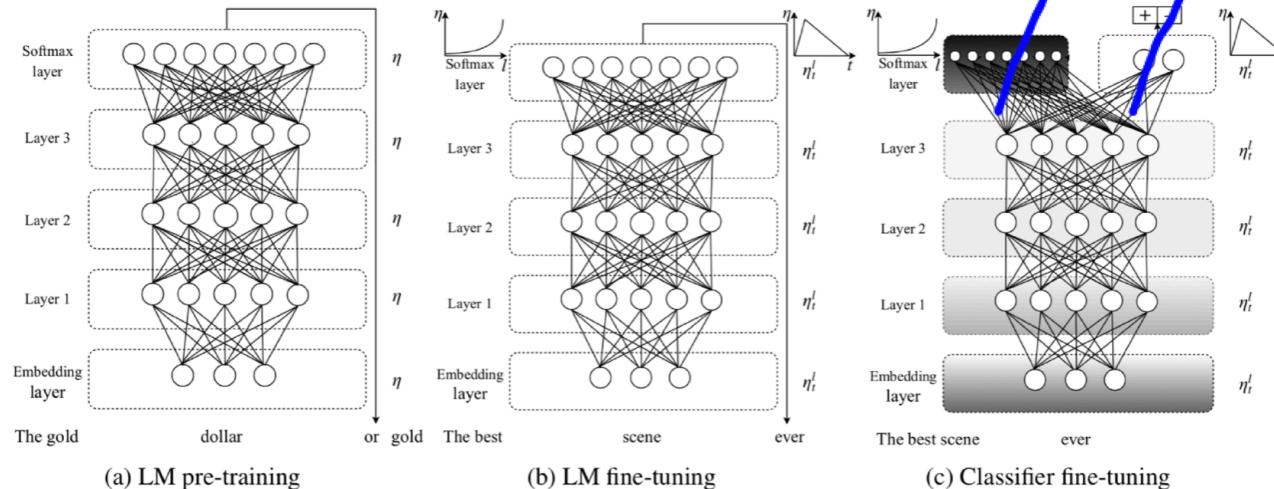
LSTM



فاین تیون مدل زبانی

ULMFit

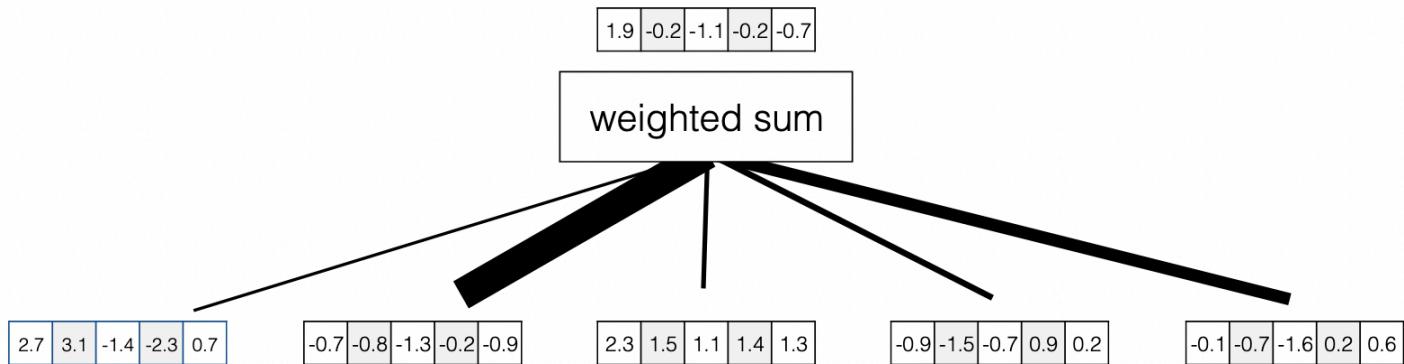
- Train LM on big general domain corpus (use biLM)
- Tune LM on target task data
- Fine-tune as classifier on target task



$$\sigma(wx + b)$$

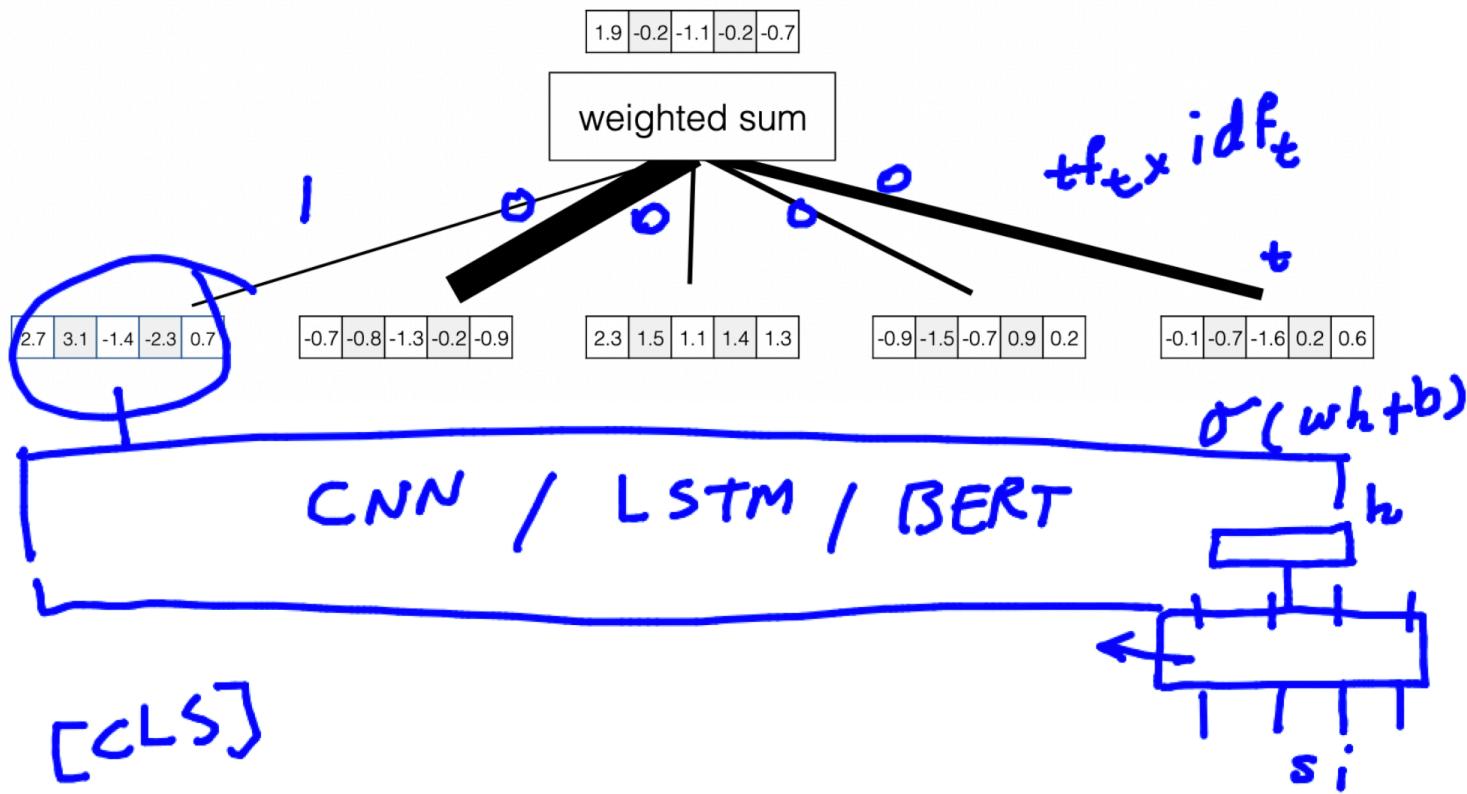
- معرفی طبقه‌بندی
- ارزیابی و یادگیری طبقه‌بندی
- **naive bayes**
- **logistic regression**
- مدل مناسب برای تسک
- شبکه‌های عصبی دیگر

ترکیب خطی امبدینگ‌ها



- معرفی طبقه‌بندی
- ارزیابی و یادگیری طبقه‌بندی
- **naive bayes**
- **logistic regression**
- مدل بازنمایی مناسب برای تسک
- شبکه‌های عصبی دیگر

ترکیب خطی امبدینگ‌ها



معرفی طبقه‌بندی

ارزیابی و یادگیری طبقه‌بندی

مدل naive bayes

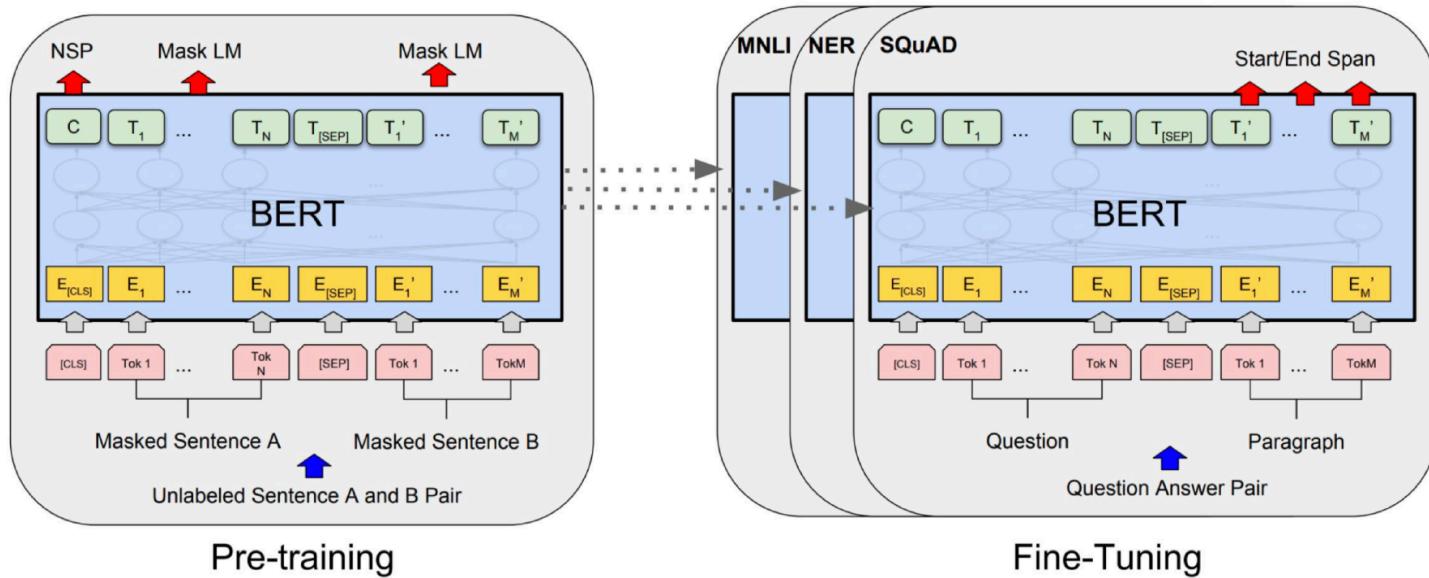
مدل logistic regression

بازنمایی مناسب برای تسک

شبکه‌های عصبی دیگر

- معرفی طبقه‌بندی
- ارزیابی و یادگیری طبقه‌بندی
- **naive bayes**
- **logistic regression**
- مدل
- بازنمایی مناسب برای تسک
- شبکه‌های عصبی دیگر

