Foundations of statistical NLP ← chapter 1

speech and language processing (کتاب)

Lingusitics knowledge ← دانش زبان‌شناسی ← ۶ سطح مختلف

1) phonetics and phonology

دانش آوا شناسی ← هر کلمه چه بیانی وجود دارد.

↳ به پردازش وارد نمی‌شویم ← به پردازش گفته می‌شود

↳ زمانی استفاده می‌شود که خروجی سیستم گفتاری است

2) Morphology ← به ساخت / داخلی کلمه می‌پردازیم ← واحد کلمه

که از چه زیربخشی تشکیل شده است. منظور ساز

نمی‌شناسیم ← روی شخص مفرد ← خود فعل

استمراری

واحد جمله ← اجزای جمله با هم در ارتباط هستند

3) syntax ← Morphology یک سطح بالاتر از

I / saw / the / man

4) semantics ← معناشناسی ← بر اساس معنای واژه تقسیم می‌شویم.

↳ ارتباط بین واژه‌ها ←

is a ← Hyponomy

part of ← Meronymy

مترادف ← synoymy

متضاد ← Antonymy

۵) pragmatic

یک سطح دانش را بالاتر می‌بریم ← کلی‌تر می‌باشد دارد

← در اشتراک قبلی مدل‌سازی پیچیده است.

دانش از کل نشدن

آیا می‌دونی ساعت چند است؟

← جواب مورد انتظار بله و خیر نیست.

← اما شخص سعی می‌کند که منظور جمله را بفهمد

۶) Discourse

از سطح جمله بالاتر می‌رویم.

← ارتباط بین جملات نیاز داریم.

← ضمایر مراجع آنها

NLP challenges

① paraphrasing

چند کلمه با یک معنی

کلمات و عبارات که معنی یکسانی را بیان می‌کنند

یک مفهوم را می‌توان با بیان‌های مختلف بیان کنیم.

Fall, autmn

② Ambiguity

یک کلمه با چند معنی

Fall / افتادن / پاییز

یا برای یک جمله ممکنه دو تا تعبیر وجود داشته باشه.

The door is open → یا بسته

یا Fact

جاها ایجاد می‌شود؟ از سطح phonetic شروع می‌کنیم

از نگارش به متن

ماد کم مستعمل رسم

ماد

Morphology — در سطح

unionize → { union + ize / un + ion + ize } — یونیزه / نیونیزه

Syntax — در سطح

I saw the man with a telescope — تلسکوپ داری یا از دربین مرد با تلسکوپ به دست دیدی

star → ستاره — semantic — در سطح

→ celebrity

pragmatic — در سطح

would you please give me the result?

Can you give me the salt → Do you have the ability?

در سطح discourse ← ضمیر به ۲ تا چیز مختلف می‌تونه برگرده

duck → اردک / شنا کردن آب در ماشین / سرتکون دادن سر

اگر فی‌جمله دارای ابهام تلقی می‌شود ← به خاطر این هستش که تک تک کلمات آن دارای ابهام هستند

همین شعرهای بی‌سر و ته رو مشاهده می‌کنیم

قرار یک واقعیت که در توزیع احتمالی در متن وجود دارد بیان است.

Zipf's law

که مختص همه جا است نه فقط متن و زبان شناسی

① استخراج کلمات و در آوردن تعداد هر کدام و freq

$\frac{\text{تعداد کلمه مورد نظر}}{\text{کل}} \times 100$

② مرتب کردن بر اساس تعداد بدست آمده که بعضی معنی خاصی مستقل نکند

(The / of / and / to)

(اضافه / بود / است)

③ برای هر کلمه rank داریم

که رتبه کلمه بر اساس شمار

با ضرب کردن rank × میاد (freq)

که به یک عدد نسبتاً ثابتی رسیم.

چون کلمه دوم ×۲ می‌شود ← یعنی مثلاً کلمه دوم نصف کلمه اول است

چون کلمه سوم ×۳ می‌شود. کلمه سوم نصف کلمه دوم است.

که ولی به یک عدد می‌رسیم ← نشان می‌دهد بسیاری کلمه سوم نصف کلمه دوم است

در انتهای جدول که عدد ثابت شود و به واریانس داشته باشیم ← در ابتدای لیست بیشتر دیده می‌شود و هرچی پایین‌تر می‌رویم ← واریانس کمتر می‌شود

Probabilistic language Modeling

مدل‌های زبانی احتمالاتی

در پردازش متون

- motivation → چرایی آشنا بیم
- estimation → نحوه محاسبه

Motivation:

Language Modeling

احتمال پیدا کردن یک زنجیره از کلمات ← با توجه به متونی که سیستم دیده

① word prediction ← یکی از کاربردهای NLP

چرا افراد پیشنهاد می‌دهد؟ ← NL ← processing ← management

از روی احتمال تشخیص می‌دهد.

prediction با توجه به چیزهایی که دیده

② speech recognition ← چطور سیستم‌ها جواب درست‌یابی کردند؟

با استفاده از احتمالی

③ machine translation

یه سری فصل‌ها در ترجمه هستند چندتا کلمه داشته باشیم و از اونجا تشخیص می‌دهد

که کدوم رو جواب می‌شه ← اونی که محتمل‌تر است

④ Spellchecker اشتباه کلمه‌ای پیدا می‌خواهد کلمه درست پیشنهاد دهد از روی

احتمال می‌فهمد کدوم محتمل‌تر است پیشنهاد می‌دهد.

سیستم اگر داده زیادی ببیند محتمل با کمک احتمال تصمیم می‌گیرد

← آموزش هم بینه اولیه

Estimation :

براساس شمارش کار می‌کنیم ← تعداد کلمه‌های زبان

یک مجموعه از متون ← پیکره ← corpus

یا آرشیو روزنامه

① The Brown corpus

② The switchboard corpus ← مکالمات صوتی

corporal

امیدوارم با آرشیو استخراج کنیم برای ساخت مدل زبانی

vocabulary (V words) = تعداد کلمه تشکیل شده

که هر کلمه می‌تواند دستش منبع شود

می‌توانیم دو مدل تعریف کنیم

1) word token ← شامل تعداد تکرار کلمات دستش هم می‌شود

2) word type ← تعداد تکرار unique است

یکبار می‌شمریم

چرا اینقدر زیاد؟

غلط‌های املایی / مخفف‌سازی / اسامی / اعداد حرف و فونت

user generated data ← هر کاربری تند تند بنویسید

$P(S)=P(w_1, w_2, \ldots, w_n)$ محاسبه احتمال ترتیبی از احتمالات کلمات

طبق احتمال شرطی به صورت ترتیبی از احتمال شرطی بنویسیم

Scanned with CamScanner

recognize نیستیم و می‌توانیم در احتمال شرطی فقط همان عبارات را می‌توانیم

به صورت تضمین که با همان شرطی باشد تشخیص نمی‌دهد

از طرفی وابسته بودن n کلمه قبلی کار را سخت می‌کند

پس سعی می‌کنیم فقط به یکی یا دو تا واژه قبل وابسته باشیم

markov assumption

$P(S)=\prod_{i=1}^{n} P(w_i \mid w_{i-1})$ ← Bigram زنجیره دوتایی لحاظ می‌کند

$\prod_{i=1}^{n} P(w_i \mid w_1, w_2 \ldots w_{i-1})$

مدل زبانی ساده‌تر

Smoothing

می‌توانیم مدل زبانی را بر اساس یک پیکره بسازیم و بعد برای هر جمله‌ای بخواهد

می‌شه بر اساس اون مدل زبانی احتمالش را حساب کند.

اگر احتمال یک زنجیره دوتایی در یک پیکر صفر بشود ← به این معنا نیست که

به هیچ وجه شانس رخداد وجود نداشته باشد ← واقعیت

باعث می‌شه که احتمال جمله صفر بشود

smoothing 7:

smoothing ⇝ می‌دونیم یک عبارت داشتیم باید به احتمال کوچک نسبت بدیم و اون جلوی صفر شدن تخمین را می‌گیره

↓

① laplace smoothing

اضافه کردن عدد 1 به تمامی count های ما است.

که می‌توانیم کمتر اضافه کنیم

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

$$P(w_i|w_{i-1}) = \frac{\#(w_{i-1}, w_i)+1}{\#(w_{i-1})+V}$$

↓

ماهیت اصلی شمارش از دست می‌دهیم

هوش 0 ← 1 ، 1 ← 2 تبدیل شده

اما فاصله (1-0) خیلی بیشتر از (2-1) است

چون صفر یعنی (ناممکن - ممکن) (ممکن - ممکن)

⇝ پرش خیلی زیاد داریم

دلیلی نداره 1+ کنیم

↓

ε هم کافی است.

اثر وجود داشتن چیزی اضافه نمی‌کنیم خودش می‌مونه

$$P(w_i|w_{i-1}) \begin{cases} \frac{\#(w_{i-1}, w_i)}{\#(w_{i-1})} & \#(w_{i-1}, w_i) > 0 \\ PBG & \text{اگر وجود نداشت اضافه می‌کنیم (مگر نه) PBG} \end{cases}$$

← ماشین احتمال ← Backoff ①

interpolation ②

↓

کاری نداریم احتمال صفر هست یا نه در هر حالت PBG اضافه می‌شه

$$P(w_i|w_{i-1}) = \lambda_1 \frac{\#(w_{i-1}, w_i)}{\#(w_{i-1})} + \underbrace{\lambda_2}_{1-\lambda_1} PBG \qquad \Sigma\lambda = 1$$

ضریب نرمالیزه اثر

parameter tuning

Backpropagation

Elipon

Scanned with CamScanner

smoothing 2:

trigram ⇝ bigram
bigram ⇝ unigram
unigram ⇝ zerogram ($1/V$)

(P_{BG}) Backpropagation محاسبه احتمال

هر مدل زبانی یک مدل زبانی بالاتر به عنوان PBG

bigram ⇝ unigram
فوق‌العاده تر می‌شود
zerogram

محاسبه می‌شود (که فوق‌العاده به تر تبدیل می‌شوند)

* با استفاده از ۲ روش بالا چند مدل ساخته شد:

Advanced smoothing

1) Bayesian smoothing with dirichlet prior

$$P(w_i \mid w_{i-1}) = \frac{\#(w_{i-1}, w_i) + 1}{\#(w_{i-1}) + V}$$

$$P(w_i \mid w_{i-1}) = \frac{\#(w_{i-1}, w_i) + k}{\#(w_{i-1}) + kV}$$

عدد صفر و یک

$$P(w_i \mid w_{i-1}) = \frac{\#(w_{i-1}, w_i) + M\left(\frac{1}{V}\right)}{\#(w_{i-1}) + M}$$

$M = kV$

PBG ← zero gram

$$P(w_i \mid w_{i-1}) = \frac{\#(w_{i-1}, w_i) + M\, P_{BG}}{\#(w_{i-1}) + M}$$

Scanned with CamScanner

۲) Absolute Discounting

به مقداری از اونایی که غیر صفر است احتمال نمیدیم و به مقداری به بقیه میدهیم

$$P(w_i|w_{i-1}) = \begin{cases} \dfrac{\#(w_{i-1}, w_i) - \delta}{\#(w_{i-1})} & \text{if } \#(w_{i-1}, w_i) > 0 \\ \alpha P_{BG} & \text{در غیر این صورت} \end{cases}$$

تعداد اون هایی که احتمال بزرگتر از صفر دارند

$$\alpha = \frac{S}{\#(w_{i-1})} \cdot B$$

به صورتی:

$$P(w_i|w_{i-1}) = \frac{\max(\#(w_{i-1}, w_i) - S, 0)}{\#(w_{i-1})} + \alpha P_{BG}$$

چون منفی میشد.

۳) Kneser-Ney smoothing

تنوع واژه‌های قبل از یک واژه

شمارش تنوع یک واژه در کنار بقیه واژه‌ها

تعداد کلمه را نشان می‌داد

جای unigram استفاده میشد که شمارش

مبنای کلماتی که یک بافت خاص دارند توجه میشد

که چقدر زیاده

continuation محاسبه‌ی احتمال BG

$$P_{continuation}(w) \propto |w_{i-1} : \#(w_{i-1}, w)| > 0$$

تعداد کلمات متنوع قبل از کلمه w

کلمه‌ی w

Elipon

Scanned with CamScanner

باید normalize کنیم.

برای مدل‌های Bigram به بالاتر قبل استفاده است.

$$P(w_i \mid w_{i-1}) = \frac{\max(\#(w_{i-1}, w_i) - S, 0)}{\#(w_{i-1})} + \alpha P_{continuation}$$

$$\alpha = \frac{S}{\#(w_{i-1})} \cdot B$$

f) Bayesian smoothing based on pitman-yor process

۱) Drichlet prior ۲) Absolute discounting

ضریب discount یا حاصل تعداد یک Bigram

جمع تمامی t ها یا discount ها $t_0 = B$

$$P(w_i \mid w_{i-1}) = \frac{\#(w_{i-1}, w_i) - S + (M + S \cdot B) P_{BG}}{\#(w_{i-1}) + M}$$

قرار شد هر اونایی که احتمال دارند S کم شود ← اما اگر count زیاد دارد بیشتر کم شود. و سعی به تعادل متفاوت است.

$t = 1 \rightsquigarrow$ همان طبیعی

$M = 0 \rightsquigarrow$ absolute discount

Scanned with CamScanner

Evaluation

محاسبه t کار سخت است برای تک تک موارد

این همه قانون زیاد — یک مدل روی رابطه‌های نمایی داریم

رابطه‌های count رایج

محاسبه نمایی روش می‌کند

Entropy

مقادیر که در یک متغیر تصادفی وجود دارد

چه تعداد بیت کافی که نیاز داریم برای انتقال اطلاعات

$$H(V) = \sum_{w_i \in V} -P(w_i)\log(P(w_i))$$

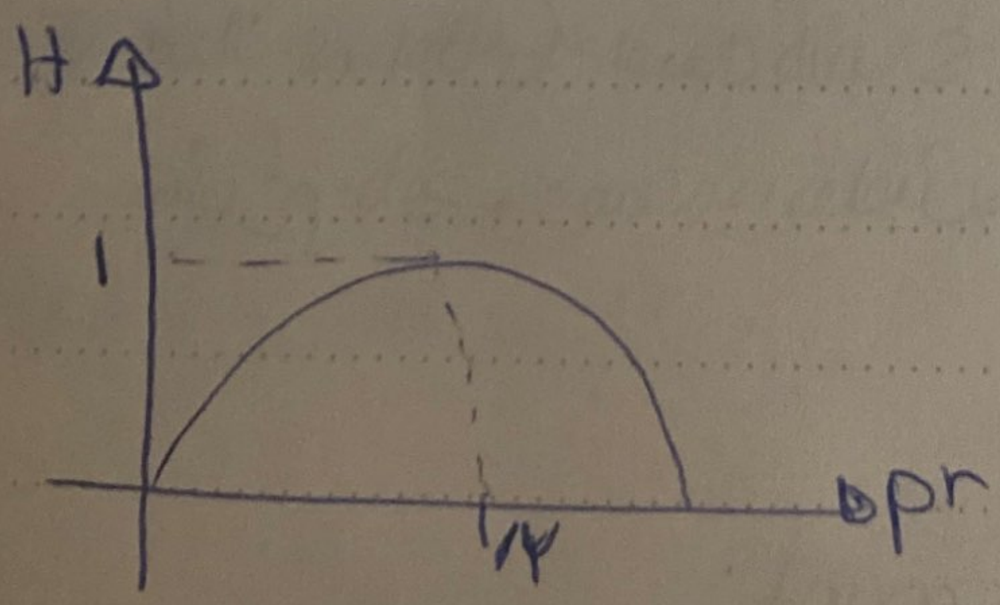
می‌توانیم اعداد را در مبنای دو بهشان دهیم.