Transformer-based Classifications



بسم الله الرحمن الرحيم

پردازش زبانهای طبیعی

جلسه ۲۳

احسان الدین عسگری

اردیبهشت ۱۴۰۲

<http://language.ml/>

دانشکده مهندسی کامپیوتر

آزمایشگاه پردازش هوشمند متن و زبان و علوم انسانی محاسباتی

asgari@berkeley.edu



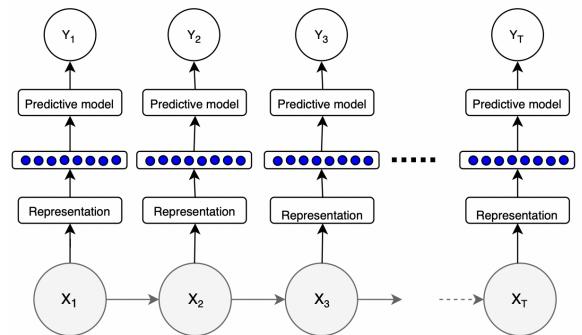
طبقه‌بندی اجزاء رشته

Sequence labeling/ token classification

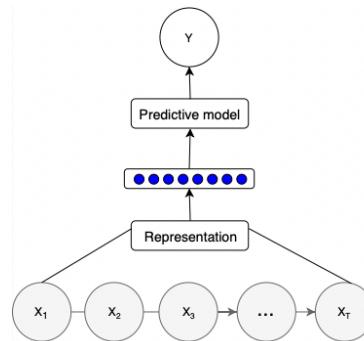
۱) Part-of-speech Tagging



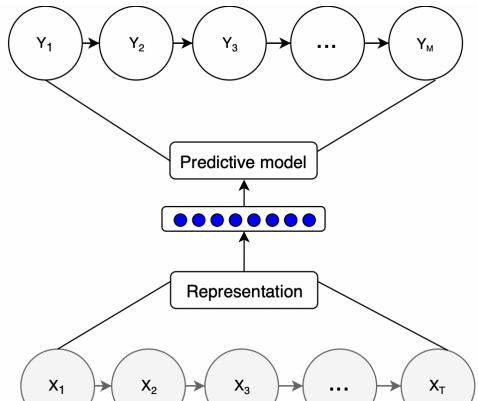
مسائل پردازش زبان



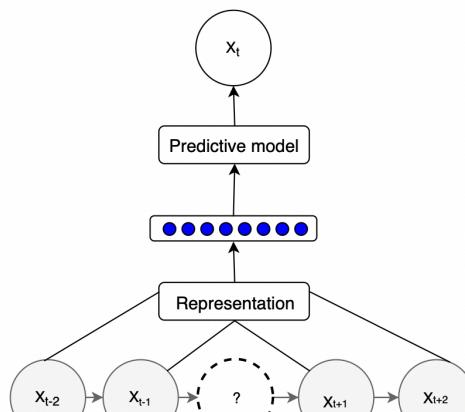
برچسبزنی



طبقه‌بندی



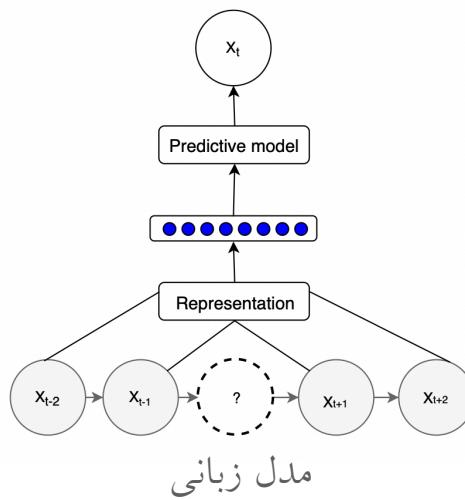
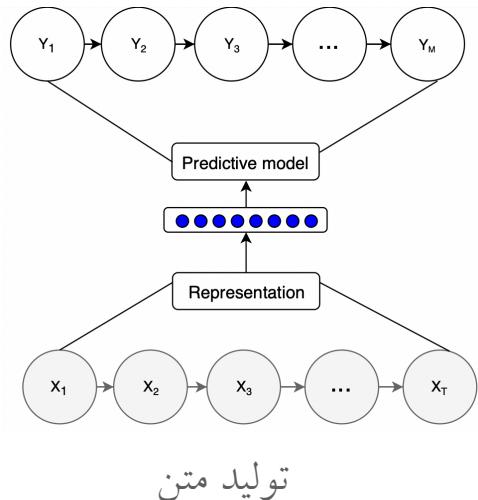
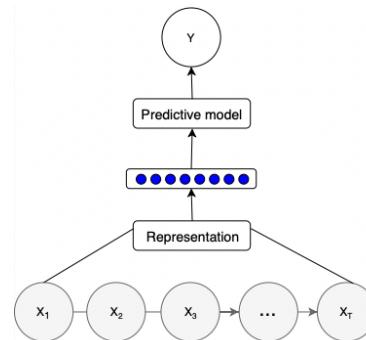
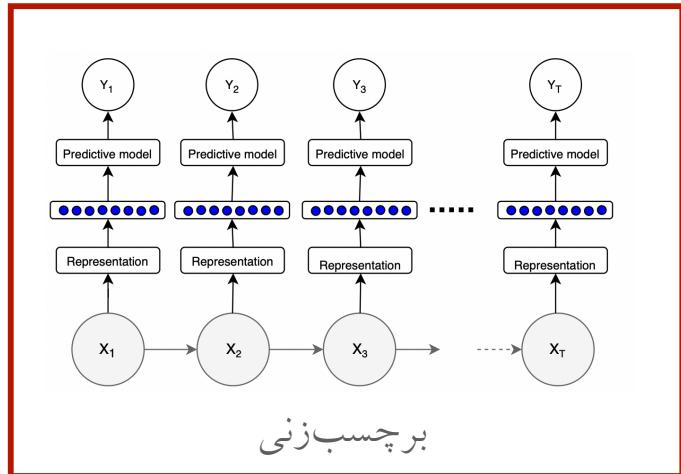
تولید متن



مدل زبانی

- مرور طبقه‌بندی
- طبقه‌بندی توکن
- مثالها
- POS
- NER
- یادگیری
- HMM
- CRF
- Neural
- LM

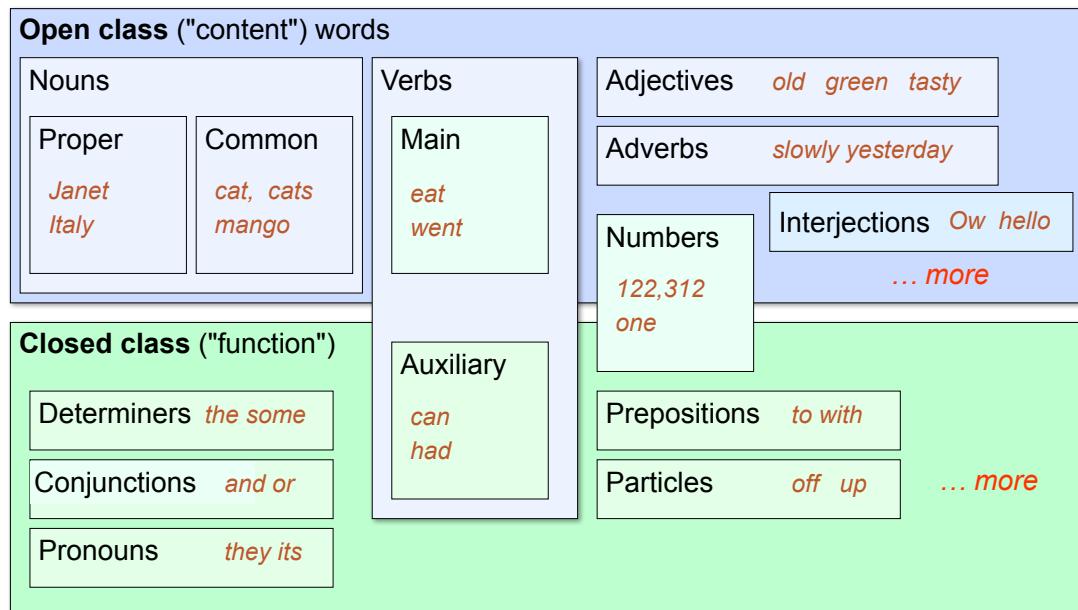
مسائل پردازش زبان



- مرور طبقه‌بندی
- طبقه‌بندی توکن
- مثالها
- POS
- NER
- یادگیری
- HMM
- CRF
- Neural
- LM

برچسبزنی اجزای کلام

- دسته‌بندی کلمات به دسته‌های دستوری.
- کاربرد در مسائل NLP
- ترجمه ماشینی کلاسیک
- تحلیل نظر (مثلا تشخیص صفتها)
- ... تحلیل‌های زبانشناسانه
- بررسی تغیرات زبانی



- مرور طبقه‌بندی
- طبقه‌بندی توکن
- مثال‌ها
- POS
- NER
- یادگیری
- HMM
- CRF
- Neural
- LM

چالش‌ها

- در انگلیسی حدود ۱۵ درصد لغات ابهام دارند.
- خورد و خوراک این دام‌ها خوب است.
- دیروز خودروی علی به جدول خورد.
- داشت وسط خیابون پول خوردهاش رو می‌شمرد..

- مرور طبقه‌بندی
- طبقه‌بندی توکن
- مثال‌ها
- POS
- NER
- یادگیری

- HMM
- CRF

- Neural

- LM

دیتاست‌های فارسی

- دیتاست بیژن‌خان

[/https://dbrg.ut.ac.ir/%D8%A8%DB%8C%DA%98%D9%86%E2%80%8C%D8%AE%D8%A7%D9%86](https://dbrg.ut.ac.ir/%D8%A8%DB%8C%DA%98%D9%86%E2%80%8C%D8%AE%D8%A7%D9%86)

<http://hdl.handle.net/11234/1-3195>

- مرور طبقه‌بندی
- طبقه‌بندی توکن

- مثال‌ها

POS •

NER •

• یادگیری

HMM •

CRF •

Neural •

• **LM**

طبقه‌بندی اجزاء رشته

Sequence labeling/ token classification

۲) Named Entity Recognition

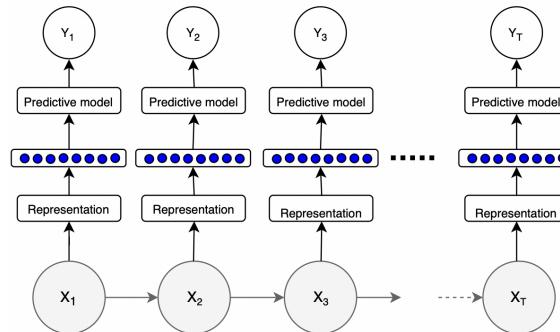


موجودیت‌های نامدار

"Person" شخص): "مولانا امیرالمؤمنین علیہ السلام" PER ◦
"Location" مکان): "حافظیہ شیراز" LOC ◦
"Organization" موسسه): "دانشگاہ صنعتی شریف" ORG ◦
Geo-Political Entity) موقعیت ژئوپولیتیک): "جمهوری اسلامی ایران" GPE ◦

شيخ	B- PER
بهایی	I-PER
معمار	O
و	O
دانشمند	O
برجسته	O
ایران	B-GPE

- معمولاً چند کلمه‌ای است.
- شامل موارد متعدد دیگری نیز می‌شود:
 - زمان، تاریخ، قیمت



- مرور طبقه‌بندی
- طبقه‌بندی توکن
- مثالها
- POS
- NER
- یادگیری
- HMM
- CRF
- Neural
- LM

چالش‌های موجودیت‌های نامدار

- دشواریهای تقطیع
- ابهام موجودیت‌های نامدار
- ایران واکسن بیشتری تولید کرد.
- مسابقات جام ملت‌های آسیا در ایران برگزار خواهد شد.

• مرور طبقه‌بندی
• طبقه‌بندی توکن

• مثالها

POS •

NER •

• یادگیری

HMM •

CRF •

Neural •

LM •

- مرور طبقه‌بندی
- طبقه‌بندی توکن
- مثالها

POS •

NER •

یادگیری •

HMM •

CRF •

Neural •

LM •

دیتاست‌های فارسی

Label	#
Organization	16964
Money	2037
Location	8782
Date	4259
Time	732
Person	7675
Percent	699

<https://hooshvare.github.io/docs/datasets/ner>

• مرور طبقه‌بندی

• طبقه‌بندی توکن

• مثال‌ها

POS •

NER •

• یادگیری

HMM •

CRF •

Neural •

LM •

دیتاست‌های فارسی

پیما (۷۱۴۵ جمله)

آرمان (۷۶۸۲ جمله)

Label	#
Organization	16964
Money	2037
Location	8782
Date	4259
Time	732
Person	7675
Percent	699

Label	#
Organization	30108
Location	12924
Facility	4458
Event	7557
Product	4389
Person	15645

<https://hooshvare.github.io/docs/datasets/ner>

روش‌های باناظریادگیری NER و POS

- Hidden Markov Models
- Conditional Random Fields (CRF)/ Maximum Entropy Markov Models (MEMM)
- Neural sequence models (RNNs or Transformers)
- Large Language Models (like BERT), finetuned

• مرور طبقه‌بندی

• طبقه‌بندی توکن

• مثالها

POS •

NER •

• یادگیری

HMM •

CRF •

Neural •

LM •

مدل کردن رشته زبانی

مدل زبانی مارکف

$$p(\text{start}, w_1, w_2, \dots, w_n, \text{stop}) = \prod_{i=1}^{n+1} \gamma(w_i | w_{i-1})$$

مدل زبانی مبتنی بر کلاس

$$p(\text{start}, w_1, w_2, \dots, w_n, \text{stop}) = \prod_{i=1}^{n+1} \eta(w_i | \text{cl}(w_i)) \times \gamma(\text{cl}(w_i) | \text{cl}(w_{i-1}))$$

<https://dl.acm.org/doi/abs/10.5555/176313.176316>

Hidden Markov Model

$$p(\text{start}, s_1, w_1, s_2, w_2, \dots, s_n, w_n, \text{stop}) = \prod_{i=1}^{n+1} \eta(w_i | s_i) \times \gamma(s_i | s_{i-1})$$

- مرور طبقه‌بندی
- طبقه‌بندی توکن
- مثالها
- POS
- NER
- یادگیری
- HMM
- CRF
- Neural
- LM

- مرور طبقه‌بندی
- طبقه‌بندی توکن
- مثالها

POS •

NER •

یادگیری •

HMM •

CRF •

Neural •

LM •

HMM استقلال‌های

استقلال از حالات قبلی

$$P(Y_i = y_i | Y_{i-1} = y_{i-1}, Y_{i-2} = y_{i-2}, \dots, Y_1 = y_1) = P(Y_i = y_i | Y_{i-1} = y_{i-1})$$

همگونی ترنزیشن

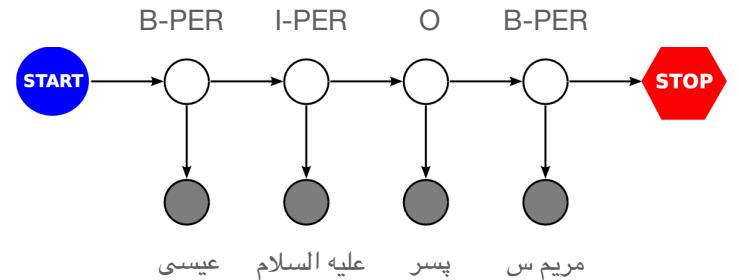
$$P(Y_i = c_k | Y_{i-1} = c_l) = P(Y_t = c_k | Y_{t-1} = c_l)$$

استقلال مشاهده

$$P(X_i = x_i | Y_1 = y_1, \dots, Y_i = y_i, \dots, Y_N = y_N) = P(X_i = x_i | Y_i = y_i)$$

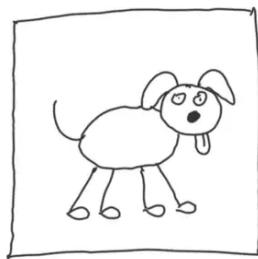
Hidden Markov Model

$$P(X_1 = x_1, \dots, X_N = x_N, Y_1 = y_1, \dots, Y_N = y_N) = P_{\text{init}}(y_1|\text{start}) \times \left(\prod_{i=1}^{N-1} P_{\text{trans}}(y_{i+1}|y_i) \right) \times P_{\text{final}}(\text{stop}|y_N) \times \prod_{i=1}^N P_{\text{emiss}}(x_i|y_i),$$



HMM distributions			
Name	probability distribution	short notation	array size
initial probability	$P(Y_1 = c_k Y_0 = \text{start})$	$P_{\text{init}}(c_k \text{start})$	K
transition probability	$P(Y_i = c_k Y_{i-1} = c_l)$	$P_{\text{trans}}(c_k c_l)$	$K \times K$
final probability	$P(Y_{N+1} = \text{stop} Y_N = c_k)$	$P_{\text{final}}(\text{stop} c_k)$	K
emission probability	$P(X_i = w_j Y_i = c_k)$	$P_{\text{emiss}}(w_j c_k)$	$J \times K$

Notation	
\mathcal{D}_L	training set (including labeled data)
\mathcal{D}_U	training set (unlabeled data only)
M	number of training examples
$x = x_1 \dots x_N$	observation sequence
$y = y_1 \dots y_N$	state sequence
N	length of the sequence
x_i	observation at position i in the sequence, $i \in \{1, \dots, N\}$
y_i	state at position i in the sequence, $i \in \{1, \dots, N\}$
Σ	observation set
J	number of distinct observation labels
w_j	particular observation, $j \in \{1, \dots, J\}$
Λ	state set
K	number of distinct state labels
c_k	particular state, $k \in \{1, \dots, K\}$



MODEL

0.5 → DOG PROBABILITY
0.3 → CAT PROBABILITY
0.2 → PANDA PROBABILITY

TARGET

1
0
0

Loss for class X = $- \underbrace{p(x)}_{\text{probability of class } X \text{ in TARGET}} \cdot \log \underbrace{q(x)}_{\text{probability of class } X \text{ in PREDICTION}}$

Regression

$$MSE = \frac{1}{n} \sum (y - \hat{y})^2$$

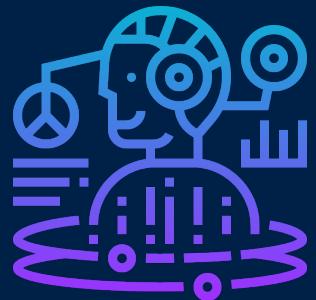
Multi-class

$$L_{\text{cross-entropy}}(\hat{\mathbf{y}}, \mathbf{y}) = - \sum_i y_i \log(\hat{y}_i)$$

Binary / Multi-label

$$\text{Loss} = - \frac{1}{\text{output size}} \sum_{i=1}^{\text{output size}} y_i \cdot \log \hat{y}_i + (1 - y_i) \cdot \log (1 - \hat{y}_i)$$

ماژولهای درس



مسائل کاربردی دیگر



تولید متن



طبقه‌بندی کلمه
و پرسش و پاسخ



طبقه‌بندی متن



مدلهای زبانی



آشنایی با متن و کلمات
روشهای پیش‌پردازش



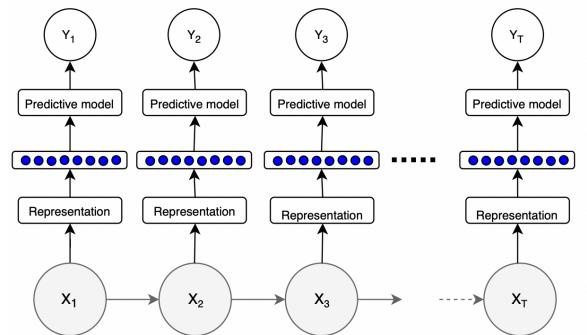
طبقه‌بندی اجزاء رشته

Sequence labeling/ token classification

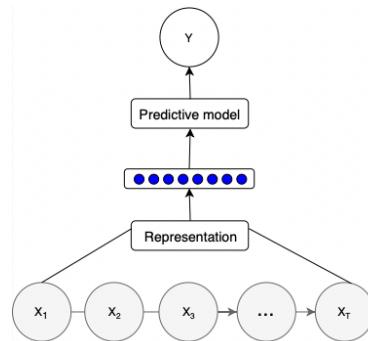
۱) Part-of-speech Tagging



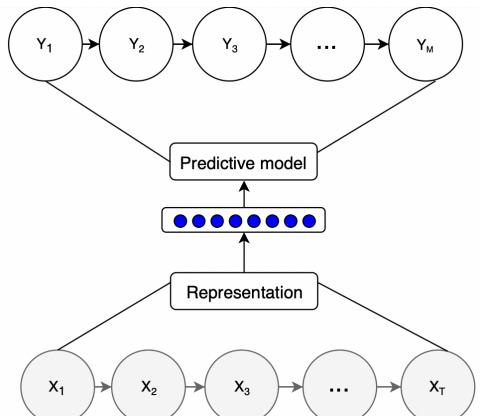
مسائل پردازش زبان



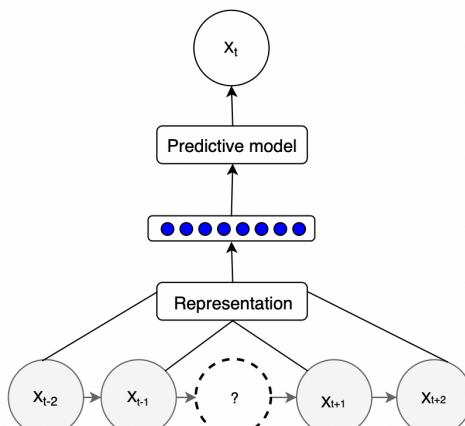
برچسبزنی



طبقه‌بندی



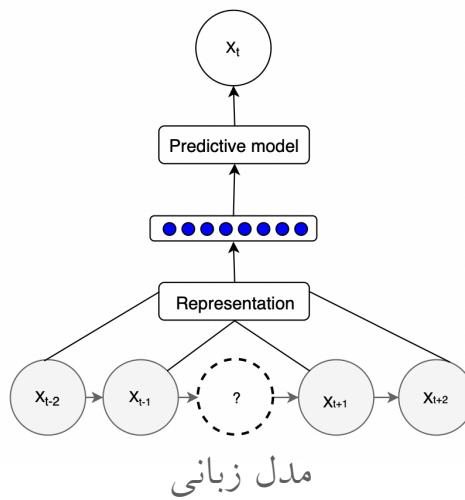
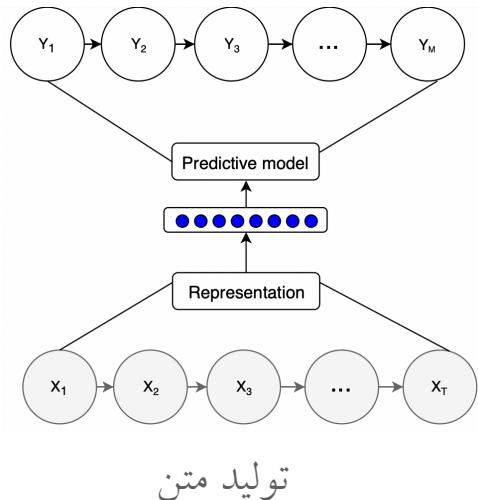
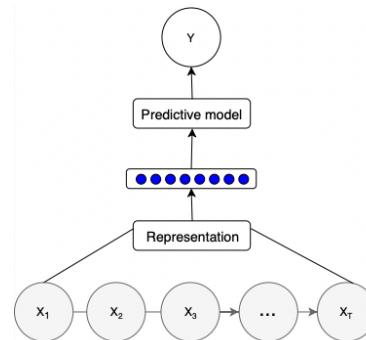
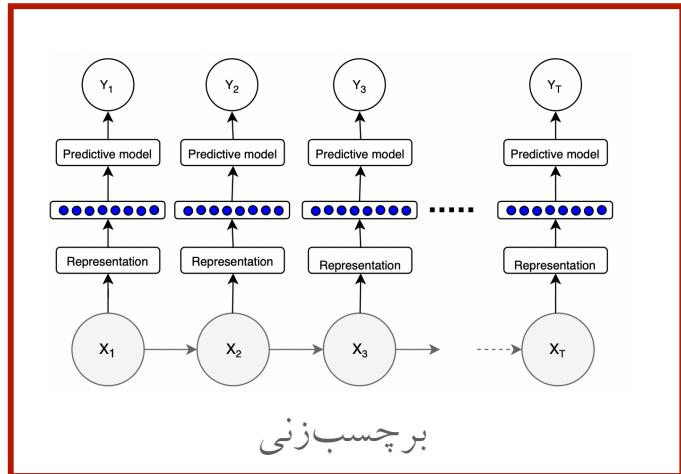
تولید متن



مدل زبانی

- مرور طبقه‌بندی
- طبقه‌بندی توکن
- مثالها
- POS
- NER
- یادگیری
- HMM
- CRF
- Neural
- LM

مسائل پردازش زبان



- مرور طبقه‌بندی
- طبقه‌بندی توکن
- مثالها
- POS
- NER
- یادگیری
- HMM
- CRF
- Neural
- LM

برچسبزنی اجزای کلام

- مرور طبقه‌بندی
- طبقه‌بندی توکن
- مثال‌ها

دسته‌بندی کلمات به دسته‌های دستوری.

تاریخچه: اقلال سده ۴ قبل از میلاد. پانینی زبان‌شناس هندی.