برای همه اتفاقات

اگر دو کلمه داشتیم فقط احتمال یکی صفر باشد — یعنی هیچ گاه یک چیزی رو نداریم و فقط

اون یکی که صفر نیست احتمالش وجود دارد و نیازی به انتقال اطلاعات

نداریم و آنتروپی صفر می‌شود.



اگر دو کلمه باشد

اگر w کلمه داشته باشیم $P(w_i) = \frac{1}{w}$

$$H = -\log\left(\frac{1}{w}\right) = \log(w)$$

— چه تعداد بیت های لازم برای کدگذاری کلمه

Elipon

Scanned with CamScanner

آنتروپی شرطی هم داریم / آنتروپی قانون زنجیره‌ای هم دارد / اصطلاحات متقابل

آنتروپی داده

* ارتباط آنتروپی و زبان‌شناسی

← برای مدل زبان

① به یک نفر استامش نشان می‌دهیم و کاراکترها را مدل بزنه

② می‌تونه همه‌ی هر کاراکتر با چندمین و خصوصیات مدل شده است

↓

آنتروپی نشان می‌دهد چقدر نامطمئن هستیم ← ارتباط با آنتروپی یک زبان داره

برای پیش‌بینی واژه‌ها رفتن

* ارزیابی زبان

① perplexity ← معیار ارزیابی

چقدر توزیع داریم برای مدل ← یک واژه در یک متن

چقدر سردرگمی داریم برای اینکه بتوانیم واژه را مدل بزنیم

$$2^{H(P)} = 2^{-\sum_{w_i \in V} P(w_i) \log(P(w_i))}$$

آنتروپی ← $H(P)$

هرچی perplexity کمتر باشد برای ما بهتر است

← نشان می‌دهد میزان سردرگمی کم است.

یک اصطلاح دیگر که استفاده می‌شود Branching Factor است.

چندشاخه‌ای‌ها توسعه بخوص هر زمان

ماکسیمم برای هر زبان ۷ است (یعنی تمام کلمات) ← ابتدای کار ← upper bound

با استفاده از درک خوبی از تخمین زبانی از پیش بدست آورده باید همه کلمات را لحاظ کنیم

و فشرده شده

lower bound ← میفهم

* دنبال مدل زبانی هستیم که ← ۱) بیشتر از همه جوانب کلمه نامناسب را کنار بگذارد

← ۲) سرگشتگی روی یک مقدار محدود باشد.

احتمال یک تخمین زبانی

$$P(S) = P(w_1, w_2, \dots, w_n)$$

$$\text{perplexity}(S) = \sqrt[N]{\frac{1}{P(S)}} = P(S)^{-\frac{1}{N}} = 2^{H(P)}$$

ارتباط بین احتمال با استفاده از perplexity ← هرچی احتمال ↑ و P(S) کمتر.

با استفاده از مدل زبانی قوی

مدل زبانی روی ← train ساخته می‌شود

← bigram unigram ،

← test داده ← احتمال تک تک کلمات جمله test را حساب می‌کنیم

perplexity در 4-gram و 5-gram بالاتر آمده.

Scanned with CamScanner

chapter(۳)

مقدار پارامترها را باید مناسب انتخاب کنیم (m و S و ...)

برای انتخاب پارامترها داده را به دست می‌آوریم و cross validation وارد مقدمه می‌شود

با استفاده از مقادیر مختلف که پارامتر می‌تواند بپذیرد مدل زبانی می‌سازیم

بعد از آن perplexity روی داده cv حساب

بر اساس آن می‌بینیم که کدام مقادیر پارامتر مناسب است.

* ما هیچ اطلاعاتی نداریم که ما دصدی که به test میدیم بخش شده یا بخش نیست

برای حل مشکل: ← ما دصدی از کل بخش داده نباشد و کنار بذاریم

۱۰ بار ← ما بخش‌های مختلف داده را برای test بدهیم و نمایشات میانگین بین ۱۰ بار اجرا عدد عادلانه‌تری می‌دهد.

10-fold cross validation

k-fold cross validation ← یکی اگر ۲۰ درصد داده تست داشتیم می‌شد ۵ fold

* داده‌ها وقتی خیلی کم است ← از تست کردن داده به عنوان test خوب نیست

هر دفعه یک sample ۵ تکه تست و بقیه می‌شه داده train

تعداد داده‌ها زمان زیاد نیست

۵۰۰ داده داریم ← ۵۰۰ بار اجرا ← هر دفعه یکی از داده‌ها را کنار می‌ذاریم

leave-one-out-cross-validation

Scanned with CamScanner

Neural language Modeling — مدل‌های زبانی عصبی

single layer perceptron ← اندازه ثابتی نداشته باشیم

multi layer perceptron ← اندازه ثابتی داشته باشیم

Error function ⇝ MSE

updating weights and bias based on the error
gradient descent

در مدل‌های زبانی با استفاده از تعدادی از کلمات کلمه بعد را پرمیلیت می‌کردیم،
prediction ⇝ با استفاده از شبکه عصبی

شبکه عصبی ← ورودی کلمات (w_t)
خروجی ← کلمه بعدی باشد. (w_{t+1})

← فعلا بر اساس Feed forward استفاده می‌کنیم،
اما بعدا می‌گوییم از RNN و transformer هم استفاده کنیم

تقریبی که قبل

$$P(w_t \mid w_1^{t-1})$$

← کلمه بعدی

در این شبکه ما با استفاده از یک پنجره که
تحت sliding window هست کار را می‌بریم جلو.

خروجی ما کلمه است که
در پیکره وجود دارد.

پس خروجی ما است که
هر کلمه یک احتمالی دارد.

Elipon

Scanned with CamScanner

قبلاً می‌دیدیم N-gram استفاده می‌شد هیچ ابتدایی از معناد مفهوم کلمات نداشتیم

اما در neural model ساختار معنایی استخراج می‌کنیم.

chapter (7)

تبدیل خروجی به one hot در شبکه عصبی

?

Sparse word representation

word representation

جای نمایش کلمات

← از مدل‌های پایه‌ای شروع می‌کنیم.

sparse ← مدل‌های قدیمی

Dense ← مدل‌های جدید

word concept

مفهوم کلمه

← ① مواجهه به لغت نامه و یافتن معنای آن

sense = معنای مختلف واژه ← معنای مختلف واژه

به ازای هر sense (concept) یک definition (تعریف) داریم

words, lemmas (n/v/adj), senses, definition

زمانی که می‌گوییم دو کلمه به هم مرتبط هستند این ارتباط روی sense آن کلمه هست نه خود کلمه ← مثل شیر و پلنگ / شیر و قهوه

منظور کل واژه شیر نیست و فقط یک sense خاص مورد توجه است.

Scanned with CamScanner

* منظور از "جامع" معنای واژه" و "ارتباط واژه" صحبت می‌کنیم منظورمان می‌از

sense های واژه است نه کل واژه

* روابط بین واژه‌ها دسته‌بندی می‌شوند تحت چند دسته مختلف:

① synonyms → تقریباً یعنی یعنی هم (نیستند)

که‌شه مورد استفاده آنها متفاوت باشند

مثل تو جیب اتو چوله

هرجا تو جیب استفاده می‌شود نمی‌شود تو چوله بگذاری تو

politness / slang / register / genre

واژه‌ها وجود دارد.

بشین و بفرما

② Antonymy → binary opposition up / down

شوخ

یک نفر سرعت داریم که انتها شده است (دو سر زنجیر را داریم می‌بینیم)

revesives (زنجیر داریم)

③ Similarity → هم معنا و جایگزین نیستند اما

یک معنا شبیه هم هستند

ماشین و خودرو — synonyms

ماشین دودوزه — similarity

گاو و اسب — similarity

برای تعیین similarity می‌توانیم برای دو واژه "به similarity" تعریف کنیم.

Elipon

Scanned with CamScanner

④ word relatedness

یعنی دو واژه چقدر به هم مرتبط هستند

Similarity ← type یکسان بودن مفاهیم / میزان شباهت بین

اما اینجا نیز type یکسان نیست مثل خودرو / کارخانه

type متفاوت هستند

یک مفهومی وجود دارد که این ها را به هم مرتبط می کند

word relatedness تعریف اسم یک دیدگاه دیگر

hospital

surgeon, nurse, scalpel

resturant

water, chef, menu ← type زیر یک چتر موضوعی یا یک

type متفاوت

نیستند ولی به هم ارتباط دارند.

⑤ subordinate / superordinate

اگر بخواهیم مفاهیم واژه ها بیشتر بیاوریم یا توصیف کنیم یا نام بگذاریم

باید بدانیم در چه سطحی می خواهیم این اطلاعات منتقل کنیم

و اگر مدل نیز وقتی بخواهد subordinate

اگر کلی و general ← superodniate

مثلا ماشین رو وسیله نقلیه خطاب کنیم

اگر مبلی رو furniture بگیریم مثلا اینه رو میز خطاب کنیم

پس مهم است بدانیم در چه سطحی داریم نامگذاری می کنیم

Scanned with CamScanner

subordinate ← که تعلق بیشتر شود

حیوان ← سگ ← نژاد سگ

وسیله نقلیه ← برند ماشین ← نام ماشین

گیاه ← گل ← اسم دقیق گل

(۴) connotection subordinate / basic / superordinate

با استفاده از این چند دقیق تعریف می‌کنیم می‌توانیم فاصله‌ها بین مفاهیم را مشخص کنیم.

اگر سگ و حیوان تعریف کنیم از نظر ویژگی و سگ ربط دارد

اما اگر سگ را با نژاد بیان کنیم از نظر ویژگی فاصله‌ش بیشتر و حساب‌دیگه هست

قرار دهیم.

* برای تعریف یک واژه ← یک سری تعاریف ثانوی داشته باشیم ← مشترک داشته باشیم

مثلاً میز ← که پایه داشته باشه

سخته انقدر معنی‌ها ریز و دقیق هستند مثل cup / bowl

* برای تعریف مفهوم ← یک سری feature-based تعریف کنیم

تعریف ویژگی که ساده‌ای نیست به بچه‌ها

نمی‌آییم با ویژگی بگیم بعد چند بار مقایسه‌ش تفاوت را متوجه می‌شود.

← یک سری context-based تعریف کنیم.

مفهوم را از کاربرد بدست آوریم ← تو ظرف‌هایی سرو شده

چای ← cup

قهوه ← bowl

Elipon

Scanned with CamScanner

* اگر دو واژه A و B کلمه باشیم اگر در جمله بیشتر استفاده شوند دو واژه Similar و یا synonym هستند

* اگر بتوانیم واژه ها را بر اساس معنایی مثل دمیم در یک فضای برداری ببریم می توانیم بر اساس فضای کاربردی بازنمایی کنیم و معنا استخراج کنیم

word vectors word concept

واژه هایی که بهم مرتبط هستند کنار هم قرار دهیم.

واژه ها را در فضای برداری بازنمایی کنیم

با استفاده از context و امید باشیم معنا هم دریافت شده

embeding → جاسازی

خلاصه:

① بر اساس context based (فضای کاربردی) واژه را بازنمایی

② سعی بر استخراج معنا

③ با استفاده از context و معنای دریافت شده واژه را در فضای برداری بازنمایی کنیم

برای بردن واژه در فضای برداری (embeding)

sparse کند

Dense چقالتر

کوتاه تر

sparse vector representation 2

word Term document * ماتریس

سطرها کلمات و ستون‌ها اسناد

متن، فیلم، داستان، ...

اطلاعاتی که داریم را با ماتریس نشان می‌دهیم.

اطلاعاتی که از اسناد داریم با استفاده از این ماتریس نمایش می‌دهیم

سندها را در فضای برداری نمایش دهیم.

محورها ← کلمات ← که برای نمایش ۲ تا سند در نظر می‌گیریم.

می‌توانیم مشاهده کنیم هم سندها به هم نزدیک هستند

و شباهت بیشتری دارند

* می‌توانیم به جای بازنمایی اسناد ← کلمات را بازنمایی کنیم. (یعنی محور سند اسناد)

جای سطر و ستون عوض می‌شود. (بردار هستند کلمات)

اگر دو واژه در بیشتر اسناد زیاد تکرار حدیف بسی کم تکرار شده اند

می‌توانیم بگوییم هم‌معنی هستند.

دو واژه مشابه هستند اگر در اسناد مشابه ظاهر شده باشند

word word ب term term *ماتریس

در این ماتریس می‌توییم دو کلمه نزدیک به کلمه‌ی آماده است

که بخوره تعریف کنیم

بردار واژه نشان می‌دهد به کلمه‌ای در نزدیکی آن هستند

زمانی که همسایگی شبیه‌شان داشتند ← به صورتی انتظار می‌توانیم جای هم استفاده کنیم

* برای مشابهت واژه‌ها ← بر اساس همین فضای برداری ← بردار واژه‌ها را استخراج کنیم

و سپس فاصله cos را محاسبه کنیم ← هر چه زاویه کمتر مشابهت بیشتر

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

شبیه‌ها

raw frequency (شمارش ساده‌ی کلمات) ← محاسبات ماتریس چگونه پر شود

tf-idf

term frequency

inverse document frequency

تعداد تکرار در یک سند

یک کلمه چقدر مهم هست

مثلا ترجیح می‌دهیم موقع سرچ قرص استامینوفن ← قرص استامینوفن ✓

دومی را ترجیح می‌دهیم چون خاص‌تر و نزدیک‌تر هست.