POS

NER

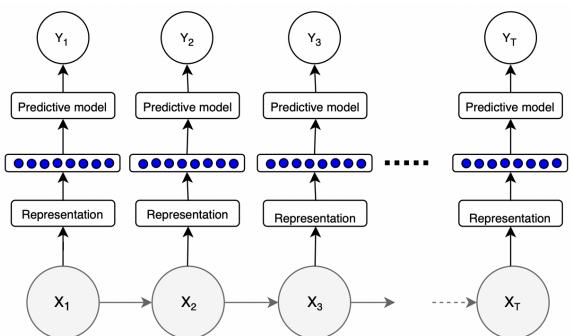
• یادگیری

HMM

CRF

• **Neural**

• **LM**

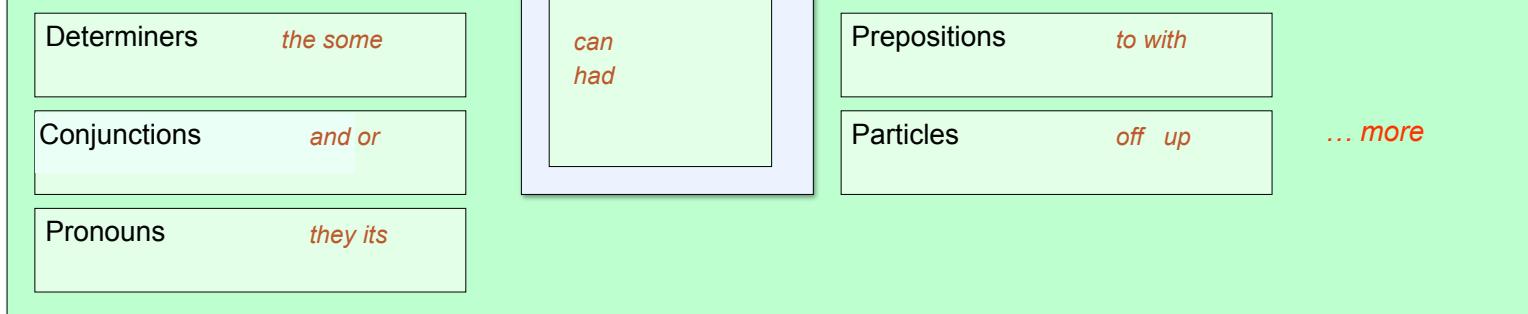


انواع POS تگ‌ها

Open class ("content") words



Closed class ("function")



- ملکه طبقه‌بندی
- طبقه‌بندی توکن
- مثال‌ها
- POS
- NER
- یادگیری
- HMM
- CRF
- Neural
- LM

"universal dependencies" تگهای

	Tag	Description	Example
Open Class	ADJ	Adjective: noun modifiers describing properties	<i>red, young, awesome</i>
	ADV	Adverb: verb modifiers of time, place, manner	<i>very, slowly, home, yesterday</i>
	NOUN	words for persons, places, things, etc.	<i>algorithm, cat, mango, beauty</i>
	VERB	words for actions and processes	<i>draw, provide, go</i>
	PROPN	Proper noun: name of a person, organization, place, etc..	<i>Regina, IBM, Colorado</i>
	INTJ	Interjection: exclamation, greeting, yes/no response, etc.	<i>oh, um, yes, hello</i>
Closed Class Words	ADP	Adposition (Preposition/Postposition): marks a noun's spacial, temporal, or other relation	<i>in, on, by under</i>
	AUX	Auxiliary: helping verb marking tense, aspect, mood, etc.,	<i>can, may, should, are</i>
	CCONJ	Coordinating Conjunction: joins two phrases/clauses	<i>and, or, but</i>
	DET	Determiner: marks noun phrase properties	<i>a, an, the, this</i>
	NUM	Numeral	<i>one, two, first, second</i>
	PART	Particle: a preposition-like form used together with a verb	<i>up, down, on, off, in, out, at, by</i>
	PRON	Pronoun: a shorthand for referring to an entity or event	<i>she, who, I, others</i>
Other	SCONJ	Subordinating Conjunction: joins a main clause with a subordinate clause such as a sentential complement	<i>that, which</i>
	PUNCT	Punctuation	<i>; , ()</i>
	SYM	Symbols like \$ or emoji	<i>\$, %</i>
	X	Other	<i>asdf, qwfg</i>

• مرور طبقه‌بندی

• طبقه‌بندی توکن

• مثال‌ها

• POS

• NER

• یادگیری

• HMM

• CRF

• Neural

• LM

چرایی پرداختن به POS-tag‌ها

- کاربرد در مسائل NLP
 - ترجمه ماشینی کلاسیک
 - تحلیل نظر (مثلا تشخیص صفتها)
- تحلیل‌های زبانشناسانه
 - بررسی تغییرات زبانی

- مرور طبقه‌بندی
- طبقه‌بندی توکن
- مثالها
- POS
- NER
- یادگیری
- HMM
- CRF
- Neural
- LM

چالش‌ها

- در انگلیسی حدود ۱۵ درصد لغات ابهام دارند.
- خورد و خوراک این دام‌ها خوب است.
- دیروز خودروی علی به جدول خورد.
- داشت وسط خیابون پول خوردهاش رو می‌شمرد..

- مرور طبقه‌بندی
- طبقه‌بندی توکن
- مثال‌ها
- POS
- NER
- یادگیری

- HMM
- CRF

- Neural

- LM

دیتاست‌های فارسی

- دیتاست بیژن‌خان

[/https://dbrg.ut.ac.ir/%D8%A8%DB%8C%DA%98%D9%86%E2%80%8C%D8%AE%D8%A7%D9%86](https://dbrg.ut.ac.ir/%D8%A8%DB%8C%DA%98%D9%86%E2%80%8C%D8%AE%D8%A7%D9%86)

<http://hdl.handle.net/11234/1-3195>

- مرور طبقه‌بندی
- طبقه‌بندی توکن

- مثال‌ها

POS •

NER •

• یادگیری

HMM •

CRF •

Neural •

• **LM**

طبقه‌بندی اجزاء رشته

Sequence labeling/ token classification

۲) Named Entity Recognition



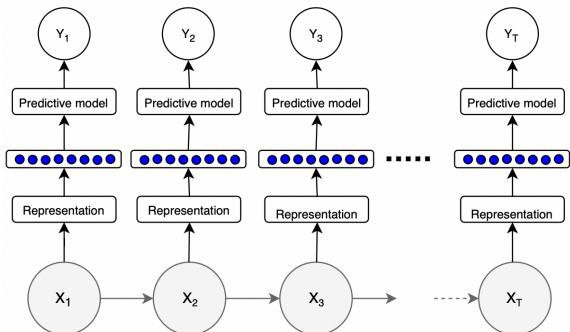
موجودیت‌های نامدار

- مرور طبقه‌بندی
- طبقه‌بندی توکن
- مثال‌ها
- POS
- NER
- یادگیری
- HMM
- CRF
- Neural
- LM

◦ "مولانا امیرالمؤمنین علیہ السلام" Person) PER ◦
◦ "حافظیه شیراز" Location) LOC ◦
◦ "دانشگاه صنعتی شریف" Organization) ORG ◦
◦ "جمهوری اسلامی ایران" Geo-Political Entity) GPE ◦

◦ معمولاً چند کلمه‌ای است.

◦ شامل موارد متعدد دیگری نیز می‌شود:
◦ زمان، تاریخ، قیمت



چالش‌های موجودیت‌های نامدار

- دشواریهای تقطیع
- ابهام موجودیت‌های نامدار
- ایران واکسن بیشتری تولید کرد.
- مسابقات جام ملت‌های آسیا در ایران برگزار خواهد شد.

◦ مرور طبقه‌بندی
◦ طبقه‌بندی توکن

◦ مثالها

◦ POS

◦ NER

◦ یادگیری

◦ HMM

◦ CRF

◦ Neural

◦ LM

تگ‌های BIO

شیخ	B- PER
بهایی	I-PER
معمار	O
و	O
دانشمند	O
برجسته	O
ایران	B-GPE
در	O
...	...

- مرور طبقه‌بندی
- طبقه‌بندی توکن
- مثالها
- POS •
- NER •
- یادگیری
- HMM •
- CRF •
- Neural •
- LM •

دیتاست‌های فارسی

- مرور طبقه‌بندی
- طبقه‌بندی توکن
- مثالها

POS •

NER •

یادگیری •

HMM •

CRF •

Neural •

LM •

Label	#
Organization	16964
Money	2037
Location	8782
Date	4259
Time	732
Person	7675
Percent	699

دیتاست‌های فارسی

پیما (۷۱۴۵ جمله)

Label	#
Organization	16964
Money	2037
Location	8782
Date	4259
Time	732
Person	7675
Percent	699

آرمان (۷۶۸۲ جمله)

Label	#
Organization	30108
Location	12924
Facility	4458
Event	7557
Product	4389
Person	15645

<https://hooshvare.github.io/docs/datasets/ner>

• مرور طبقه‌بندی

• طبقه‌بندی توکن

• مثال‌ها

• POS

• NER

• یادگیری

• HMM

• CRF

• Neural

• LM

مدل کردن رشته زبانی

مدل زبانی مارکف

$$p(\text{start}, w_1, w_2, \dots, w_n, \text{stop}) = \prod_{i=1}^{n+1} \gamma(w_i | w_{i-1})$$

مدل زبانی مبتنی بر کلاس

$$p(\text{start}, w_1, w_2, \dots, w_n, \text{stop}) = \prod_{i=1}^{n+1} \eta(w_i | \text{cl}(w_i)) \times \gamma(\text{cl}(w_i) | \text{cl}(w_{i-1}))$$

<https://dl.acm.org/doi/abs/10.5555/176313.176316>

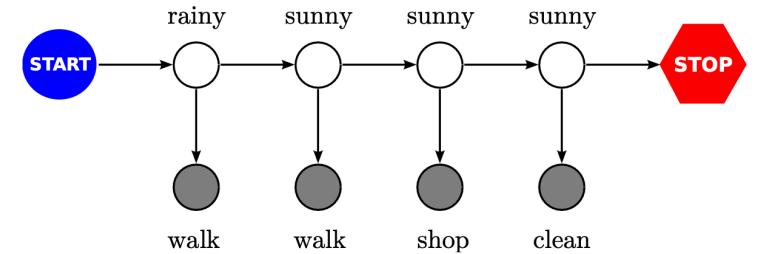
Hidden Markov Model

$$p(\text{start}, s_1, w_1, s_2, w_2, \dots, s_n, w_n, \text{stop}) = \prod_{i=1}^{n+1} \eta(w_i | s_i) \times \gamma(s_i | s_{i-1})$$

- مرور طبقه‌بندی
- طبقه‌بندی توکن
- مثالها
- POS
- NER
- یادگیری
- HMM
- CRF
- Neural
- LM

Hidden Markov Model

$$P(X_1 = x_1, \dots, X_N = x_N, Y_1 = y_1, \dots, Y_N = y_N) = P_{\text{init}}(y_1|\text{start}) \times \left(\prod_{i=1}^{N-1} P_{\text{trans}}(y_{i+1}|y_i) \right) \times P_{\text{final}}(\text{stop}|y_N) \times \prod_{i=1}^N P_{\text{emiss}}(x_i|y_i),$$



HMM distributions			
Name	probability distribution	short notation	array size
initial probability	$P(Y_1 = c_k Y_0 = \text{start})$	$P_{\text{init}}(c_k \text{start})$	K
transition probability	$P(Y_i = c_k Y_{i-1} = c_l)$	$P_{\text{trans}}(c_k c_l)$	$K \times K$
final probability	$P(Y_{N+1} = \text{stop} Y_N = c_k)$	$P_{\text{final}}(\text{stop} c_k)$	K
emission probability	$P(X_i = w_j Y_i = c_k)$	$P_{\text{emiss}}(w_j c_k)$	$J \times K$

Notation	
\mathcal{D}_L	training set (including labeled data)
\mathcal{D}_U	training set (unlabeled data only)
M	number of training examples
$x = x_1 \dots x_N$	observation sequence
$y = y_1 \dots y_N$	state sequence
N	length of the sequence
x_i	observation at position i in the sequence, $i \in \{1, \dots, N\}$
y_i	state at position i in the sequence, $i \in \{1, \dots, N\}$
Σ	observation set
J	number of distinct observation labels
w_j	particular observation, $j \in \{1, \dots, J\}$
Λ	state set
K	number of distinct state labels
c_k	particular state, $k \in \{1, \dots, K\}$

maximum likelihood تخمین

$$P(X_1 = x_1, \dots, X_N = x_N, Y_1 = y_1, \dots, Y_N = y_N) = \\ P_{\text{init}}(y_1|\text{start}) \times \left(\prod_{i=1}^{N-1} P_{\text{trans}}(y_{i+1}|y_i) \right) \times P_{\text{final}}(\text{stop}|y_N) \times \prod_{i=1}^N P_{\text{emiss}}(x_i|y_i),$$

$$\arg \max_{\theta} \sum_{m=1}^M \log P_{\theta}(X = x^m, Y = y^m),$$

Initial Counts: $C_{\text{init}}(c_k) = \sum_{m=1}^M \mathbf{1}(y_1^m = c_k);$

Transition Counts: $C_{\text{trans}}(c_k, c_l) = \sum_{m=1}^M \sum_{i=2}^N \mathbf{1}(y_i^m = c_k \wedge y_{i-1}^m = c_l);$

Final Counts: $C_{\text{final}}(c_k) = \sum_{m=1}^M \mathbf{1}(y_N^m = c_k);$

Emission Counts: $C_{\text{emiss}}(w_j, c_k) = \sum_{m=1}^M \sum_{i=1}^N \mathbf{1}(x_i^m = w_j \wedge y_i^m = c_k);$

$$P_{\text{init}}(c_k|\text{start}) = \frac{C_{\text{init}}(c_k)}{\sum_{l=1}^K C_{\text{init}}(c_l)}$$

$$P_{\text{final}}(\text{stop}|c_l) = \frac{C_{\text{final}}(c_l)}{\sum_{k=1}^K C_{\text{trans}}(c_k, c_l) + C_{\text{final}}(c_l)}$$

$$P_{\text{trans}}(c_k|c_l) = \frac{C_{\text{trans}}(c_k, c_l)}{\sum_{p=1}^K C_{\text{trans}}(c_p, c_l) + C_{\text{final}}(c_l)}$$

$$P_{\text{emiss}}(w_j|c_k) = \frac{C_{\text{emiss}}(w_j, c_k)}{\sum_{q=1}^J C_{\text{emiss}}(w_q, c_k)}$$

بسم الله الرحمن الرحيم

پردازش زبانهای طبیعی

جلسه ۲۱

احسان الدین عسگری

خرداد ۱۴۰۳

<http://language.ml/>

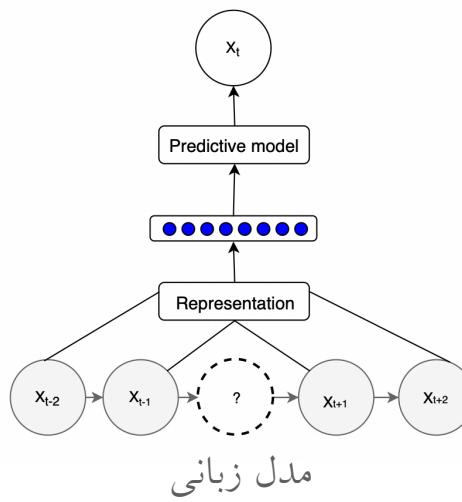
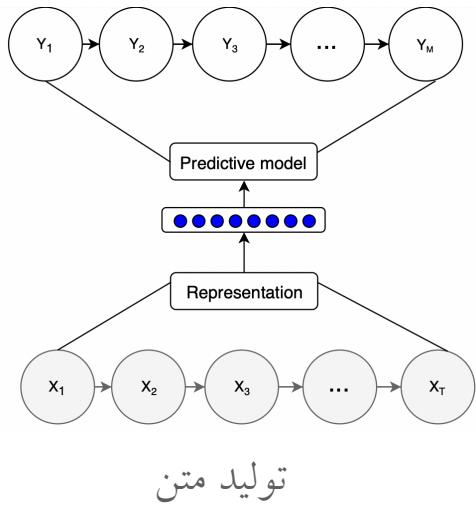
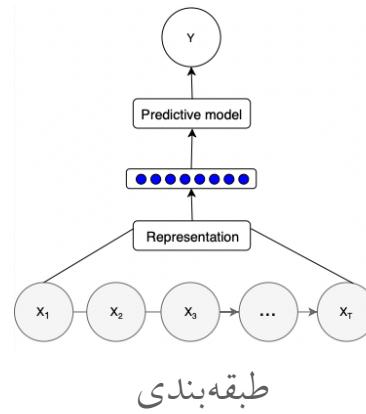
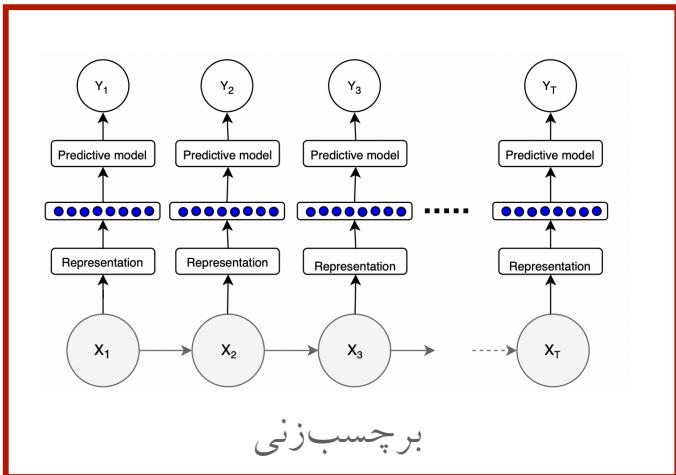
دانشکده مهندسی کامپیوتر

آزمایشگاه پردازش هوشمند متن و زبان و علوم انسانی محاسباتی

asgari@berkeley.edu



مسائل پردازش زبان



مدل کردن رشته زبانی

مدل زبانی مارکف

$$p(\text{start}, w_1, w_2, \dots, w_n, \text{stop}) = \prod_{i=1}^{n+1} \gamma(w_i | w_{i-1})$$

مدل زبانی مبتنی بر کلاس

$$p(\text{start}, w_1, w_2, \dots, w_n, \text{stop}) = \prod_{i=1}^{n+1} \eta(w_i | \text{cl}(w_i)) \times \gamma(\text{cl}(w_i) | \text{cl}(w_{i-1}))$$

<https://dl.acm.org/doi/abs/10.5555/176313.176316>

Hidden Markov Model

$$p(\text{start}, s_1, w_1, s_2, w_2, \dots, s_n, w_n, \text{stop}) = \prod_{i=1}^{n+1} \eta(w_i | s_i) \times \gamma(s_i | s_{i-1})$$



maximum likelihood تخمین

$$P(X_1 = x_1, \dots, X_N = x_N, Y_1 = y_1, \dots, Y_N = y_N) = \\ P_{\text{init}}(y_1|\text{start}) \times \left(\prod_{i=1}^{N-1} P_{\text{trans}}(y_{i+1}|y_i) \right) \times P_{\text{final}}(\text{stop}|y_N) \times \prod_{i=1}^N P_{\text{emiss}}(x_i|y_i),$$

$$\arg \max_{\theta} \sum_{m=1}^M \log P_{\theta}(X = x^m, Y = y^m),$$

Initial Counts: $C_{\text{init}}(c_k) = \sum_{m=1}^M \mathbf{1}(y_1^m = c_k);$

Transition Counts: $C_{\text{trans}}(c_k, c_l) = \sum_{m=1}^M \sum_{i=2}^N \mathbf{1}(y_i^m = c_k \wedge y_{i-1}^m = c_l);$

Final Counts: $C_{\text{final}}(c_k) = \sum_{m=1}^M \mathbf{1}(y_N^m = c_k);$

Emission Counts: $C_{\text{emiss}}(w_j, c_k) = \sum_{m=1}^M \sum_{i=1}^N \mathbf{1}(x_i^m = w_j \wedge y_i^m = c_k);$

$$P_{\text{init}}(c_k|\text{start}) = \frac{C_{\text{init}}(c_k)}{\sum_{l=1}^K C_{\text{init}}(c_l)}$$

$$P_{\text{final}}(\text{stop}|c_l) = \frac{C_{\text{final}}(c_l)}{\sum_{k=1}^K C_{\text{trans}}(c_k, c_l) + C_{\text{final}}(c_l)}$$

$$P_{\text{trans}}(c_k|c_l) = \frac{C_{\text{trans}}(c_k, c_l)}{\sum_{p=1}^K C_{\text{trans}}(c_p, c_l) + C_{\text{final}}(c_l)}$$

$$P_{\text{emiss}}(w_j|c_k) = \frac{C_{\text{emiss}}(w_j, c_k)}{\sum_{q=1}^J C_{\text{emiss}}(w_q, c_k)}$$

دیکد کردن برای برچسبهای نهان!

- طبقه‌بندی توکن

- مثالها

- POS

- NER

- یادگیری

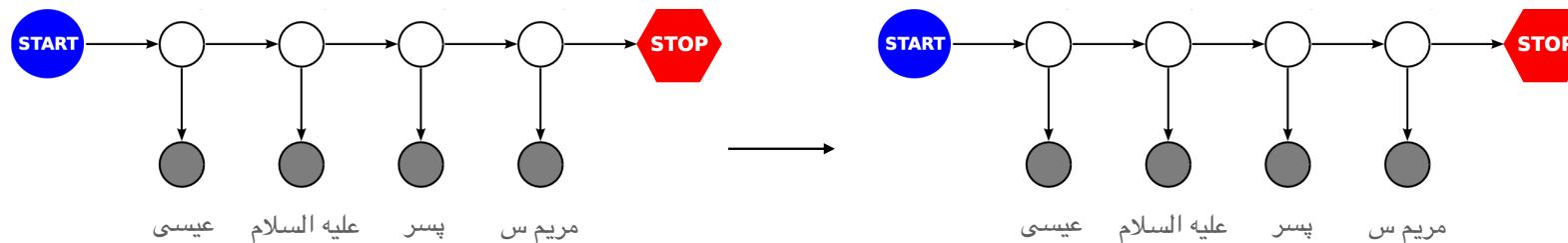
- HMM

- Neural

$x = x_1 \dots x_N$ ورودی

$y^* = y_1^* \dots y_N^*$ خروجی

$$\begin{aligned} P_{\text{init}}(c_k | \text{start}) &= \frac{C_{\text{init}}(c_k)}{\sum_{l=1}^K C_{\text{init}}(c_l)} \\ P_{\text{final}}(\text{stop} | c_l) &= \frac{C_{\text{final}}(c_l)}{\sum_{k=1}^K C_{\text{trans}}(c_k, c_l) + C_{\text{final}}(c_l)} \\ P_{\text{trans}}(c_k | c_l) &= \frac{C_{\text{trans}}(c_k, c_l)}{\sum_{p=1}^K C_{\text{trans}}(c_p, c_l) + C_{\text{final}}(c_l)} \\ P_{\text{emiss}}(w_j | c_k) &= \frac{C_{\text{emiss}}(w_j, c_k)}{\sum_{q=1}^J C_{\text{emiss}}(w_q, c_k)} \end{aligned}$$



دیکد کردن (یافتن **hidden state** ها)

- مرور طبقه‌بندی
- طبقه‌بندی توکن
- مثالها

Posterior decoding

$$y_i^* = \arg \max_{y_i \in \Lambda} P(Y_i = y_i | X_1 = x_1, \dots, X_N = x_N).$$

Viterbi

- POS
- NER
- یادگیری
- HMM

$$\begin{aligned} y^* &= \arg \max_{y=y_1 \dots y_N} P(Y_1 = y_1, \dots, Y_N = y_N | X_1 = x_1, \dots, X_N = x_N) \\ &= \arg \max_{y=y_1 \dots y_N} P(Y_1 = y_1, \dots, Y_N = y_N, X_1 = x_1, \dots, X_N = x_N). \end{aligned}$$

• طبقه‌بندی توکن

• مثالها

POS •

NER •

• یادگیری

HMM •

Neural •

Viterbi الگوریتم

$$\begin{aligned}y^* &= \arg \max_{y=y_1 \dots y_N} P(Y_1 = y_1, \dots, Y_N = y_N | X_1 = x_1, \dots, X_N = x_N) \\&= \arg \max_{y=y_1 \dots y_N} P(Y_1 = y_1, \dots, Y_N = y_N, X_1 = x_1, \dots, X_N = x_N).\end{aligned}$$

Viterbi $\text{viterbi}(i, y_i) = \max_{y_1 \dots y_{i-1}} P(Y_1 = y_1, \dots, Y_i = y_i, X_1 = x_1, \dots, X_i = x_i)$

الgoritم Viterbi

$$\begin{aligned}y^* &= \arg \max_{y=y_1 \dots y_N} P(Y_1 = y_1, \dots, Y_N = y_N | X_1 = x_1, \dots, X_N = x_N) \\&= \arg \max_{y=y_1 \dots y_N} P(Y_1 = y_1, \dots, Y_N = y_N, X_1 = x_1, \dots, X_N = x_N).\end{aligned}$$

Viterbi $\text{viterbi}(i, y_i) = \max_{y_1 \dots y_{i-1}} P(Y_1 = y_1, \dots, Y_i = y_i, X_1 = x_1, \dots, X_i = x_i)$

	N	V	Adj
<شرع>			
حال			
اميد			
خوب			
است			
<بيان>			

Viterbi الگوریتم

	N	V	Adj
است			
امید			
بد			
داشتن			
حالت			
حال			
خوب			
<مشروع>			
<بيان>			

	N	V	Adj
N			
V			
Adj			

	N	V	Adj
<مشروع>			
حال			
امید			
خوب			
است			
<بيان>			

• طبقه‌بندی توکن

• مثالها

• POS

• NER

• یادگیری

• HMM

• Neural

Viterbi الگوريتم

	N	V	Adj
حال			
امید			
خوب			
است			

$$\begin{aligned}
 \text{viterbi}(1, c_k) &= P_{\text{init}}(c_k | \text{start}) \times P_{\text{emiss}}(x_1 | c_k) \\
 \text{viterbi}(i, c_k) &= \left(\max_{c_l \in \Lambda} P_{\text{trans}}(c_k | c_l) \times \text{viterbi}(i-1, c_l) \right) \times P_{\text{emiss}}(x_i | c_k) \\
 \text{backtrack}(i, c_k) &= \left(\arg \max_{c_l \in \Lambda} P_{\text{trans}}(c_k | c_l) \times \text{viterbi}(i-1, c_l) \right) \\
 \text{viterbi}(N+1, \text{stop}) &= \max_{c_l \in \Lambda} P_{\text{final}}(\text{stop} | c_l) \times \text{viterbi}(N, c_l) \\
 \text{backtrack}(N+1, \text{stop}) &= \arg \max_{c_l \in \Lambda} P_{\text{final}}(\text{stop} | c_l) \times \text{viterbi}(N, c_l).
 \end{aligned}$$