Scanned with CamScanner

* total docs in collection

$$\log\left(\frac{N}{df_i}\right)$$

↳ # of docs that have word i

سیستم چقدر متوجه می‌شود کلمه واژه مهم‌تر است؟

کلمه‌ای که خاص‌تر است ← مفهوم دقیق‌تری می‌رساند

← از تعداد تکرار اسناد متوجه می‌شویم کلمه خاص هست یا نه

هر چی کمتر می‌تواند خاص‌تر باشد.

$$tf_{t,d} \begin{cases} 1+\log_{10} count(t,d) & \text{if } count(t,d)>0 \\ 0 & \text{otherwise} \end{cases}$$

$$\boxed{w_{t,d}} = \overset{\text{ساده}}{tf_{t,d}} \times idf_t$$

برای محاسبه وزن با استفاده از tf_idf مقایسه‌ی context

ظاهر شده اش ← سن یا فاصله دو شباهت بدست آوریم.

Elipon

Scanned with CamScanner

یک معیار دیگر برای مقایسه دو واژه

positive point wise mutual information (ppmi)

اطلاعات متقابل

هست.

مثل هرچی و هرچی

نقیه واژه‌ها

* وقتی دو واژه با تعداد تکرار زیاد در کنار هم ظاهر شده باشند و در کنار بقیه واژه‌ها کمتر باشند در این صورت این دو واژه اطلاعات متقابل زیادی دارند

احتمال با هم آمدن

$$pmi = \log_2 \frac{p(x,y)}{p(x)p(y)}$$

احتمال تک تک آنها

pmi از $-\infty$ تا $+\infty$ است که ما فقط مثبت‌ها را در نظر می‌گیریم

زیرا

positive

$$ppmi(x,y) = \max\left(\log_2 \frac{p(x,y)}{p(x)p(y)}, 0\right)$$

$p(x,y)=1$

$p(x)=1$

$p(y)=1$

جایی که آمار خیلی زیاد هستند یا اشتباه تایپی داریم

وقتی زیاد می‌شوند

واژه‌هایی که در کنار هم‌اتفاقی می‌شوند

weighting pmi

در کنار pmi کلاً خود واژه‌ها را هم در نظر می‌گیریم

اینکار جلوی واژه‌هایی که تکرار کم ولی pmi بالا دارند را می‌گیرد

Dense word representation ؟

feature_based ← فرهنگ لغت ← برای بدست آوردن مفهوم واژه

context-based ← بافت معنایی نمایش داده شده

① همیشه update نیست

تمید بافت کار سختی نیست ولی منبع هست

google یک شرکت است در فرهنگ لغت

اما از قدر متاخل شده که امروزه خودش عمل استفاده می‌شود و باید در فرهنگ لغت

آپدیت شود.

② تمید فرهنگ لغت میزوه انسانی می‌خواهد و سخت است

③ انسانی خاص در فرهنگ لغت نیست اما با فضای کاربردی می‌توانیم ارتباط دو نوع و ... بفهمیم

برای ایجاد فضای برداری با استفاده از context-based :

می‌توانیم فضای برداری بزرگ و تنک داشته باشیم

کوچک و چگال

بیشتر مقادیر ماتریس صفر

sparse

Dense

sparse → در بردار هستند ← ماتریس مقادیر صفر زیاد داریم

→ در بردار onehot ← واژه را بدون کلمات context آن را نشان می‌دهیم

به اندازه‌ی سایز داده‌های vocab بردار درست می‌کنیم و اندیس متناظر واژه را 1 می‌کنیم.

Elipon

Scanned with CamScanner

* در one-hot شباهت واژه ها را نمی توانیم درک کنیم.

* تعداد vocab زیاد است اگر بخواهیم یک بردار برای واژه ها بسازیم چه به صورت one-hot چه به صورت همسایگی خیلی ابعاد بزرگ است ← memory efficency

↓

فضای برداری sparse خیلی بزرگ

اما فضای برداری Dense ← دوستش داریم

و مفاهیم منتقل می شد و داده صفر تقریباً هیچ است.

با استفاده از matrix factorization

کاهش بعد دهیم

Dense → SVD

- Neural language model → prediction مبنای
  و از آن برای بازنمایی کلمات بردارها استفاده می کنیم
- brown clustering خوشه بندی

* کاهش بعد ماتریس ← PCA / SVD

← ۲ تا ماتریس داشتیم ← term/document term/term

در SVD یک ماتریس $w \times c$ به عنوان ورودی گرفته می شود. این ماتریس به ۳ ماتریس تقسیم می شود $w \times m$ و $m \times m$ و $m \times c$

← مقادیر ویژه به صورت آماده در متلب sort می شوند و ماتریس قطری

می تواند حداکثر $\min(w, c)$ بعد ماتریس دومی باشد

اما باز هم خیلی بزرگ باشد.

می توانیم ماتریس مقادیر ویژه را چون sort شده هستند از یک جایی قطع کنیم

فقط از ابتدا تا آنجا را در نظر بگیریم

$$w \times c = \underbrace{w \times k}_{\text{عصاره کلمات}} , \underbrace{k \times k}_{\text{به سرعت محاسبه می‌شود}} , \underbrace{k \times c}_{\text{عصاره context}}$$

* از ماتریس term document را روش SVD اعمال کنیم LSA به وجود می‌آید

latent semantic analysis

* ماتریس term term را می‌توانیم تجزیه کنیم:

$$|V| \times |V| = \underbrace{|V| \times |k|}_{w} \text{ و } \underbrace{|k| \times |k|}_{S} \text{ و } \underbrace{|k| \times |V|}_{C}$$

مقایسه Dense (Embeding) با Sparse:

① با فرض شباهت فضای برداری در Dense سریعتر است

② در Dense اطلاعات معنایی بهتر نهفته است.

مشکلات SVD ← هزینه بالا ← کار سخت خواهد بود.

← اگر هر لحظه ایجاد و اضافه شود کار پرهزینه است.

← داده جدید کل پروسه از اول

Elipon

Scanned with CamScanner

Dense word respresentation 2

* کلماتی که context مشابه دارند ← باهم ارتباط معنایی دارند.

* کلماتی که context مشابه دارند باهم ارتباط داشته باشند

با استفاده از prediction وزن‌های نزدیک را بدست می‌آوریم.

prediction_based models

① هزینه کمتری داریم

② پروسه train از اول رخ نمی‌دهد با اضافه شدن داده جدید می‌توانیم از مدل‌ها pretrain استفاده کنیم

skip-gram CBOW

با استفاده از word2vec این کار را انجام می‌دهد.

فقط ما predict انجام می‌دهیم و دیگر نمی‌شماریم در word2vec

یعنی می‌دانیم چقدر محتمل هستیم که کلمه بعدی می‌باشد.

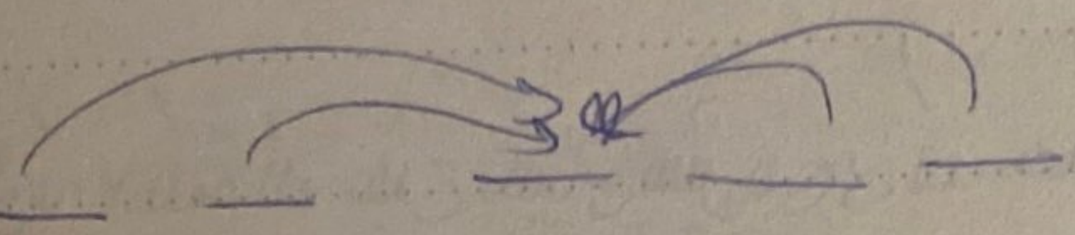
از فضای برداری متن prediction استفاده می‌کنیم

Scanned with CamScanner

برای train شبکه نیاز به برچسب نداریم ← صرفاً وجود داده‌ی متن هم کافی

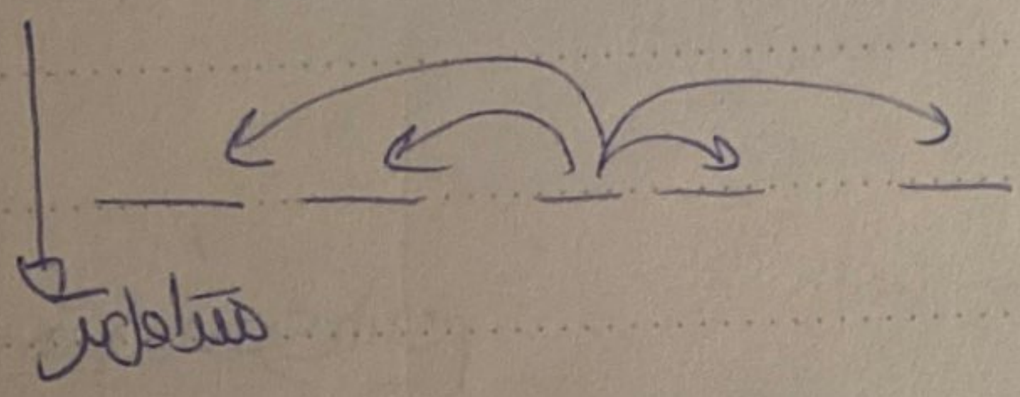
CBOW ← قرار است یک کلمه target داریم و n کلمه قبل و بعد را داریم

و می‌خواهیم آن کلمه را حدس بزنیم



skip-gram

n کلمه قبل و بعد را حدس بزنیم



متداول‌تر

پیش‌بینی c از کلمات اطراف c+ c-

همسایه

target

$p(t \mid c)$

احتمال شرطی قرار گرفتن t و c

* شباهت $p(w_{t+1} \mid w_t)$

* تابع هدف ← احتمال $p(w_{t+j} \mid w_t)$ را بیشینه کنیم

$$J(\theta) = \frac{1}{T} \sum_{t=1}^{T} \sum_{-m \le j \le m,\ j \ne 0} \log p(w_{t+j} \mid w_t)$$

m اندازه window است.

Elipon

Scanned with CamScanner

input embedding
output embedding

می‌توانیم دو بازنمایی برای هر واژه داشته باشیم

دو وزن داریم.

input layer	hidden projection layer	output
one hot input vector	embedding	probabilites of context word
1×V	1×d	1×V

* اگر می‌خواهیم به ازای دو ورودی خود و اتومبیل پیش‌بینی یکسانی داشته باشیم چون تک فضا هستند ← تقریباً وزن‌های این دو مشابه هم می‌شوند

* پس اگر کلمات با هم ارتباط و شباهت داشته باشند ← وزن‌های تقریباً یکسانی دارند و خروجی یکسانی می‌دهند.

از وزن‌ها برای بازنمایی واژه‌ها استفاده می‌شود

چه از hidden و input
چه از output و hidden

d

هیپرپارامتر شبکه

فرض نه مثلاً

چون تقریباً یکسان هستند

در فضای برداری زاویه نمی‌دارند

input embedding $|V| \times d$

output embedding $d \times |V|$

از input به hidden

از hidden به output

$P(w_{t+j} | w_t)$

می‌توانیم این احتمال را با softmax و ارزش‌های همسایه انجام دهیم

$$P(w_s | w_t) = \frac{\exp(u_{w_s} \cdot v_t)}{\sum_{w \in V} \exp(u_w \cdot v_t)}$$

همسایه

کلمه‌ها رفیل می‌شوند

سشن هست

می‌توانیم V بار نمایی را ← ① طی شان را ضرب شرط

② جمع بزنیم

③ concat بزنیم

از تعداد negative همسایه sample بگیریم و به تعدادی لینک بزنیم

تا این‌ها قرار نمی‌شود

positive sample اونایی که همسایگی قرار دارند

negative sample ← اونایی که همسایگی target نیستند

$$L(\theta) = \log P(+ | t, c) + \sum_{i=1} \log P(- | t, n_i)$$

رابطه skipgram و pmi :

$$w w'^T = M^{pmi}_{|V| \times |V|} - \log k$$

استفاده از هر دو اطلاعاتی را از دست نمی‌دهیم

بیشتر هستیم.

Elipon

Scanned with CamScanner

این دو روش skip-gram و CBOW با هم در ارتباط هستند و به تولید یک روش مهم برای بازنمایی کلمات هست

Glove ←

بهترین را از هر دو جمع کرده

$$J(\theta)=\frac{1}{2}\sum_{i,j=1}^{W} f(P_{ij})\,(u_i^T v_j-\log P_{ij})^2$$

Dense word representation 3

* ارزیابی بازنمایی ① برای هر واژه بازنمایی برداری بدست آوردیم.

بعد به ازای هر واژه n تا واژه مشابه بر اساس شباهت cos

با آن واژه بدست بیاوریم.

چقدر واژه‌های درست و منطقی هستند.

② در بردارهای word 2 vec بدست می‌آوریم — یک رابطه بین کلمات وجود دارد

woman — man

king

$$V(king)-V(man)+V(woman)=V(queen)$$

نزدیکترین بردار queen بوده.

اگر شبیه تشخیص دهد ← شبیه است

ارزیابی روی چه داده‌هایی App

استفاده می‌کنیم

- internal evaluation
  - word similarity ①
  - word analogy ②
- external evaluation ③ ← بازنمایی‌ها task خاص استفاده کنیم

Scanned with CamScanner

④ برای ارزیابی می‌توانیم از یک سری dataset استفاده کنیم.

① word sim-353

یک مجموعه pair کلمات + عدد که نشان‌دهنده شباهت دو کلمه است

می‌آییم این کلمات را با embedding می‌کنیم به vec ← در فضای برداری ← cos محاسبه می‌کنیم

و عدد رو می‌بریم تو رنج ۰ تا ۱۰ با عدد مشخص شده در جدول مقایسه می‌کنیم.

با استفاده از Spearmen ارتباط را بدست آوریم

② word analogy ← / google analogy / هم دستوری / هم روابط معنایی

لیستی از جملاتی داریم

king / man / woman / ?

جای خالی زده می‌شود اگر درست ← امتیاز می‌گیرد

دصدی که از این ۴ تایی درست تشخیص داده شود فرض داده می‌شود.

external evaluation → تو App اعمال کنیم

③ می‌توانیم مقایسه را چیزای external و interal بین دو روش embedding مثل skip gram

و cmob اعمال کنیم

Elipon

Scanned with CamScanner

similarity depends on window size

similarity depends on training data

Text preprocessing → practical

پیش پردازش متن

* تمامی که متن‌ها می‌تواند usergenerator باشند تنوع متن‌ها زیاد می‌شود.

* web 2 ← جایی که افراد راحت می‌توانند متن بنویسند (شبکه‌های اجتماعی)

* web ← خبرگزاری ← یک فرد خبر می‌نویسد

چت‌بات ← یک دور کاربر با هم رد و بدل است.

یک دور چت‌بات است.