• طبقه‌بندی توکن
• مثالها

POS •

NER •

یادگیری •

HMM •

Neural •

• طبقه‌بندی توکن

• مثالها

POS •

NER •

یادگیری •

HMM •

Neural •

Viterbi الگوریتم

```
1: input: sequence  $x_1, \dots, x_N$ , scores  $P_{\text{init}}, P_{\text{trans}}, P_{\text{final}}, P_{\text{emiss}}$ 
2: Forward pass: Compute the best paths for every end state
3: Initialization
4: for  $c_k \in \Lambda$  do
5:    $\text{viterbi}(1, c_k) = P_{\text{init}}(c_k | \text{start}) \times P_{\text{emiss}}(x_1 | c_k)$ 
6: end for
7: for  $i = 2$  to  $N$  do
8:   for  $c_k \in \Lambda$  do
9:      $\text{viterbi}(i, c_k) = \left( \max_{c_l \in \Lambda} P_{\text{trans}}(c_k | c_l) \times \text{viterbi}(i - 1, c_l) \right) \times P_{\text{emiss}}(x_i | c_k)$ 
10:     $\text{backtrack}(i, c_k) = \left( \arg \max_{c_l \in \Lambda} P_{\text{trans}}(c_k | c_l) \times \text{viterbi}(i - 1, c_l) \right)$ 
11:   end for
12: end for
13:  $\max_{y \in \Lambda^N} P(X = x, Y = y) := \max_{c_l \in \Lambda} P_{\text{final}}(\text{stop} | c_l) \times \text{viterbi}(N, c_l)$ 
14:
15: Backward pass: backtrack to obtain the most likely path
16:  $\hat{y}_N = \arg \max_{c_l \in \Lambda} P_{\text{final}}(\text{stop} | c_l) \times \text{viterbi}(N, c_l)$ 
17: for  $i = N - 1$  to  $1$  do
18:    $\hat{y}_i = \text{backtrack}(i + 1, \hat{y}_{i+1})$ 
19: end for
20: output: the viterbi path  $\hat{y}$ .
```

تفاوت‌های Viterbi و Posterior decoding

$$\begin{aligned}y^* &= \arg \max_{y=y_1 \dots y_N} P(Y_1 = y_1, \dots, Y_N = y_N | X_1 = x_1, \dots, X_N = x_N) \\&= \arg \max_{y=y_1 \dots y_N} P(Y_1 = y_1, \dots, Y_N = y_N, X_1 = x_1, \dots, X_N = x_N).\end{aligned}$$

$$y_i^* = \arg \max_{y_i \in \Lambda} P(Y_i = y_i | X = x)$$

• طبقه‌بندی توکن

• مثالها

• POS

• NER

• یادگیری

• HMM

• Neural

Posterior decoding الگوریتم

• طبقه بندی توکن

مثالها

POS .

NER •

یادگیری

HMM •

Neural •

$$\begin{aligned}
P(X = x) &= \sum_{c_k \in \Lambda} P(X_1 = x_1, \dots, X_N = x_N, Y_i = c_k) \\
&= \sum_{c_k \in \Lambda} \underbrace{P(X_1 = x_1, \dots, X_i = x_i, Y_i = c_k)}_{\text{forward}(i, c_k)} \times \underbrace{P(X_{i+1} = x_{i+1}, \dots, X_N = x_N | Y_i = c_k)}_{\text{backward}(i, c_k)} \\
&= \sum_{c_k \in \Lambda} \text{forward}(i, c_k) \times \text{backward}(i, c_k).
\end{aligned}$$

• طبقه‌بندی توکن

• مثالها

POS •

NER •

• یادگیری

HMM •

Neural •

الگوریتم – Posterior decoding فوروارد

Forward Probability: $\text{forward}(i, c_k) = P(Y_i = c_k, X_1 = x_1, \dots, X_i = x_i)$

$$\begin{aligned} P(y_i, x_{1:i}) &= \sum_{y_{i-1} \in \Lambda} P(y_i, y_{i-1}, x_{1:i}) = \sum_{y_{i-1} \in \Lambda} P(x_i | y_i, y_{i-1}, x_{1:i-1}) \cdot P(y_i | y_{i-1}, x_{1:i-1}) \cdot P(y_{i-1}, x_{1:i-1}) \\ &= \sum_{y_{i-1} \in \Lambda} P(x_i | y_i) \cdot P(y_i | y_{i-1}) \cdot \text{forward}(i-1, y_{i-1}) \end{aligned}$$

$$\begin{aligned} \text{forward}(1, c_k) &= P_{\text{init}}(c_k | \text{start}) \times P_{\text{emiss}}(x_1 | c_k) \\ \text{forward}(i, c_k) &= \left(\sum_{c_l \in \Lambda} P_{\text{trans}}(c_k | c_l) \times \text{forward}(i-1, c_l) \right) \times P_{\text{emiss}}(x_i | c_k) \\ \text{forward}(N+1, \text{stop}) &= \sum_{c_l \in \Lambda} P_{\text{final}}(\text{stop} | c_l) \times \text{forward}(N, c_l). \end{aligned}$$

• طبقه‌بندی توکن

• مثالها

• POS

• NER

• یادگیری

• HMM

• Neural

الگوریتم - بکوارد **Posterior decoding**

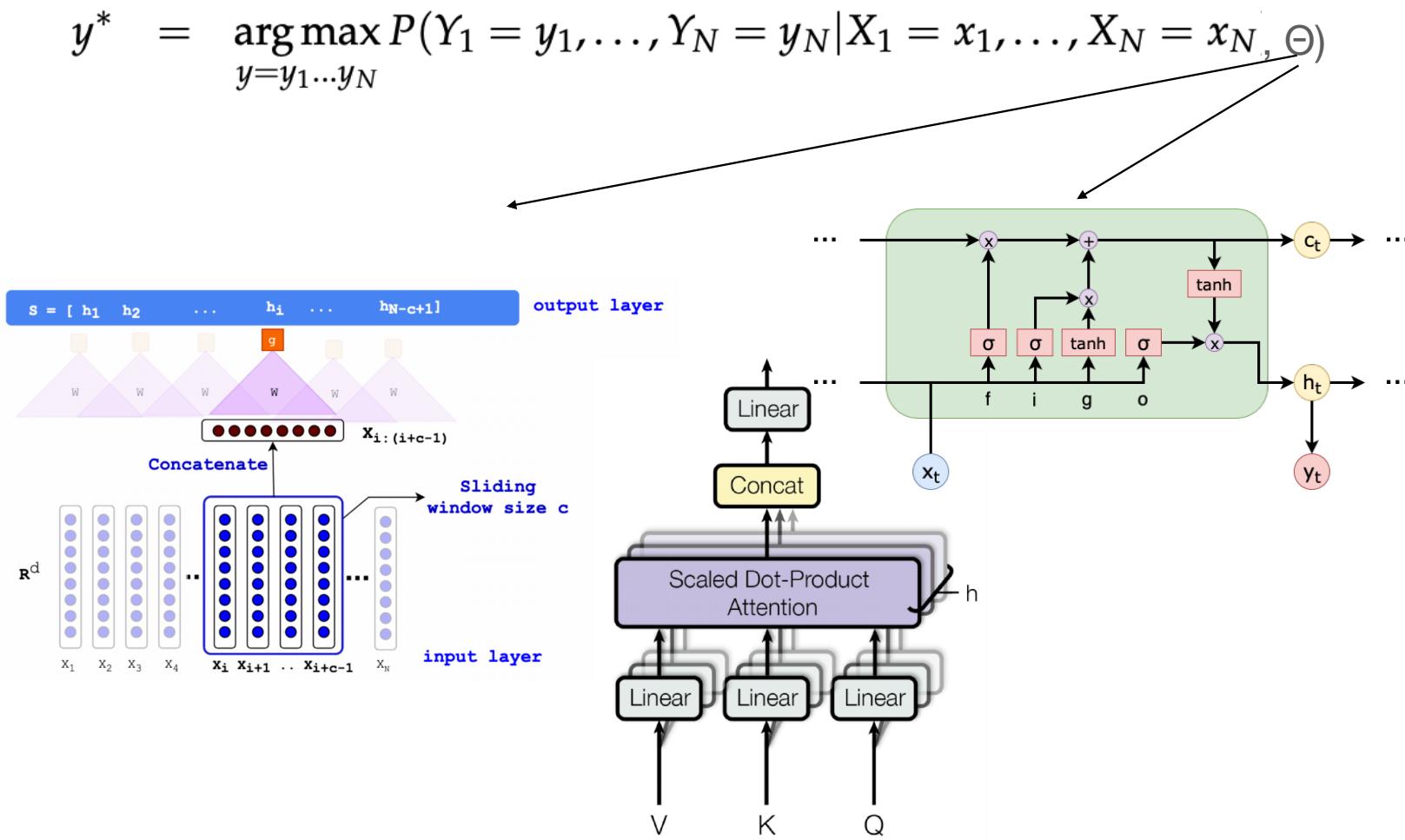
Backward Probability: $\text{backward}(i, c_l) = P(X_{i+1} = x_{i+1}, \dots, X_N = x_N | Y_i = c_l)$

$$P(x_{i+1:N}|y_i) = \sum_{y_{i+1} \in \Lambda} P(x_{i+1:N}, y_{i+1}|y_i) = \sum_{y_{i+1} \in \Lambda} P(x_{i+2:N}|y_i, y_{i+1}, x_{i+1})P(x_{i+1}, |y_{i+1}, y_i)P(y_{i+1}|y_i)$$

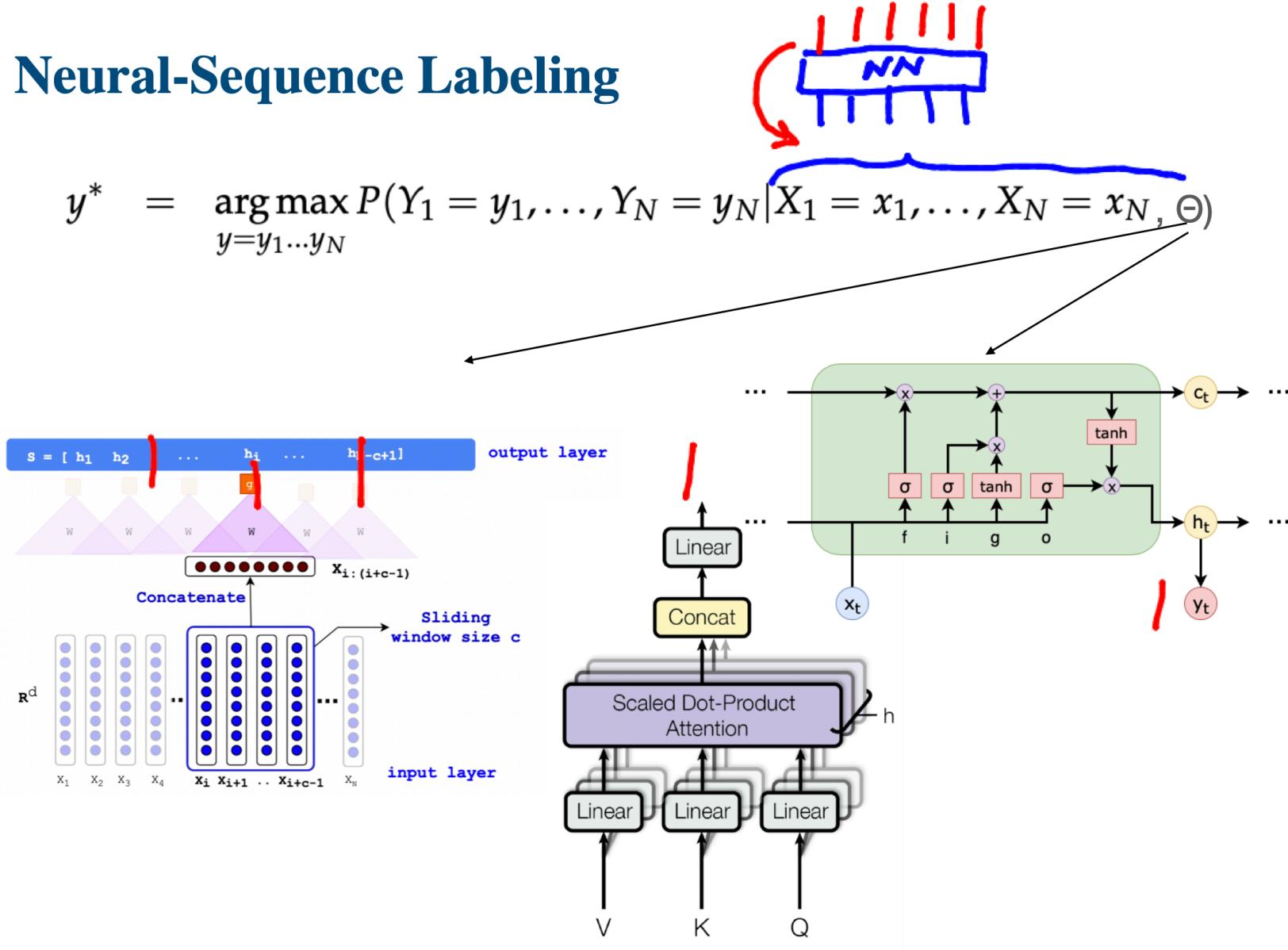
$$\begin{aligned}\text{backward}(N, c_l) &= P_{\text{final}}(\text{stop}|c_l) \\ \text{backward}(i, c_l) &= \sum_{c_k \in \Lambda} P_{\text{trans}}(c_k|c_l) \times \text{backward}(i+1, c_k) \times P_{\text{emiss}}(x_{i+1}|c_k) \\ \text{backward}(0, \text{start}) &= \sum_{c_k \in \Lambda} P_{\text{init}}(c_k|\text{start}) \times \text{backward}(1, c_k) \times P_{\text{emiss}}(x_1|c_k).\end{aligned}$$

- طبقه‌بندی توکن
- مثالها
- POS
- NER
- یادگیری
- HMM
- Neural

Neural-Sequence Labeling



Neural-Sequence Labeling



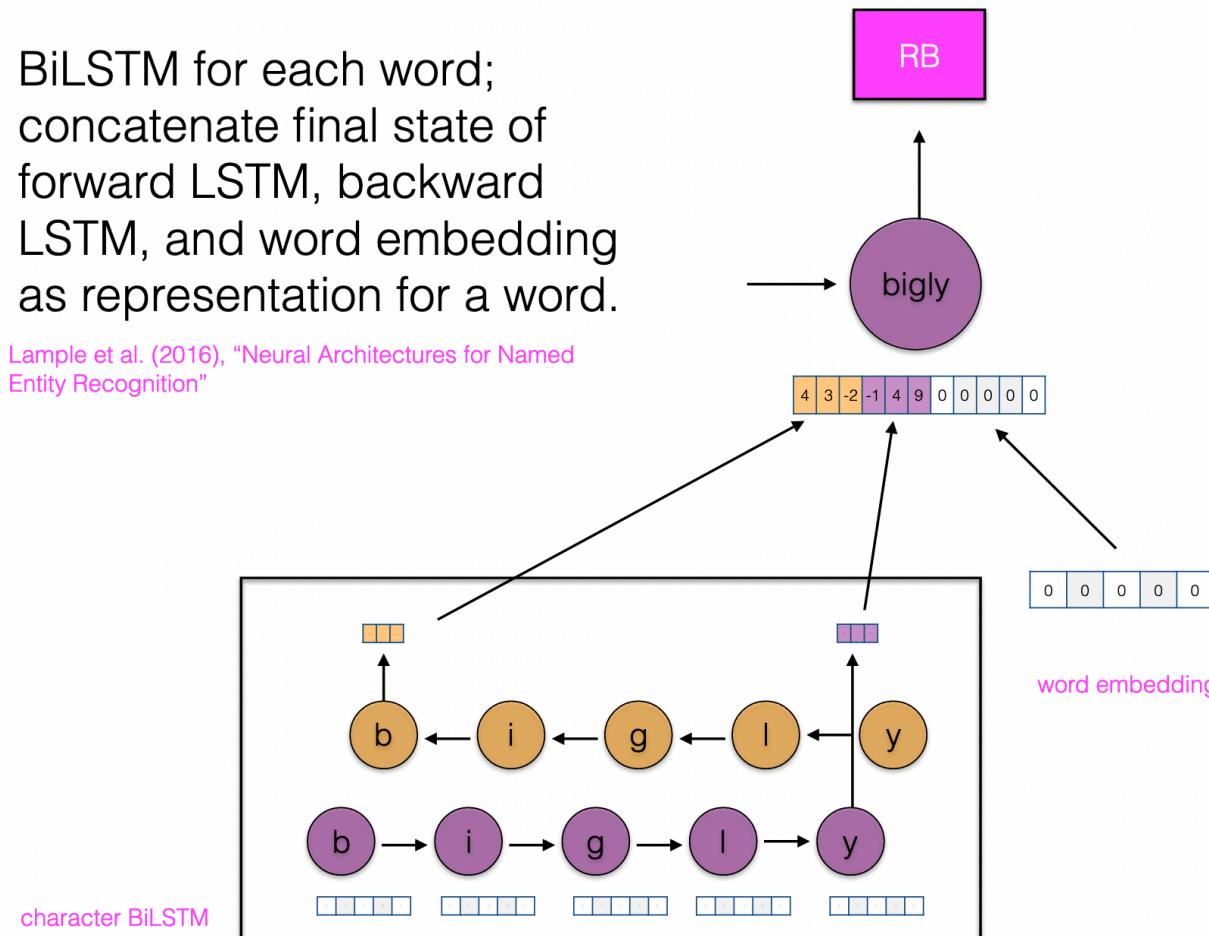
- طبقه‌بندی توکن
- مثالها
- POS
- NER
- یادگیری
- HMM
- Neural

- مرور طبقه‌بندی
- طبقه‌بندی توکن
- مثالها
- POS
- NER
- یادگیری
- HMM
- MEMM
- CRF
- Neural
- LM

Neural-Sequence Labeling

BiLSTM for each word;
concatenate final state of
forward LSTM, backward
LSTM, and word embedding
as representation for a word.

Lample et al. (2016), "Neural Architectures for Named Entity Recognition"

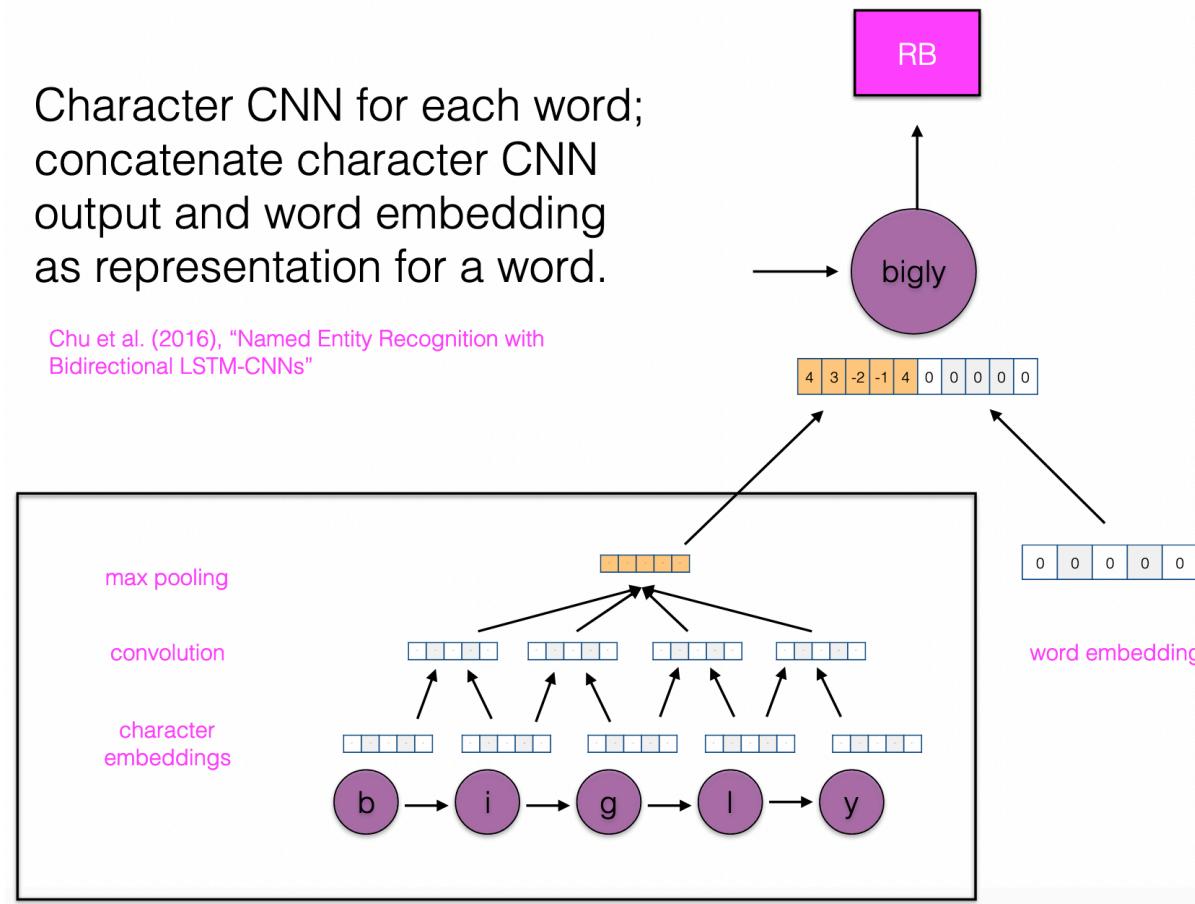


- مرور طبقه‌بندی
- طبقه‌بندی توکن
- مثالها
- POS
- NER
- یادگیری
- HMM
- MEMM
- CRF
- Neural
- LM

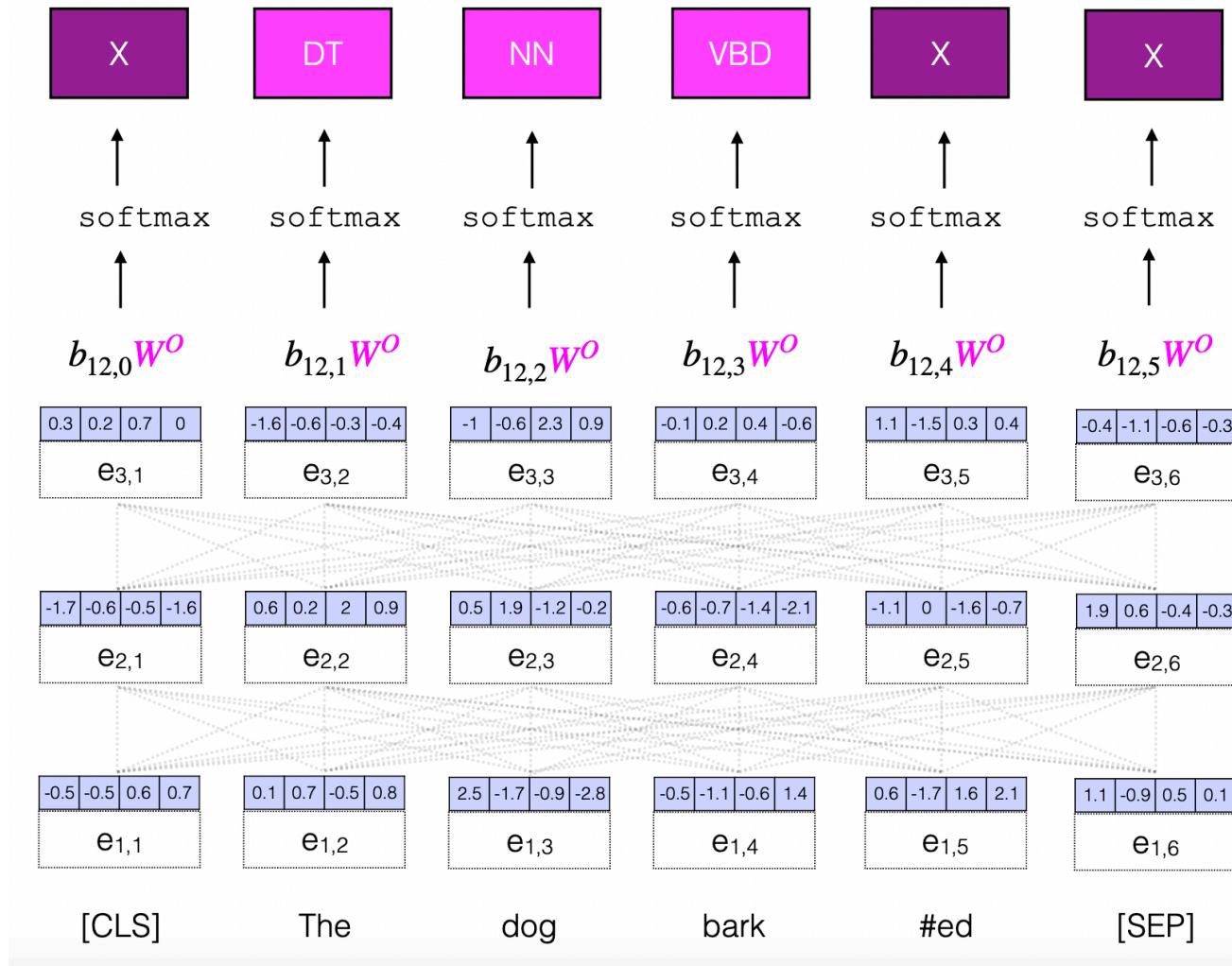
Neural-Sequence Labeling

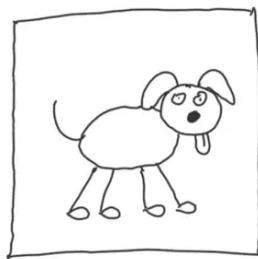
Character CNN for each word;
concatenate character CNN
output and word embedding
as representation for a word.