کاربران بیان و نگارش متفاوتی دارند

ion't = i don't

می‌خواهیم ترکیب دو زبان داشته باشیم

رابطه جنسیت متفاوت ← خبری / علمی / تخصصی

یک تراند افراد خیلی تأثیرگذار است

نشان دادن تنوع آن

① سن

② جنسیت

③ پیشزمینه‌های اقتصادی و سیاسی: شاغل

social economy status

* تشابه هستند داریم

① Tokenization ② normalization ③ lemmatization

④ Stemming ⑤ stopword removal

Scanned with CamScanner

*Tokenization ← قطعه‌بندی متن ← یک قطعه یا واحد از آن می‌شود

متن را قطعه جمله تقسیم کنیم

متن را قطعه کلمات تقسیم کنیم

* قطعه‌بندی در زبان‌های لاتینی و یونانی چون فاصله داریم راحت است ← str.split(" ")

اما در چینی چون فاصله نداریم سخت است

* دستور tr در shell بر اساس کلمات

numerical

sort | uniq -c | sort -nr

sort: بر اساس کلمات

uniq: unique

-c: count

sort -nr: به صورت کاهشی صورت شود.

* نمی‌توانیم punctuation برداریم چون ممکن است وسط جمله باشد و مهم باشد

m.p.h
P.h.D
99055
01/02/09
url
* hashtag

punctuation های مهم دارند و حذف آنها مشکل‌ساز است.

* اسم‌های چندکلمه‌ای MWE داریم یعنی space داریم ولی باهم معنا دارند مثل "New york"

* ابزار NLTK برای Tokenization ← pattern تعریف کنیم و بر اساس آن tokenize کنیم

Scanned with CamScanner

* در بعضی زبان‌ها space نداریم و کار tokenization خیلی سخت است ← حاشیه زیادی داریم

* در sentence segmentation ما یک سری علامت‌ها را به عنوان جدا کننده جمله می‌دانیم ؟!

مثلاً yahoo answers! ← قرار است یک پارامتری در متن ایجاد کنیم

یک کلمه است نه جمله

همه وقت از یادگیری ماشین استفاده کنیم که هر نقطه‌ای که از متن بدیم classify بشه این یک نقطه هست یا نه.

یا می‌توانیم rule base بدیم جلو که در این صورت regular expression‌های خیلی دقیقی نیاز داریم

*Normalization

U.S.A → USA
uhhuh → uh-hun
fed → Fed

در کل متن مثلاً بین حروف کوچک بشه یا نه یا - داشته باشیم یا نه یا F بزرگ داشته باشیم یا نه

① کوچک کردن تمامی حروف در متن متن ← باید تعیین کنیم استثناها را چک کنیم

مثلاً اسامی خاص

US → us
آمریکا ما

MAY → may
مه فعل

* یک سری تغییرات می‌توانند معنی را عوض کنند

* Lemmatization

در افعال تنوع فعلی زیاد داریم

به دلیل صرف فعل ها متفاوت

در پردازش متن خود فعل برای ما مهم است اما جزئیات آن برای ما مهم نیست

در این شیوه lemmatization انجام می شود

می بریم شما را در حالت مصدری آنها

am, is, are → be

He is reading stories → He be read story

Morphemes → می گیریم اجزای یک کلمه را پیدا کنیم

Stems → ریشه / بن

affixes → پیشوند و پسوند

morphemes انجام می شود پیشوند

برای انجام lemmatization : ①

می شود

② به حالت مصدری تبدیل می کنیم ریشه را

* stemming

lemmatization تقریباً هدف مشابه

از بین بردن تنوع برای رسیدن به یک حالت پایه

parsing انجام نمی دهیم ← شما لغات را

① این کار سخت تر است و کار تحلیلی و

پیچیده می شود ← ممکن است در لغت نامه وجود نداشته باشد و وارد بحث شده

و جلو عقب به چیز بدهیم

اما باز Stem می گیریم

copy ~ copi

اگر همه شیوه ها تغییر کند مشکلی ایجاد نمی کند

Elipon

Scanned with CamScanner

* وقت فرود می‌توانی اهمیت کلمه‌ها را سرعت تحلیل و مدل‌سازی بیشتر است از stemming استفاده می‌کنیم

* stopword removal

زمانی که یک پیکره داریم خیلی کلمات اسم‌ها زیادی دارند معنای خاصی منتقل نمی‌کنند
مثل the و and یا و و که

side effect دارند

که چون در بسیاری اسمیت مستقیمی خیلی زیاد تکرار شده‌اند و حدود ۲/۱ یا ۳/۱ پیکره
می‌باشند و حذف آنها می‌تواند کمک بزرگی به ما در خیلی مسیرها بکند

خرید از آمازون

اما اینکه حذف کنیم یا نه به task بستگی دارد

اهمیت کمتری از ۲ کلمه دیگر دارد

می‌توانیم با استفاده از tf-idf اهمیت را کم کنیم

حذف کنیم ← هنوز استفاده یافته از بین می‌رود.

to be or not to be

اگر stopword removal انجام دهیم کلش از بین می‌رود و هیچی باقی نمی‌ماند

در حالی که این ۶ تا stop word داریم و معنا داریم

---

spell correction

غلط‌یابی و اصلاح غلط املایی

زمانی که با user generated data سروکار داریم احتمال خطا بالا است

مثلاً در فضای اجتماعی یا چت‌بات

ما داده‌ها را به صورت بردار نشان می‌دهیم ← زمانی که با فضای نیستیم مستقیماً نمی‌توانند تشخیص دهد و از آن استفاده کند

Scanned with CamScanner

برای کاهش زمان کمتر task ها خیلی خوب است

* روش مقابل استفاده می‌شود. Minimum edit distance

Accdring → According ← همینم

چون زمان سرچ زیاد می‌شه می‌آد فضایی تعریف بالقوه سریع تر را capture کنه و تشخیص بدهد

همینطور ابتدا کامل‌تر انجام می‌دهیم

اشتباه غلط املایی زیاد ← وقتی املا ما اشتباه هاست توجه غلط املایی نمی‌شه و بهمینم برمی‌گردد

← طبیعی موجود نیست تشخیص خطا ← Detection

Spell correction

correction ← اصلاح خطا ← علاوه بر تشخیص

وارد دست تشخیص هم هم

aters detection

correction ← waters later?

نزدیک هر آنچه در dic نیست خطا نیست

ممکنه اسم خاص باشد.

typeهایی که موتورها خطا دارند

① استفاده نکردن از space

② نزدیکی char ها در کیبورد

③ شباهت شکل ظاهری

④ خطاهایی که بر اساس املای phonetic

⑤ خطاهایی که از دانش کم افراد حاصل می‌شود

Elipon

Scanned with CamScanner

① همین جا به اشتباه میگید space فرامونش میشه
the fuzz → the fuzz

equalization → equali zation یک واحد سمت چپ و راست

② دو char بهم نزدیک هستند jack shack

③ tight for fight

④ کلمات بیان یکسان دارند اما نوشته متفاوت دارند
red / read ← خوانش یکسان
ملت نشسته
دفاعی تهاجمی کلماتی که خوانش یکسان
زارز / خا مل / س
یا cite / site / sight
یا نوشته
جنبه لحن سنگ الله
site ologee

⑤ داشتم برای املایی وارد نداریم

motivation برای رفع خطای املایی

در بحث minimum edit distance ← کلماتی که خطا داریم به احتمال خیلی زیاد کلمه ای که اشتباه نوشته شده
باید بهترین است

باید بگردیم دنبال واژه با هیستوگرام فاصله از نظر edit مرا دارند.

یعنی کلمه را align کنیم از روی کاراکتر ببینیم رمیز کلمه متفاوت هستند
graffe — giraffe / grail / graft / graf
کدام کاراکترها مثل هم هستند و باهم match میشوند

insert ① insert شده مثلا اینجا

delete ② delete شده مثلا اینجا

Subsitute/replace ③ جایگزین شده از

نمایش Align می‌تونه حالت رخ دهد

- AGGCTAT

TAGCTBT

AGGCTAT ③

TAG_CTBT

① ②

edit distance ← با alignment استفاده می‌شود

بیشترین match را داشته باشیم

برای alignment هزینه حساب می‌کنیم

هزینه insert/delete ← insert/delete

sub ← دو هزینه ۲

۱ است

اما شبیه‌سازی هزینه نداریم

فقط برای spell correction نیست ← machine translation هم استفاده می‌شود

ترجمه‌ای که ما کردیم با ترجمه‌های متفاوت است

یک ماشین editdistance یک راه بهینه استفاده از روش levenstian distance

↓

base اصلی dynammic programming

۲ رشته با طول m و n داریم

برای edit ← به صورت بازگشتی distance رشته‌های

فکر می‌کنیم

رشته را کوچک و کوچکتر می‌کنیم

$$D(i,j) = \min \begin{cases} D(i-1,j)+1 & \text{insert} \\ D(i,j-1)+1 & \text{delete} \\ D(i-1,j-1)+2 & \text{subbstitution} \end{cases}$$

if $S_1(i) \neq S_2(j)$

0 $S_1(i) = S_2(j)$ match

Elipon

Scanned with CamScanner

minimum edit distance به ما کمک

برای پیدا کردن مسیر درست ← بالاترین خانه درست است جدول

باید backtrace انجام دهیم

باید مشخص کنیم کدوم عملیات انجام شده و بیایم از کدوم خونه بیاییم

عملیات

یه آن خونه

هزینه زمانی $O(nm)$

فاصله $O(nm)$

backtrace $O(n+m)$

contextual information ← اگر چیزی کاندید داریم اونی

را انتخاب کنیم که بیشتر برای این context

source of error ← از کجا خطا تولید شده

Domain of knowledge ← بر اساس کدوم زبان است

داده درست را انتخاب کنیم

common mistake ← یک دیکشنری از خطاهای متداول

---

persian text processing

character encoding ← برای کاراکترها encoding های متفاوت داریم

مهم است ← چنین مشکلی نداشته باشیم

روی بدنه عملی بنویسیم ؟ S vs ک vs ک

S عربی است ← unicode متفاوتی با S دارد

ایجاد تنوع داریم فارسی و عربی ←

ک ۲ ← ۲ ۲ ۳ ، با ک دارد

در فارسی ه و ه و ک و ک داریم

یکی از کارهای مهم پیش پردازش تبدیل سازی است که باید انجام شود

در الفبای فارسی متفاوت از ساختار الفبای unicode است

بعد از همه کلمات عربی در unicode

پس آخر unicode قرار گرفته چون unicode ها از عربی آمده و عربی اولویت دارد

در الفبای عربی ه و برعکس الفبای فارسی است

این روند نمی‌خواهیم sort کنیم ← sort مورد انتظار نیستیم

مثلاً می‌خواهیم عکس بگیریم

علت آخر است ببینیم

مطلوب نیست کاربر عادی

داشتن مشکل داشته باشیم و بعد راه حل درست کنیم

باید تعریف sort مجزا با استفاده از این داشتن تعریف کنیم کلمات بر اساس الفبای فارسی قرار گیرند

* مرتب‌سازی با کلید space را مت قابل تشخیص نیست.

چند اتفاق در زبان فارسی می‌افتد که افراد بعد مشکل‌اند. ← چند حرف فارسی جمع می‌شوند که مقدار کلید disjoint هستند

ماد

پدر

۲ در ذرات و ک نمی‌چسبد

Elipon

Scanned with CamScanner

وقتی موضوعی که جدید نمی‌شدند با توجه به این موضوع خیلی افراد توجه نمی‌کنند که شده تمام شده و باید یک فاصله جدا از آن بی‌آید

ما هر چیم نمی‌خورده ← ملاحظه علی

چسبیده

مطلوبی

ما هر چیم می‌ریزه

چون عبارتی چسبیده خود نشسته احساس می‌کنه این‌ها از هم جدا هستند

و اسپیس نمی‌گذارد (احساس می‌کند به درج space نمی‌شد)

* این مشکل با ناملاحظه وجود داره مشکل ← زمانی که بیم بچسبند و کلمه جدید معنا دارند

{ ما در = ماد
و یا = و یا
اگر بچسبند

* مشکل بیم فاصله داریم مفهوم

← رعایت می‌شوند

① سر فاصله و نیم فاصله مشکل داره

② سبک نگارش مورد استفاده تنوع زیادی داریم

↳ تفاوت محاوره و رسمی زیاد است

↳ اگر در داده train داده رسمی داشته باشیم و بعد در real application داده محاوره‌ای استفاده شود مشکل می‌خوریم

رسمی ← Standard ← اگر

محاوره ← sub-standard ← نه

ادبی ← super-standard ← بر

نوعی فارسی دارد

استایل ادبی که در اشعار دیده می‌شود

Scanned with CamScanner

۳) مشکل خاصیت زبانی داریم

← احتمال تولید واژه جدید توسط کاربر زیاد است

↓

تشبیه ← تقلید شده با اینا می‌تونیم train کنیم مدل سوپرایز شه

۴) تفاوت نوشتاری برای کلمه‌های عربی ← می و مت

۵) homographs and homonyms ← کلمات هم‌جنس

لهجه‌ها نگارش حروف صدادار کوتاه

حرف برای اسامی خاص با حرف بزرگ

شروع می‌کنیم May, may

انجام رفع می‌شود

آذر / آذر / آذر

آتش فصل نام دختر

← اگر حروف بزرگ نداشتیم می‌تونستیم تفکیک کنیم

← چون حروف بزرگ نداریم همه یک جنس هستن

۶) کلمه‌هایی که از زبان عربی آمده است

← از نفست آنها اجتناب می‌کنیم IGNN

می می‌نویسیم جای متی

باید normalization آنها را نرمال کنیم

← که حذف آنها باز ابهام ایجاد می‌شود

جدا و جدا / بنا و بنا

۷) الف اول کلمه ← تنوع داره

Elipon

Scanned with CamScanner

⑨ لغاتی که از زبان انگلیسی آمده‌اند

⑧ کسر اضافه ← فاتحه / فاتحی / فاتحه‌ی

intermidiate ← معدل translate

اینترمیدیت ← برای اولین بار خلق شده

← تنوع سلیقه داریم

← یک یک حرف ترجمه کنیم

translitrate

⑩ واژگانی که فراتر از یک واژه معنا دارند

← این چه چیزی است ← این چه

او کیست ← او که

یک واژه‌ی کاملا بی‌معنی مستقل می‌شد

که از او ← نزد

آیا این بچه تو است ← بچه

Sequence processing Recurrent Neural Network

پردازش دنباله‌ها

① فعالیت‌های مبتنی بر مدل‌سازی

متن ← رشته و دنباله‌ای است

کاربردها:

- مدل‌های زبانی
- تشخیص موضوع کلمات
- برچسب اجزای کلام (part of speech)
- اصلی خاص

تابع مهم ایجاد می‌شود و بین واژه‌ها ارتباط داریم.

حساسیت دنباله‌ای داده

با توجه به اینکه دنباله داریم می‌توانیم بگوییم تشخیص دهیم یه کلمه با چه احتمالی قبل اسم یا فعل می‌آید و بعد متوجه بشیم اون کلمه فعل یا اسم است.

to race tommorow

v or n?

برای تشخیص و انتخاب

۳ احتمال است

۱) to قبل می‌آید VB یا NN

۲) NR قبل می‌آید VB یا NN

۳) با چه احتمالی فعلش v یا n است؟

کل رشته مهم است

روش‌های متداولی برای sequence modeling استفاده می‌شود

① Hidden markov model

n کلمه ورودی
یا tag ها

t_i → از جمله

tag هایی که داریم

برای یک زنجیره n کلمه‌ای → یک زنجیره n تایی از tag ها می‌خواهیم

زنجیره‌ای از tag هایی رو انتخاب می‌کنیم که احتمالش برای ورودی‌ها بیشتر است

$$\hat{t}_1^n = \arg\max P(t_1^n \mid w_1^n)$$

از فرمول بایز استفاده می‌کنیم

$$P(t_1^n \mid w_1^n) = \frac{P(w_1^n \mid t_1^n) \cdot P(t_1^n)}{\cancel{P(w_1^n)}}$$

چون ورودی ثابت است

یعنی احتمال ۱ دارند و داریم tag انتخاب می‌کنیم خروجی ۱ است

Elipon

Scanned with CamScanner

$$P(t_i^n) = \prod_{i=1}^{n} \underbrace{P(t_i | t_{i-1})}$$

احتمال هر tag با شرط tag قبلی اش

$$\frac{C(t_{i-1}, t_i)}{C(t_{i-1})}$$

تعداد دفعات

$$P(w_i^n | t_i^n) = \prod_{i=1}^{n} \underbrace{P(w_i | t_i)}$$

$$\frac{C(t_i, w_i)}{C(t_i)}$$

تعداد دفعات

Recurrent Neural Network (RNNS)

زمانی یک ورودی را به عنوان دنباله ورودی پردازش کنیم

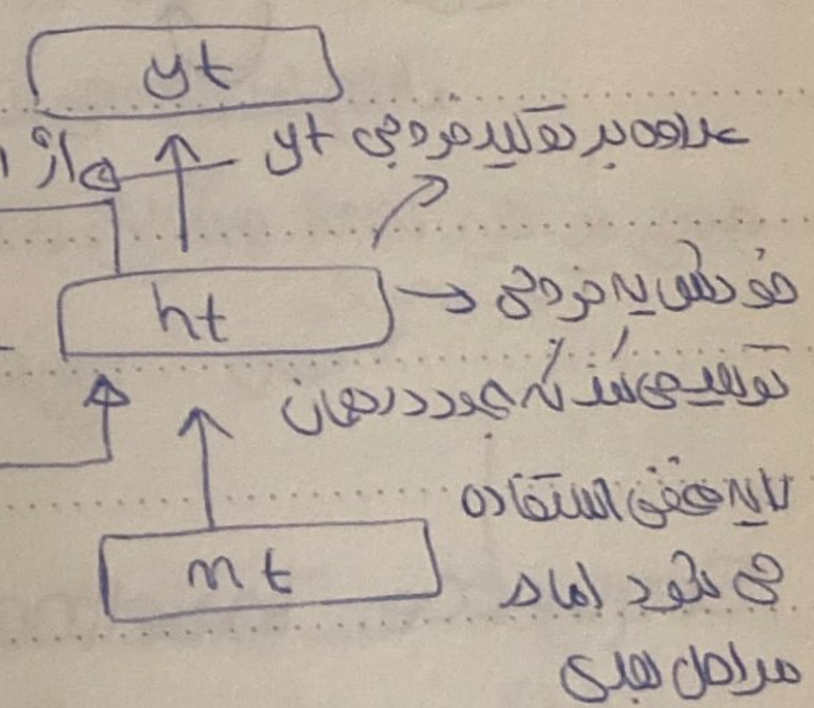
شبیه RNN

حافظه دارند و اطلاعات را می‌دهند به بعدی شده و در هر مرحله پردازش به اطلاعات قبلی نگاه شده.

ورودی ما دنباله دار پردازش می‌شود و جلو می‌رود

اطلاعات آن مرحله و مرحله قبل را داریم.

در واقعیت کرده هستیم به شکل خوب



علاوه بر تولید خروجی yt از اطلاعات لایه مخفی step قبلی استفاده می‌شد + ورودی

خروجی لایه مخفی تولید می‌شد که مجدد در همان لایه مخفی استفاده می‌شود اما مراحل بعدی

$$h_t = g(\overset{\text{ضریب}}{U} h_{t-1} + W m_t)$$

$$y_t = f(V h_t)$$

مثلا softmax

کاربردهای RNN : language model (1)

sequence labeling (2)

text generation (3)

text classification (4)

language model → n-gram models
→ fully connected neural network
→ RNN

یک چیز رو روی ثابت داریم (حجم ثابت از گذشته)

بخش ثابت نداریم ← می‌توانیم از گذشته دورتر استفاده کنیم

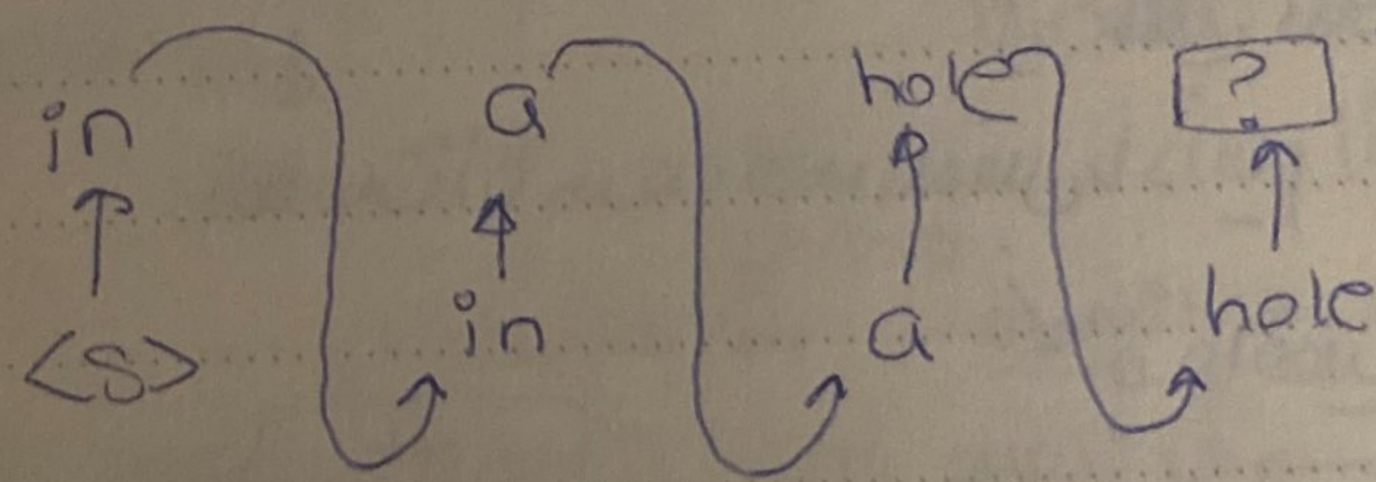
(کلمات) ورودی می‌دهیم ← خروجی تولید می‌شود (کلمه بعدی)

← با خروجی مورد انتظار مقایسه می‌شود

cost حساب می‌شود.

خروجی توسط شبکه عنوان ورودی بعدی داده می‌شود

text generation ← هدف: با یک متن شروع کنیم و در هر لحظه واژه بعدی را پیش‌بینی کند و بعد واژه قبلی به عنوان ورودی می‌دهیم و الگوهای بعدی را پیش‌بینی می‌کنیم



autoregressive generation

sequence labeling ← pos taging

یک رشته ورودی داریم می‌خواهیم برای تک تک کلمات یک tag را حساب کنیم

tag پیش‌بینی ← مقایسه و بدست آوردن cost و استفاده از آن برای بهتر کردن مدل

← پیش‌بینی دقیق‌تر برای خروجی‌های بعدی

کلمه

text classification

طبقه‌بندی متون

در این مدل‌ها اثاث هر داده ضرب می‌شد

شما اطلاعاتی که مستقل می‌شد

$h_{t-1} \rightarrow h_t$

$h_t \rightarrow h_{t+1}$

} محاسبه hidden های هر مرحله

و ما فرض می‌کنیم از آخرین سلول می‌گیریم

تجمیع از اطلاعات تمام است

به عنوان ورودی به softmax دهیم

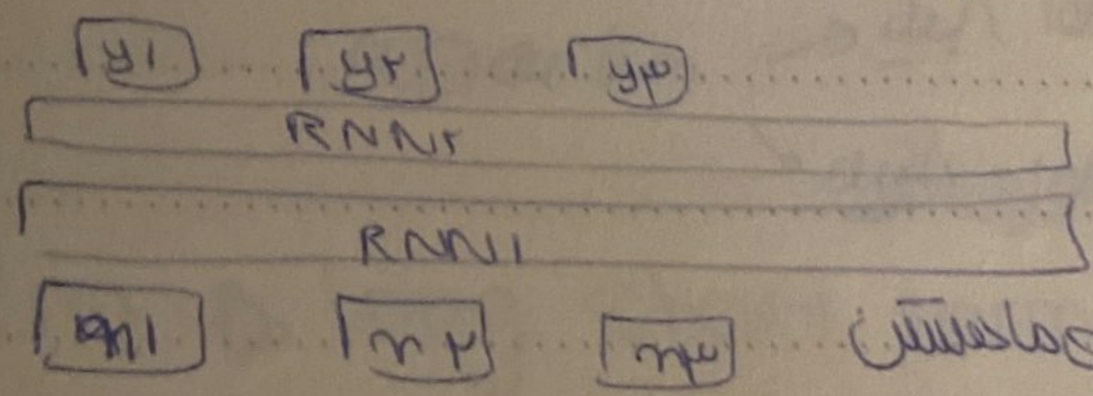
و محاسبه می‌کند یا طبقه‌بندی کرده

Stacked and bidirectional RNNs

stacked RNN از چندین لایه پشت هم RNN

جای یک لایه

داریم استفاده می‌کنیم



خروجی یک لایه RNN می‌گیریم به بعدی می‌دهیم

همان hidden layer های ماست

بیشتر شدن اطلاعات بیشتری را استخراج کنیم از متن

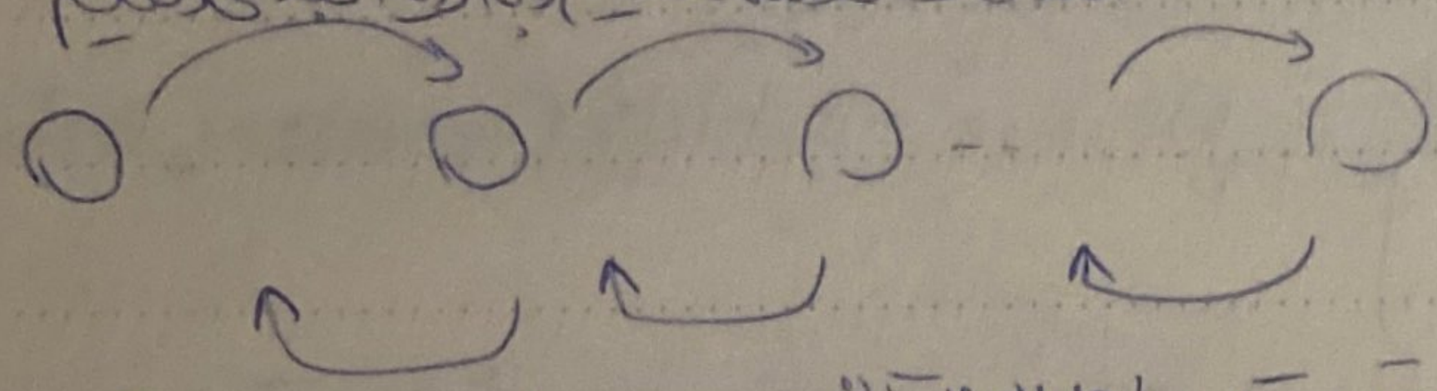
مثلاً دیدار از یک دستور ← ۱) اول کلمه‌ها را شناسایی می‌کنیم

۲) اشیا

۳) هدف از اشیا را جداگانه

Bidirectional RNNS

اطلاعات گذشته داریم و به لایه بعد می‌دهیم



یعنی اومدن اطلاعات آینده هم داریم و به لایه قبل می‌دهیم.

مثلاً وسط یک متن جای خالی داریم.