Chu et al. (2016), "Named Entity Recognition with Bidirectional LSTM-CNNs"



مدل زبانی و طبقه‌بندی توکن





MODEL

0.5 → DOG PROBABILITY
0.3 → CAT PROBABILITY
0.2 → PANDA PROBABILITY

TARGET

1
0
0

Loss for class X = $- \underbrace{p(x)}_{\text{probability of class } X \text{ in TARGET}} \cdot \log \underbrace{q(x)}_{\text{probability of class } X \text{ in PREDICTION}}$

Regression

$$MSE = \frac{1}{n} \sum (y - \hat{y})^2$$

Multi-class

$$L_{\text{cross-entropy}}(\hat{\mathbf{y}}, \mathbf{y}) = - \sum_i y_i \log(\hat{y}_i)$$

Binary / Multi-label

$$\text{Loss} = -\frac{1}{\text{output size}} \sum_{i=1}^{\text{output size}} y_i \cdot \log \hat{y}_i + (1 - y_i) \cdot \log (1 - \hat{y}_i)$$

- مرور طبقه‌بندی
- طبقه‌بندی توکن
- مثالها

POS

NER

یادگیری

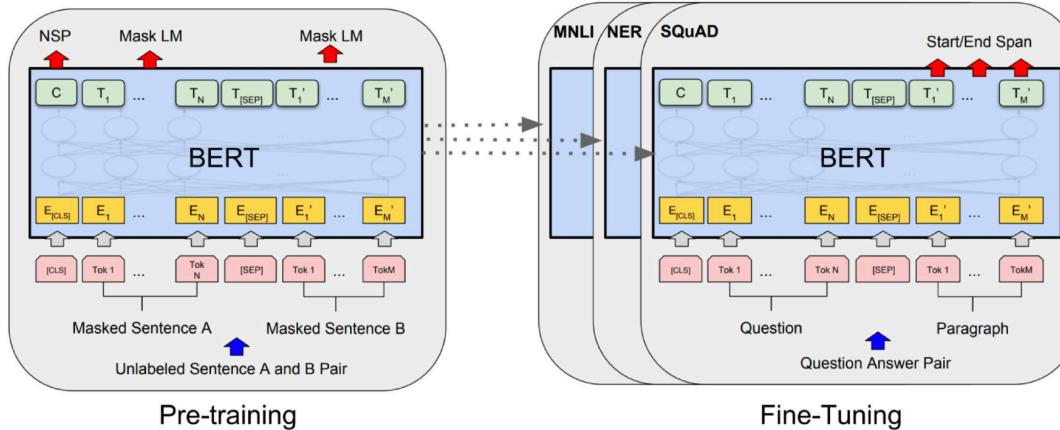
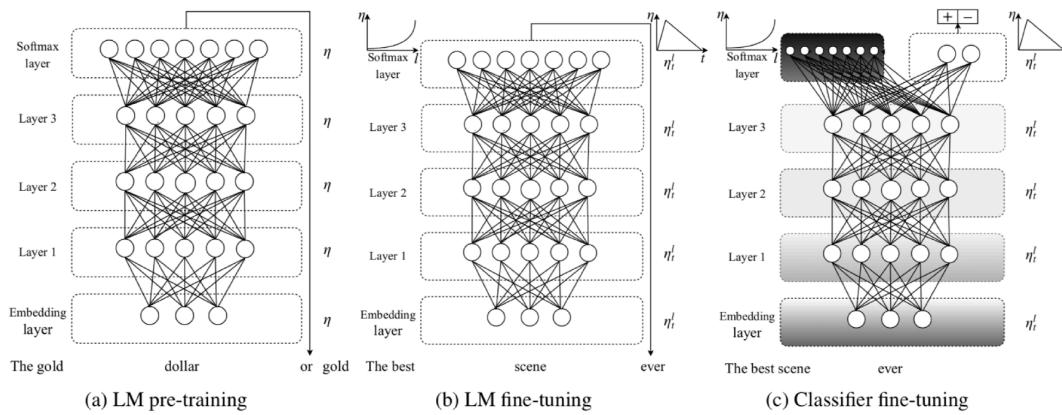
HMM

CRF

Neural

LM

Language-model based Labeling



NLP Progress

[View on GitHub](#) 

NLP-progress

Repository to track the progress in Natural Language Processing (NLP), including the datasets and the current state-of-the-art for the most common NLP tasks.

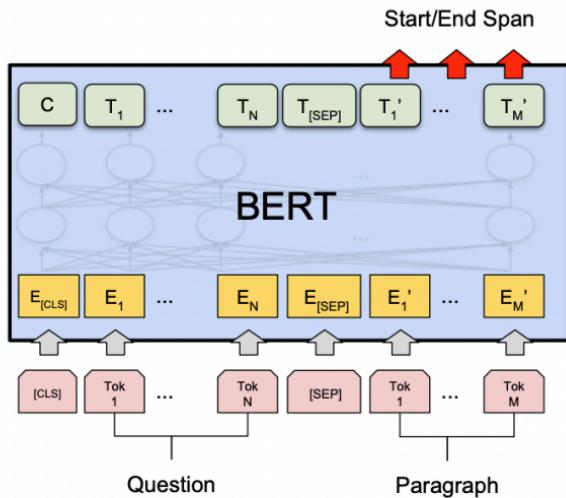
Tracking Progress in Natural Language Processing

Table of contents

English

- [Automatic speech recognition](#)
- [CCG](#)
- [Common sense](#)
- [Constituency parsing](#)
- [Coreference resolution](#)
- [Data-to-Text Generation](#)
- [Dependency parsing](#)
- [Dialogue](#)
- [Domain adaptation](#)
- [Entity linking](#)

سامانه‌های پرسش و پاسخ استخراجی



$$\mathcal{L} = -\log p_{\text{start}}(s^*) - \log p_{\text{end}}(e^*)$$

$$p_{\text{start}}(i) = \text{softmax}_i(\mathbf{w}_{\text{start}}^\top \mathbf{h}_i)$$

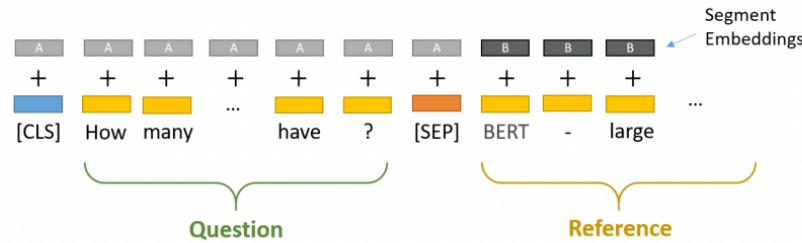
$$p_{\text{end}}(i) = \text{softmax}_i(\mathbf{w}_{\text{end}}^\top \mathbf{h}_i)$$

where \mathbf{h}_i is the hidden vector of C_i , returned by BERT

Question = Segment A

Passage = Segment B

Answer = predicting two endpoints in segment B



Question: How many parameters does BERT-large have?

Reference Text: BERT-large is really big... it has 24 layers and an embedding size of 1,024, for a total of 340M parameters! Altogether it is 1.34GB, so expect it to take a couple minutes to download to your Colab instance.

Image credit: <https://mccormickml.com/>

Question Answering – Dataset

SQuAD2.0
The Stanford Question Answering Dataset

Normans
The Stanford Question Answering Dataset

The Normans (Norman: Normands; French: Normands; Latin: Normanni) were the people who in the **10th and 11th centuries** gave their name to **Normandy**, a region in France. They were descended from Norse ("Norman" comes from "Norseman") raiders and pirates from Denmark, Iceland and Norway who, under their leader Rollo, agreed to swear fealty to King Charles III of West Francia. Through generations of assimilation and mixing with the native Frankish and Roman-Gaulish populations, their descendants would gradually merge with the Carolingian-based cultures of West Francia. The distinct cultural and ethnic identity of the **Normans** emerged initially in the first half of the 10th century, and it continued to evolve over the succeeding centuries.

In what country is Normandy located?
Ground Truth Answers: France France France France

When were the Normans in Normandy?
Ground Truth Answers: **10th and 11th centuries** in the 10th and 11th centuries 10th and 11th centuries 10th and 11th centuries

From which countries did the Norse originate?
Ground Truth Answers: Denmark, Iceland and Norway Denmark, Iceland and Norway Denmark, Iceland and Norway Denmark, Iceland and Norway

Who was the Norse leader?
Ground Truth Answers: Rollo Rollo Rollo Rollo

What century did the Normans first gain their separate identity?
Ground Truth Answers: 10th century the first half of the 10th century 10th 10th

Who gave their name to Normandy in the 1000's and 1100's
Ground Truth Answers: <No Answer>

Stanford Question Answering Dataset (SQuAD) is a reading comprehension dataset, consisting of questions posed by crowdworkers on a set of Wikipedia articles, where the answer to every question is a segment of text, or *span*, from the corresponding reading passage, or the question might be unanswerable.

Dataset v2: 100,000 questions in SQuAD1.1 with over 50,000 unanswerable

Metrics: Exact Match, F1

Pranav Rajpurkar, Robin Jia, and Percy Liang.
[Know What You Don't Know: Unanswerable Questions for SQuAD](#).
In Proceedings of the [ACL 2018](#).

Persian Question Answering – Dataset

ParSQuAD: Persian Question Answering Dataset based on Machine Translation of SQuAD 2.0

Negin Abadani^a, Jamshid Mozafari^a, Afsaneh Fatemi^{*a}, Mohammadali Nematbakhsh^a, Arefeh Kazemi^b

^aDepartment of Software Engineering, University of Isfahan, Isfahan, Iran

^bDepartment of Linguistics, University of Isfahan, Isfahan, Iran

{negin.abadani, mozafari.jamshid}@gmail.com, {a_fatemi@, nematbakhsh}@eng.ui.ac.ir, arefeh_kazemi@yahoo.com

Received: 2021/07/02

Revised: 2021/10/16

Accepted: 2021/11/01

Abstract— Recent developments in Question Answering (QA) have improved state-of-the-art results, and various datasets have been released for this task. Since substantial English training datasets are available for this task, the majority of works published are for English Question Answering. However, due to the lack of Persian datasets, less research has been done on the latter language, making comparisons difficult. This paper introduces the Persian Question Answering Dataset (ParSQuAD) based on the machine translation of the SQuAD 2.0 dataset. Many errors have been discovered within the process of translating the dataset; therefore, two versions of ParSQuAD have been generated depending on whether these errors have been corrected manually or automatically. As a result, the first large-scale QA training resource for Persian has been generated. In addition, we trained three baseline models, i.e., BERT, ALBERT, and Multilingual-BERT (mBERT), on both versions of ParSQuAD. mBERT achieves scores of 56.66% and 52.86% for F1 score and exact match ratio respectively on the test set with the first version and scores of 70.84% and 67.73% respectively with the second version. This model obtained the best results out of the three on each version of ParSQuAD.

Keywords—Question Answering; Persian Machine Reading Comprehension; Persian Question Answering Dataset; SQuAD

Reading Comprehension, the need to generate datasets has increased.

Various large-scale QA datasets have been released recently, including CNN/Daily Mail [6], MS MARCO [7], RACE [8], and SQuAD [9]. However, the majority of these datasets are designed for English QA, and there are fewer or no datasets available for other languages, such as Persian. In other words, to this date, no similar open-domain dataset has been generated for Persian QA.

Among these recently released English datasets, the Stanford Question Answering Dataset (SQuAD) [9] has been used in most recent QA works.

This dataset comes in two different versions and contains (c, q, a) triplets representing a context paragraph from Wikipedia articles, a question posed by crowdworkers, and the related answer(s). The answer is a segment of the corresponding passage; therefore, a number also comes with the answer, indicating the answer's start position in the context paragraph. This dataset is divided into training and development sets, each having 80% and 10% of the total instances, respectively.

6219v1 [cs.CL] 13 Feb 2022

PQuAD: A Persian Question Answering Dataset

Kasra Darvishi, Newsha Shahbodagh, Zahra Abbasiantaeb, Saeedeh Momtazi*

Department of Computer Engineering
Amirkabir University of Technology (Tehran Polytechnic)

Abstract

We present Persian Question Answering Dataset (PQuAD), a crowd-sourced reading comprehension dataset on Persian Wikipedia articles. It includes 80,000 questions along with their answers, with 25% of the questions being adversarially unanswerable. We examine various properties of the dataset to show the diversity and the level of its difficulty as a MRC benchmark. By releasing this dataset, we aim to ease research on Persian reading comprehension and development of persian question answering systems. Our experiments on different state-of-the-art pre-trained contextualized language models shows 74.8% Exact Match (EM) and 87.6% F1-score that can be used as the baseline results for further research on Persian QA.

Keywords: Machine Reading Comprehension - Natural Language Processing - Persian Dataset - Question Answering

Natural Language Inference

- Natural language inference datasets
 - logical relationship between a hypothesis and a premise.
 - a pair of sentences as input and classify their relationship labels from **entailment**, **contradiction**, and **neutral**

Dataset Name	SNLI (Stanford Natural Language Inference)
Task	Natural Language Inference (NLI)
Size	570,000 pairs
Metric	Accuracy
Example	Premise: "A man inspects the uniform of a figure in some East Asian country." Hypothesis: "The man is sleeping." Label: Contradiction
Reference	Bowman et al., 2015

Fine-tuning paradigm..-