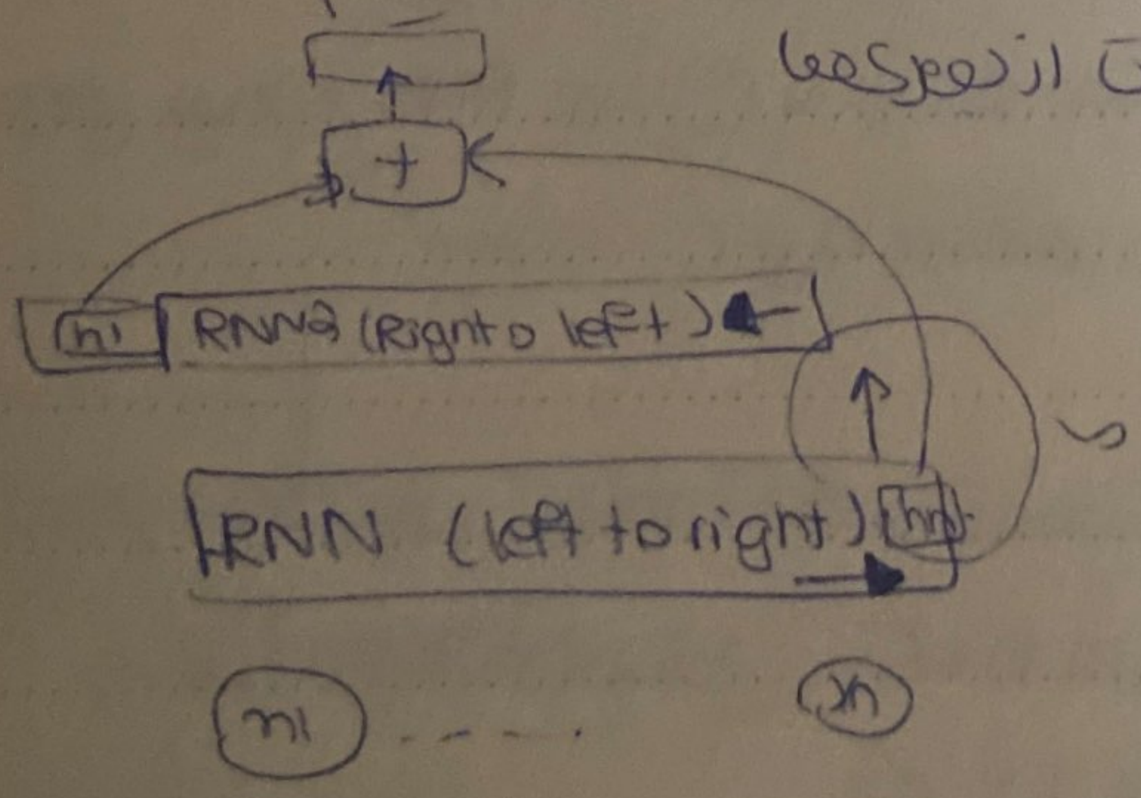
پس برای پر کردن جای خالی هم از کلمات گذشته هم کلمات آینده استفاده می‌کنیم

* این قرار است سختی انجام دهیم و اطلاعات آینده را داریم بیاییم ازش استفاده کنیم

برای هر لایه ← یکبار اطلاعات از قبلی‌ها

← یکبار اطلاعات از بعدی‌ها

استفاده کنیم



text classification

در حین آموزش داریم

LSTM & GRU

۱) RNN می‌تواند گذشته را ببیند ← اما در دنیای واقعی اینطور نیست و گذشته محدود داریم

The flights the airline was cancelling were full

فاعل این flights است و اگر flights فراموش شده باشد

علت فاش was بگذاریم

وابستگی به واژه‌هایی که دورهستند ← فقط context دورتر باشد RNN راحت‌تر

۲) در حین train شبکه ← برای محاسبه gradiant نیاز داریم

اگر مشتق‌ها کوچکتر از ۱ باشند با ضرب متوالی ترم‌ها

می‌شوند و به این اتفاق vanishing gradient گفته می‌شود

برعکس vanish هم همینطور رخ می‌دهد

برای جلوگیری از مشکلات فوق شبکه‌هایی معرفی شد

LSTM ← long short term memory (LSTM)

همان ساختار زنجیره‌ای داریم

اما داخل هر سلول ۴ تا اتصال منتقل می‌شود (سلول ساده‌تر RNN)

۳ تا gate وجود دارد

۱) input gate
۲) forget gate
۳) output gate

۲) تصمیم می‌گیرد اطلاعاتی که از گذشته می‌آید چقدر به این سلول مرتبط است و مفید است

۱) تصمیم می‌گیرد input که می‌آید چقدر می‌تواند مفید باشد

۳) خروجی از سلول منتقل می‌شود چقدر از اطلاعات منتقل بشود

Scanned with CamScanner

عملیات input ، forget ، output داریم.

Gated Recurrent units (GRU)

۲ تا گیت داریم ← reset و update

مقدار اطلاعات را تنظیم داریم

می‌خواهیم اضافه کنیم

مستقل کنیم

بسته به task باید انتخاب کنیم LSTM یا GRU کدام انتخاب کنیم:

در کار بیشتر

در تمرین در POS شناسایی به جای استفاده از RNN از LSTM استفاده کنیم

و یا حتی جای biRNN از biLSTM استفاده کنیم

---

transformers

encoder-decoder ← پایه اصلی تمام مدل‌های RNN هستند

برای تهیه از ساخت RNN استفاده کنیم چه در ورودی بدهیم چه در خروجی predict کنیم

همان ترجمه باشد.

target در <s> علامت

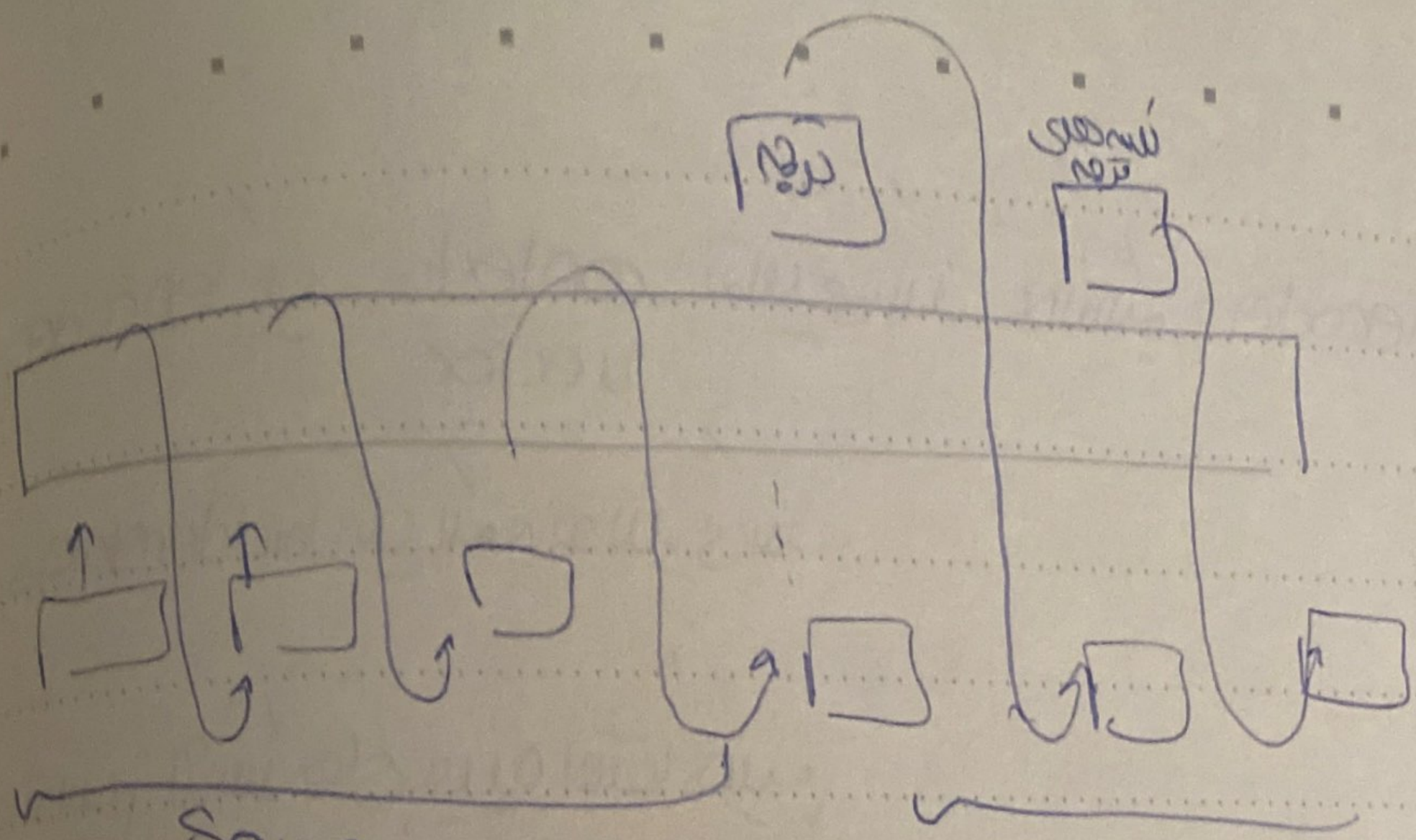
تهیه

encoder که همین ساخت S داریم هفته‌ها از ۲ بخش تشکیل شده یک بخش

یک بخش decoder

Elipon

Scanned with CamScanner



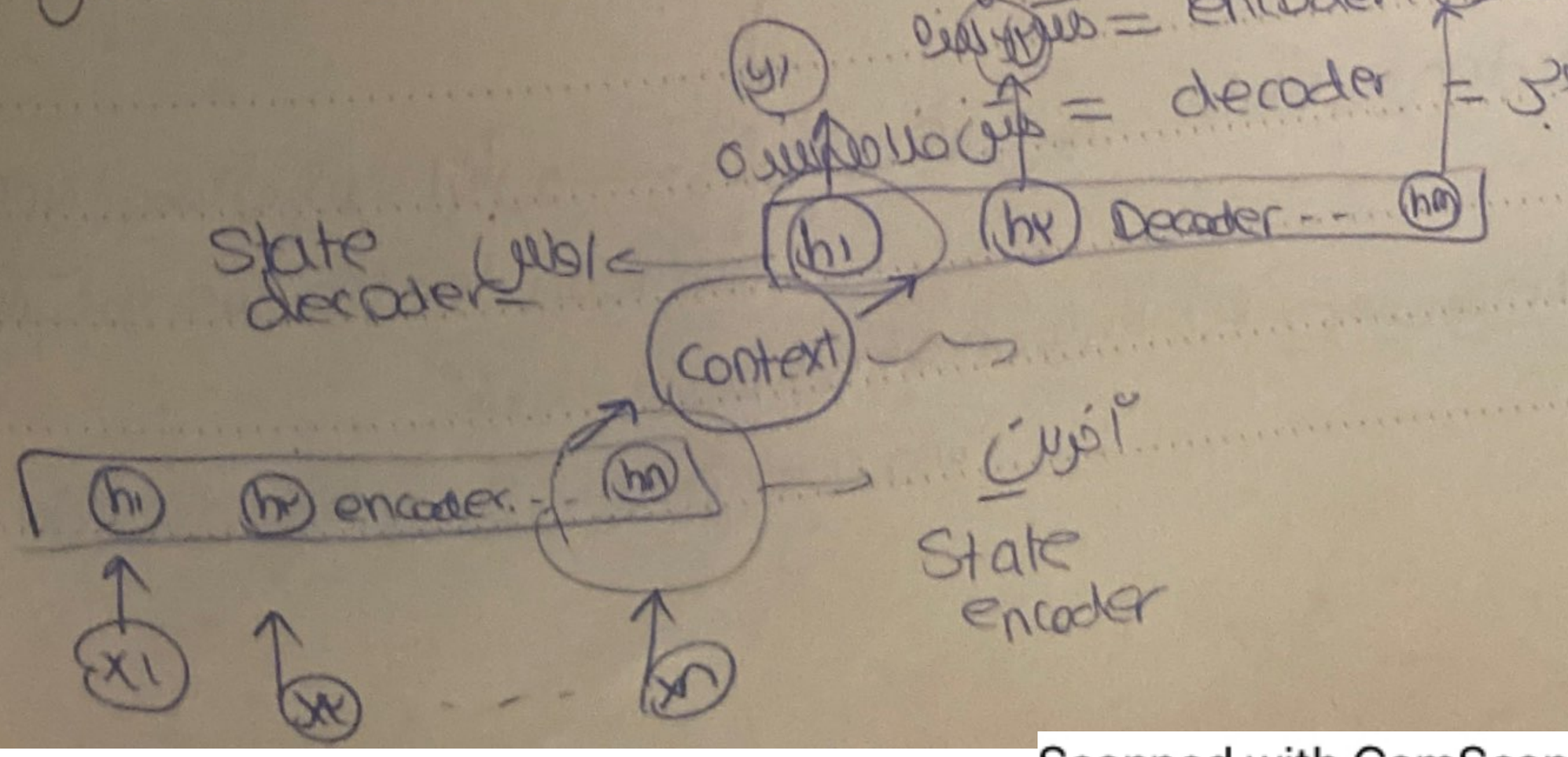
Source = encoder target = decoder

جمله انگلیسی میشه و پردازش میشه decode میشه فرانسوی برای ما جمله

generates a contextualized representation of the input

Decoder takes that state and autoregressively generates a sequence of outputs

در مدل‌ها فقط با کلمه استفاده نمی‌شود طبیعی معنوی دارد — عددی خلاصه‌سازی شدن



* یک encoder ← context vector تولید می‌کند به شبکه decoder داده می‌شود

* در هر hiddenstate یک کلمه تولید می‌کند.

* می‌توانیم از شبکه‌های پیچیده استفاده کنیم ← مثل stack bilstm

* می‌توانیم پیشرفته کنیم ← به جای اینکه context vector تنها به decoder h_0 داده شود بلکه state های decoder داده شود تا بتواند اطلاعات بیشتری داشته باشد.

* می‌توانیم از توابع مختلف استفاده کنیم ← softmax = generation ✓

~~summarization~~ MT

اندازه‌ی هد ذهن‌های مختلف برای قسمت‌های مختلف در نظر بگیریم ← مهمترین بخش SO SO

* مکانیزم attention:

context بردار ← flexible شویم

چنین بردار context در نظر بگیریم. برای هر state encoder می‌گیریم

به صورت dynamic بردار context ها تغییر یابند

وزن‌های عددی به state های encoder داده شود ← توجه به اطلاعات بیشتر و ارتباط

در اسپیک‌های مختلف decoder استفاده کنیم

مثلا برای وزن فاصل encoder نیاز داشته باشیم پس میریم سراغ context vector

این و این context وزن همنوعش خودش داده

decode

* مکانیزم توجه استفاده می‌شود ← چون در هر مرحله به بخش خاصی از ورودی که وارد encoder می‌شود توجه می‌کنیم

transformers : self Attention

مشکلات RNN های پیشرفته ← LSTM و GRU قبلی کمک کردن مشکل گذشته رو بهتر میشه
اما باز هم این مشکل وجود دارد ← چون زنجیره مرحله به مرحله پردازش می‌کند

پردازش RNN ← زنجیره مرحله به مرحله و پیوسته ← نتونیم کلمه به کلمه

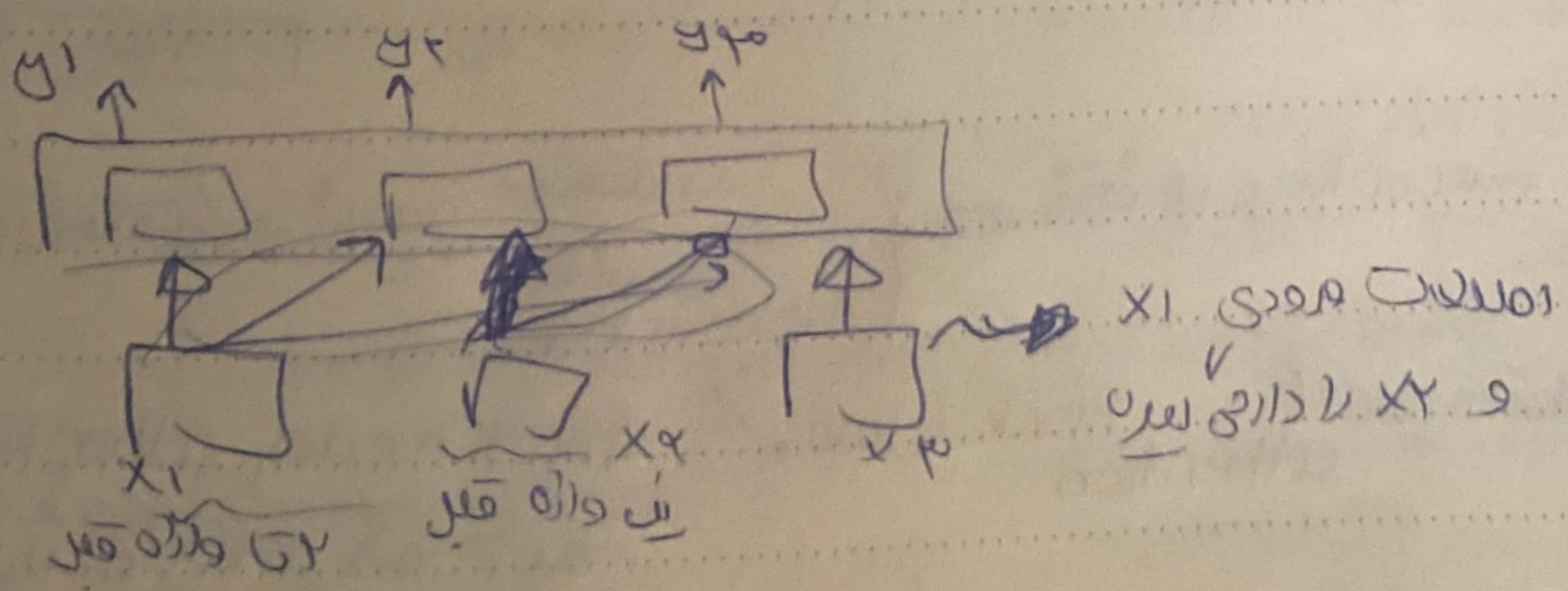
موازی پیش ببریم

feed forward + لایه‌هایی که جدید یاد گرفتیم

قرار هست برای هر واژه که قرار است پردازش کنیم نگاه کنیم
به واژه‌های قبلی ← احتمالاً یک لایه‌ی اضافی

t-1 - - - ۱

این فرآیند را می‌کاهیم به صورت موازی انجام دهیم



chapter 9 کتاب

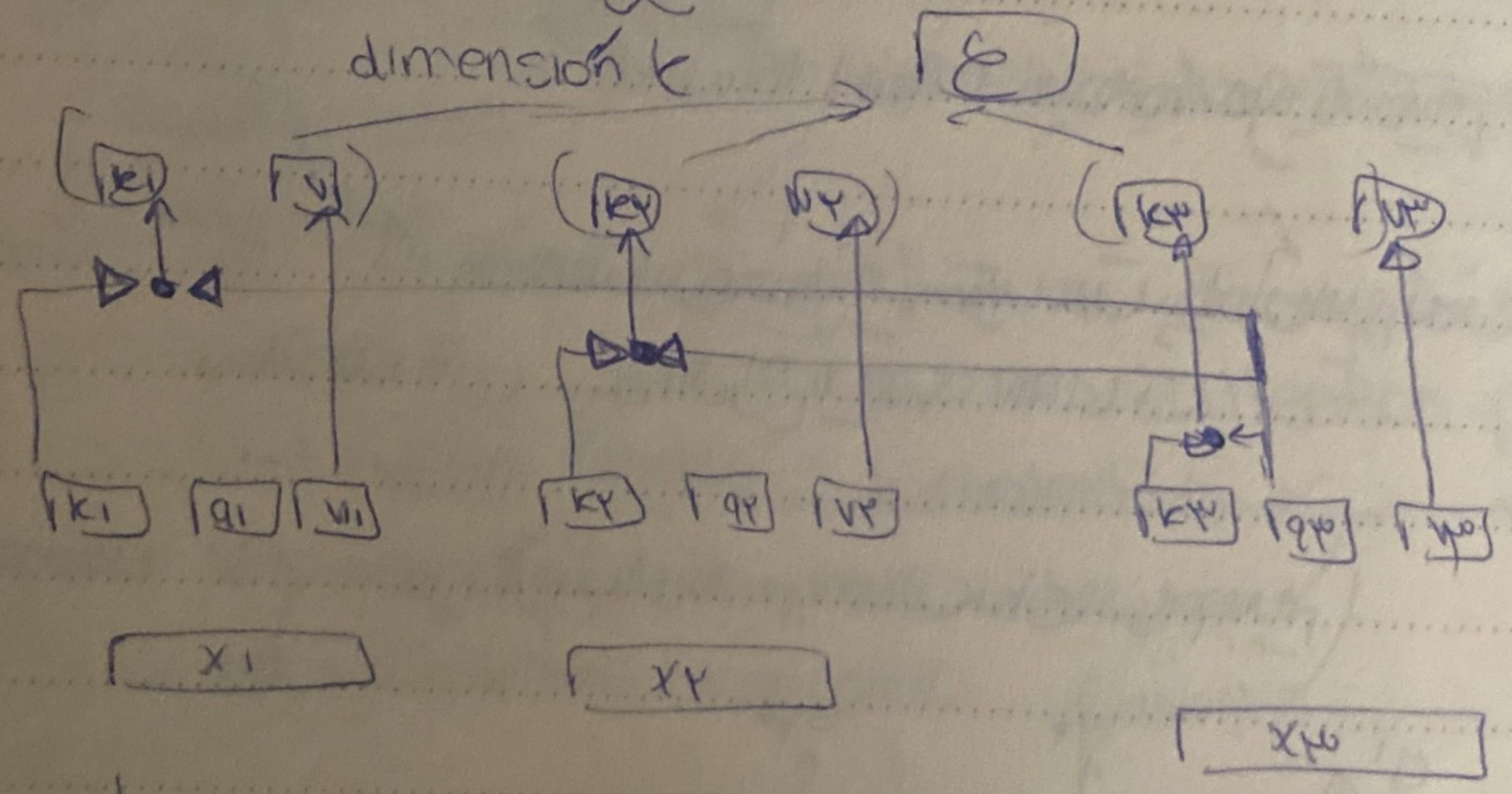
* حاصل y به این صورت است یک score بین هر دو مورد خود و state دیگر می‌گیریم $score(x_i, x_j)$ و تقسیم بر تعداد هر دو های دارد بر آن state می‌کنیم.
قبل

* پارامتر برای آموزش ندارد
← برای آموزش نیاز می‌توانیم بردار دیگر بگیریم

q و k و v
← بردارهایی که در طول train آموزش داده می‌شوند
درخواست می‌دهد به بردارهای k قبلی

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

dimension k



اصلا کاستی‌های ظاهر
{ معنایی
نوشته
discourse
می‌شود capture کنیم

multihead attention: چیستن block self attention استفاده کنیم
که هر کدوم یه کاری انجام بده

Elipon

Scanned with CamScanner

Dialogue systems

= conversational agents

text understanding

شامل بحث speech هم باشد ← speech به متن ← متن را درک کنیم

یا برعکس ← پاسخ متن را به speech و صدا می‌کنیم

conversation agent ← برای ایده بهتری ← برای ماشین کلی agent

general

داشته باشیم ← بدون این هدف خاص داشته باشیم مثلاً زدن متن و ...

↓ chat_bot

زمانی که برای هدف خاص agent داشته باشیم ← tasked_based

= goal_based

مثلاً play music و reservation timer

chat-bot ها را با taske-based استفاده می‌کنیم.

tasked base به general استفاده می‌کنیم

* معماری بحث tasked_based و conversational دست متفاوت است.

بر اساس frame هستند

می‌خواهد هدف (intension) را استخراج کند

یا بر اساس retrival (بازیابی اطلاعات)

یا بر اساس generation (تولید متن)

مجموعه‌ای از slot ها داریم

frame هدف می‌کنیم (معمولاً هدف و هدف خاصی دارد)

مثلاً میدا، مقصد، هتل، پرواز

از پیش slot برای انجام آن frame لیست می‌کنیم

تایپ‌های entity یا type خاص برای

چکش slot ها نیاز داریم

هر کدام از آن slot ها می‌تواند با یک سوال خاص یا سمتش پیدا شود و در آن slot قرار گیرد.

conversational chatbot

rule based ← تاریخچه و گذشته ← مشابه دانش بنیان

corpus-base

هنوز از مدل‌های rule-base برای بهبود مدل‌های corpus استفاده می‌کنیم که مدل‌های روش‌های ترکیبی hybrid

chatbot

eliza (اولین) ← action patten rule

parry ← دوستانه

turin هست پایه‌گذار

بر مبنای information retrival (xiaolce)

بر مبنای neural encoder-decoder که می‌شد generation

(BlenderBot)

ویژگی‌های مکالمه انسانی

turn ① ← هدف متنوع داریم

← تشخیص دهیم کی نوبت Client و کی نوبت Agent

کی هست که صحبت کند ← می‌شه کوتاه یا بلند داشته باشیم

برای سیستم همه بیفته ← تشخیص دهیم مدام انتظار چقدر صحبت داریم

Elipon

Scanned with CamScanner

end pointing Barge in ← فقط شنونده

وسط حرف

(نه زودتر نه دیرتر)

باید تشخیص دهیم ← ① کجا نوبت بی صدا است که صحبت کنه

② چند صحبت کنیم و کجا اجازه دهیم طرف مقابل صحبت کنه

① speech act ← constatives : در قالب بیانیه گویی هستند (جمله خبری)

② directives : جملات دستوری هستند یا سوالی

③ commissives : از نوع برنامه ریزی و قول دادن

④ Acknowlegments : تشکر و عذرخواهی

③ grounding ← تاثیری که تو مکالمه بین دو انسان اتفاق می افتد.

مکالمه زمانی برای دو انسان مفید است که به یک common ground برسند

حواس انسان دو دهنه است

انسان می دهد آنسند

متوجه خواست شنونده

باعث می شود حس مطلب دهد.

نزدیک بودن به معنای قبول کردن نیست به معنای درک حرف است

باعث کاهش شورانی می شود

④ subdialogues ← درون یک مکالمه یک پروتکل بازی می شود یه بحث کوتاه رخ می دهد و مشخص می شود و دایالوگ ادامه می دهیم

به بحث هایی که فعلی در مسیر دایالوگ اصلی نیستند ← subdialogue گفته می شود

مثلا تو بپرسی فلانی روزه بپرسی راستی ها الان چه ساعتی است؟

⑤ initative ← مکالمه مشخص هست ← اولی سوال ← دومی جواب

دومی سوال ← اولی جواب

اگر فقط یک سوال بپرسه یک جواب مکالمه یک سوء است

④ inference ← استنتاج ← کاربر میتواند ۱۹ تا ۱۹ اونجا باشیم
ماشین باید استنتاج کند تا بتواند ۱۱ ام بلیط بگیرد

Rule-based

pattern matching
↑
ELIZA ← مثل یک روانشناس رفتار میکند

جمله یا سوال از صحبت های کاربر تقلید میکند

کاربر: you hate me

ماشین: what makes you think I hate you?

سوال از کاربر ایجاد میشود.

کاربر احساس راحتی بودن ← شروع دادن نوع سوال ها ← نمی کند

keyword : you pattern : 0 you 0 me

Transforms : سوالات ایجاد میشود

اصلاحات جمله ای تقلید نمی کند صرفاً نحو بیان را متفاوت می کند

statement تقلید می کند.

* می توان جمله تقلید را سری کرد مثل please go on سعی می کند مکالمه را ادامه دهد

* روش سری کلمات keyword ها مساوی است

PARRY ← برای مکالمه با افراد schizophrenia

۰-۱۵ ~ ۰-۲۰

← میزان اندازه گیری و score شده ← ترس/عصبانیت/عدم اعتماد

← سعی کرده پاسخی نمی داده مخالف به نوع سوال که به حس ها می گیرد باشد

← پاس کرد turin test

Elipon

Scanned with CamScanner

corpus based retrival: بازیابی متن مبتنی بر کلیدواژه ← سیستم باید بفهمد کدام پیشوند یا پسوند پاسخ مناسب برای سوال را بدست آورد

generation: متن جدید تولید می‌شود

* زمانی که با کاربر صحبت می‌کنیم یک پیغام می‌فرستیم اما نباید جستجوها فقط بر اساس آن پیغام باشد باید کلی مشخصه در نظر بگیریم

context ← جملات قبلی

* نیاز به داده‌های زیاد برای مدل‌ها

مدل‌ها ← train دارند ← فقط جستجو مبتنی نیست ← تولید متن هم هست

امتیاز یا فیت پیش می‌شود

transcript تلفن‌ها

subtitle فیلم‌ها

hire human ← crowdworkers ← معمولاً چون کمتر از مدل‌ها است

train نمی‌کنیم. روی تست test انجام می‌دهیم

pre-training انجام می‌دهیم

یک مدل اولیه تمیز کنیم و بعد برای هدف خاص خودمون با استفاده از داده‌های pre train شروع آموزش دهیم که متن‌های مطابق خودمان تولید بشود (شبکه آموزش اولیه داده ← روی داده‌هایی که تعریف کرده هدف هستند

بعد باید آموزش از صفر شروع کنیم)

بعد روی داده‌های هدف test انجام می‌دهیم

twitter / redis

همزمان پیش می‌رود

Retrival method:

یک کوئری (متن) و دنبال یک پاسخ (از سمت سیستم) هستیم

پیدا کردن یک متن که نزدیک به متن سوال باشد

بردار query و بردار re space

مدل نزدیکی ← ① TF - iDF ← شباهت متن با فاصله cos

partphrases ← متن چت باشد

② Neural text matching

شباهت بردار روی یا یک تا بردارهای response بدست میاریم و هر کدام بیشتر فاصله دارد انتخاب می‌کند.

زمانی که می‌خواهیم از vedorspace استفاده کنیم یک سری نکات هست که باید بدانیم

۱ شمارش کلمات ← چه کلمه در چه متنی میاد

۲ مهم بودن کلمات مهم ← کدام کلمه معنی بیشتر و مهم‌تر

term-frequency

نباید انتخاب واسیقه به طول جمله جواب باشد

تعداد تکرار یک واژه درون یک متن

باید جایی بدون تکرار انتخاب شد

اهمیت روی بردار تو وما اثر داشته باشد کلمات دیگر داشته باشد.

document freq ← بر اساس متن جمله شدیم چقدر یک واژه مهم درون یک متن است

یا خاص

idf

Elipon

Scanned with CamScanner

Normalize document → نرمال‌سازی کوتاه یک معنی طولانی باشد

اتصال کلمات مشترک با query بیشتر

باز یکی مستقل از شما بودار باشد.

لماتیزه‌ها بدون فای انتخاب بهتر هستند

در فاصله کمی فرم normalize انجام می‌شود

واسبته به واژه است. ← فوزو و اتومورفایزر تقاند

sumatch

می‌رویم تو روش‌های embedding

word2vec/glove

بردار برای متن ← میانگین‌گیری از بردار word2vec

میانگین وزن‌دار بر اساس tf-idf ← وزن‌دار tf-idf آن کلمه‌ها

doc2vec ← در کنار برای هر واژه embedding می‌سازد

یک embeding برای کل سند هم می‌سازد

با استفاده از فاصله cos شباهت پیدا می‌کنیم

Neural text matching

یک نوشته داریم ← مقایسه با یک متن نمونه دیگر response ← بر اساس یک metric که شباهت را اندازه‌گیری می‌کند حساب می‌کنیم

اون متریک قبلا tf-idf بود است

الان بر اساس Neural text matching

* در learning داریم انجام می‌دهیم.

آموزش نظارت‌شده ← نیاز به داده آموزشی داریم
برای پیدا کردن شباهت دو متن

پیکره از پاسخ و سوال داریم

زوج داریم.

pair را
به عنوان داده آموزشی می‌بریم

* representation-based model ①

interaction based model ②

hybrid model ③

با داده‌های کلاس learning انجام می‌دهد

دوتا شبکه train می‌کند برای pair پاسخ و سوال

کمترین representation را تعریف هدف

مشخص کند

مشخص دهد چه پاسخ چی است → positive sample

علت بی‌ربط کردن پاسخ‌ها هستند

هم‌زمان این شبکه‌ها تو یکم بسته بیاوریم

و به عنوان negative sample بشناسیم.

شبکه یادگیری representation برای کلاس‌ها

مقایسه کند

در اساس داده‌های آموزشی

هر مقایسه‌ای که این داده‌ها هستند بهتری ارجحیت

Representation-based Models

یک representation برای همه داده‌ها می‌کند

Q D1 D2 ... Dn → rep

کوئری کاربر

داکیومنت‌هایی که هستند جواب کاربر باشند

repse کوئری را یک تک

doc ها مقایسه می‌کند

می‌تواند cos باشد.

Elipon

Scanned with CamScanner

Feed forward یا شبکه‌ها

representation آموزش می‌دهیم.

با شبکه عصبی بدست می‌آوریم ← بعد تقاضا می‌دهیم

یا فاصله cos

داده آموزشی که داریم

شبکه‌های feed forward داریم.

۲) interation base ← تعامل دوسویه یک به یک بین واژه‌های متن‌ها تولید می‌شد

بین یک یک واژه‌ها تعامل داریم

ماتریس query و ← match pyramid ← بعد شبکه بندی انجام می‌شود

response ← هر آیتم تعامل کلمه i متن اول و j متن دوم که به هم مرتبط می‌شوند

می‌توانیم lexical match با اگر بیشتر مساوی باشد

اگر بیشتر نباشد

decimal match

بردار embeding بدست بیاید

بررسی می‌شباهت

matchingscore

بعد از تولید این ماتریس ها آن وارد شبکه می‌شود و یک به ما می‌دهد

۳) Hybrid ← بین representation دست می‌شد اما قبل اینکه به شبکه‌ها در تعامل بیشتر قرار می‌دهند.

هم موازی

هم pipeline

Generation Methods ← هدف تولید متن

یک جمله از کاربر میگیریم و یک response تولید کنیم

همه کلمات در یک جمله و قبلا کلا یادگرفتیم

RNN ← encoder ← decoder

سوال ورودی — فرومی بدید

فقط برداری میانی

context-vector

text translation

و نهایتا با attention تقویت شود

با داده کاربر آموزش داده میشود (pair مکالمه)

و در کاربر فرومی به سیستم

* Retrival میتواند NN باشد

dotproduct

فقط encoder داریم کمتر و دقیق را

به یک بردار تبدیل کنیم و بعد مقایسه انجام دهیم

encoder — query — $q_1 \dots q_n$

encoder — response — $r_1 \dots r_m$

* Generation

r_1 r_2

فرومی توسط فرومیشه تولید

decoder میشود decode و

متن ورودی encoder $q_1 \dots q_n$

$r_1 \dots$

Elipon

Scanned with CamScanner

فضاهای تولیدی داشته باشیم

Generation base ← در زمان runtime نیاز به پیشینه ندارد و سریع‌تر است و از پیشینه‌ها برای آموزش استفاده کرده است.

response های تولید پیشینه‌ها انجام دهند retrival ها اما باید با آن response

از صحت را می‌دریم

خلاقیت نداریم ← اگر سوالی پرسیده شد که اصلاً نبوده

جوابی برایش نیست

* از یک سری اطلاعات فراتر از corpus استفاده کنیم.

از منابع دیگر استفاده کنیم

wikipedia

تبدیل کردن به سیستم‌های قوی‌تر

پاسخ مطالب از خارج بیاید

* مدل‌های

retrival vs generation ← hybrid

rulebased vs corpus based

retrival & generation

* معمولاً برای تعریف مدل‌های مناسب دستیار ← دقیق متوجه نمی‌شود

صرفاً دوری می‌گیرند و

تجزیه و تحلیل نمی‌کند.

هدف اصلی ← رسیدن به چت‌بات که رسیدن به frame base را دارد.

Scanned with CamScanner

contextualized Representation I

frame base model ⇝ representation model

sparse — Dense

word document — word word

Neural language model ⇝ prediction + معنا

واژه بر اساس بافت واژه معنا می‌کنیم

word2vec, glove ⇝ stastic word embedding

کاربرد

Contextualized representation

ماهیت دینامیک

مشکل:

اما یک سری واژه‌ها مثل bank دو معنی دارند ← با معنای

① معنی ساحل

② بانک و رودخانه

mouse

دارای یک پرچم داشته

← با نمایش دینامیک نیاز داریم ← وابسته به همان متنی که واژه قرار است

ELMO ⇝ bidirectional language model

دو جهته دارد

backward ← راست به چپ

forward ← چپ به راست

دهمل زبانی

آموزش می‌دهد

دو concat می‌کند و با نمایش نمایی به دست می‌آوریم.

Scanned with CamScanner

BERT → 2018 → transformer
↓
multihead attention

feed forward → Bert vs Elmo = transformer vs lstm
self attention

لایه‌ها را از ۱ شروع می‌کنیم
elmo نه در لایه‌های

semi-supervised → pre-train → label سازی

label ندارم

wikipedia کتاب‌ها

BERT → large (24 NN)
→ base (12 NN)

masked language model / train
Next sentence prediction

First token
وارد می‌کنیم <CLS> ← برای پیاده‌سازی نیاز داریم بدهیم می‌ورودی دارم

<SEP> seperator
Delimiter token برای جدا کردن

<mask> برای کلماتی که حذف شده و یا خالی

دیگر supervise نیستیم

برای آموزش مدل BERT برای پیش‌بینی کلمات

* Mask language model

کار مدل را می‌پوشانیم و خود پیش‌بینی کنیم mask دارد.

اگر خیلی هدار کم باشد ← هزینه train زیاد

اگر خیلی زیاد ← واژه‌ای از جمله نمی‌ماند و باید predict کند

یعنی وقتی mask شده می‌شه جاش (۸۰٪)

یعنی وقتی با یک واژه random قرار می‌دیم جاش (۱۰٪)

یعنی وقتی همان واژه استفاده می‌شه (۱۰٪)

* Next sentance prediction

ارتباط ← pretrain

semi_structure

دو جمله متوالی برای آموزش در نظر می‌گیریم

یک سری جملات مثل را به دو به عنوان label به

شبکه می‌دهیم یکی هم

یعنی جملات رندوم دوتایی هم می‌دهیم.

* Hybrid

(NSP) هم label ارتباط دو جمله predict شده

(MLM) هم توکن‌های mask شده predict می‌شه.

Elipon

Scanned with CamScanner

BERT هست که representation مناسب

* از بین لایه‌هایی که
بما می‌دهد

Fine tuning که همان

* اگر برای متن خاص استفاده کنیم بهتره با داده‌های همان
انجام شد ← train ←
semi-structured Pretrain

question-ask. زوج سوال و جواب را بدهیم.
classification یک متن بگیریم
paraphrasing یک متن داریم ← فرض label داریم ←
pretrain ← داده‌های کلی ← MLM
NSP
Bert
train ← fine tuning روی داده‌های همان متن

BERT FAMily ← ROBERTA / ----

① هدف اصلی: داده بیشتر train
② time epoch
← نتیجه بهتر

next sentence ← حذف شده بود prediction

XLNet ← permutation

لایه‌ات دستشان باشد
نه فقط تو کانتکست ← فقط به لایه‌ات absolute attention

relative attention

contex نوع

به کلمه قبل ← هر چیزی وقت بشه

permutation language Modeling →

به صورت رندم جایگشت در جایگاه‌ها می‌بینیم

موقع آموزش سیستم

ALBERT ~

① Factorization

embedding layer ← مدل کوچکتر از BERT

سیستم مدل

RBO1 application

Distilation ~ عملکرد یکی کمتر

state of art

مدل معلم روی

مدل شاگرد ← مستقل شده روی شاگرد

تمام پارامترها با لایه‌ها داشته باشد

طولش با لایه داشته باشد

در حین کوچک بودن مدل شاگرد طولش زیاد

train + سرعت ← inference

specialization → کلاس train

برای کاربرد خاص مدلی را

XtiBert

Elipon

Scanned with CamScanner

↑generative pre train

GPT → GRT ، BERT ، Elmo

text generative task → برای تسک هایی که text generation هستند مفید است

Sampling کافی / Sampling دارد

---

rule base → corpus base → contextualized prepresentation

GPT → generation

هر کلمه نمایه دهش که توسط قرار گرفتن با زمینه ای می شود (dynamic)

generation methods ↓ retrival methods →

از متدهای موجود جدیدترین پایه

GPT

word2vec

autoregressi

Bert → next sentence prediction

Sembeding → Token/position/segmment

→ speaker embeding

اضافه شده

Frame_based dialogue system

↓

(GUS) : ① مدل‌هایی که task خاصی رو پیاده‌سازی می‌کنم

↓

هیچ نیازی به اطلاعات خاصی نیست

* GUS ← Simple Frame based
* Dialogue state tracker

* Frame based dialogue Agent

↓

tasked based ← شبیه زدن مزید عمل

↓

ما یک سری مقادیر مشخص اولیه داریم برای هر تسک و اینا باید مشخص میشود

Frame رزرو پرواز

Slot‌های اصلی برای

① مبدا

② مقصد

③ دیل هواپیما

در قالب سوال‌ها این Slot‌ها پر شده

که مقادیر Slot‌ها باید بدونیم از چه جنسی هستند

Scanned with CamScanner

GUS ← ۴۵ سال

① control structure ← frame یا slot

کنترل روی تمام

slot ها می‌تواند تعریف شده باشند

می‌تواند پیش‌فرض باشند

اگر سیستم دقیقا کار کند ← رزرو هتل و پرواز

کنترل کردن مسیر می‌شه ← خود سیستم همه هتل و همه پرواز

رزرو کند

اگر کاربر سوال بپرسه ← subdialogue ← مکالمه فرعی

② understanding

NLU ← فهم زبان یا منظور کردن slot

1) Domain classification → Airtravel

2) intent determination ← اهداف جانبی وجود

show_flights

3) slot filling (origin / date / time

parsing ← فعل / فاعل / مفعول و ...

GUS ← deep learning می‌توانیم slot filling و پرسیم.

3) response generation → template هرچی اسم

برای پاسخ

GUS → high preasion → rule ها هستش ولی

recall → کمتر هستش دیگه

چون میرسه به سایر ها

rule_base

Dialogue state → ① معنای پیشرفته‌تر و دقیق‌تر

چی شنیدیم و معنی

Automatic speech regnoition

ASR → SLU → Dialoge state track

Spoken Language understanc

TTS

Texttospeech

NLG ← Dialogue policy

چه حرفی تولید شود

اسلش چیکنیم

قاعدتا اصلی با GUS ← slot ها با machine learning پر می‌شوند.

Elipon

Scanned with CamScanner

Dialogue Acts ~ هدف اصلی از بخش های مکالمه چه

تعاملی دارن

classification

Speech act & grounding

tag ست کامل

و استخراج براساس آن

Hello → open a dialogue → هر مرحله label مناسب

بگیرد

confirm

inform

Sequence labeling → برای تخصیص متن با tag تعیین کنیم

begin

BIO شروع اسم

inside

Slot filling , pos

داده train با فرمت

BiLSTM

pretrain BERT

RNN & transform

Begin slot
inside slot
out slot

sequence ورودی

Dialogue state tracking → چه slot های به شکل اصلی هستن

معنه تعامل چی پرسیدیم

correction

همه چی هر بار درست کنیم

correction ~> نمایش به کاربر تکرار کنه

نمایش به منفی میاد دیگه

مثلا نمودیان عوض شود

که همش correction فرق داره.

Dialogue policy ~> در هر مرحله سیستم باید تصمیم بگیرد

policy بهینه

بر اساس اطلاعات از ابتدا تا لحظه فعلی داریم.

لحظه فعلی | (یک لحظه قبل) | ابتدا

Reinforcement learning ~> پاسخ از دهد ی

سریع آموزش

Execture ~> reward

confirm ~> تایید کاربر

elicit ~> رو بدهد

explicit / implicit

تعیین مشخص

سیستم فعال شده

تعیین سوال دیگه بعد دستگیر میشه

rejection -

کامپیوتر متوجه نشد

کش و یا ماشینه

confidence score

برای هر اسلاتی که استخراج می‌کنیم

یک اسکور می‌دیم که آیا reject بشه یا confirm انجام بدیم یا نه

NLG → generation

→ content planning

→ sentence realization

تولید جمله

سوال / confirm

ترجیح می‌دهیم سوالات کمتری انجام بدهیم

Pelexcialise ← واژه‌هایی که دیتابیس دارد slot

هستند با tag مشخص کنیم

<resturant_name>

<resturant_menu>

او بین هوا رتا میز دقیق‌تر انتخاب می‌کنیم

ارزیابی chatbot

1) tasked based dialogue ارزیابی بر اساس سیگنال

2) conversational chatbot ارزیابی بر اساس انسان

Scanned with CamScanner

participant evalution ←

۸ هدف مختلف برای ارزیابی

1. اعتبار و تکرار
2. مناسب بودن
3. منطقی هست
4. fluency → روان باشد
5. listening → چقدر لغات باشد

(A+B) test → روتا سیستم متفاوت با هم مقایسه میشه

Automatic evalution ← text generate

gold ← یک متن با یک متن استاندارد که به عنوان داده

داده مقایسه می‌شه

(BLEU) → machine translation → Recall-oriented

(ROUGE) → summrazition → precision oriented

چقدر overlap وجود داره

chatbot → نوع داده

≠ مقایسه با خروجی از قبل مشخص شده بعد داده مدل

Elipon

Scanned with CamScanner