An overview of language representation

Shiwei Tong

Outline

- Brief Introduction
- Quality Evaluation
- General Methods
    - Token/ID based methods
    - Morphology based methods
- Summary

Language Representation

- Basic task of Natural Language Processing (NLP)
- To represent human language (e.g., character/word/sentence/document) in computer language/coding

Naïve Language Representation

- Character Encoding
  - ACSII
  - UTF-8
  - …
- One-hot Encoding
- Shortcoming
  - Cannot express semantics
    - Thesaurus: person & people
    - Synonyms: dad & father
    - Polysemy: bank & bank
  - Large space

ASCII表

(American Standard Code for Information Interchange 美国标准信息交换代码)

高四位		ASCII控制字符												ASCII打印字符														
		0000						0001						0010		0011		0100		0101		0110		0111		
		0						1						2		3		4		5		6		7		
低四位		十进制	字符	Ctrl	代码	转义字符	字符解释	十进制	字符	Ctrl	代码	转义字符	字符解释	十进制	字符	十进制	字符	十进制	字符	十进制	字符	十进制	字符	十进制	字符	Ctrl
0000	0	0		^@	NUL	\0	空字符	16	►	^P	DLE		数据链路转义	32		48	0	64	@	80	P	96	`	112	p	
0001	1	1	☺	^A	SOH		标题开始	17	◄	^Q	DC1		设备控制 1	33	!	49	1	65	A	81	Q	97	a	113	q	
0010	2	2	☻	^B	STX		正文开始	18	↕	^R	DC2		设备控制 2	34	"	50	2	66	B	82	R	98	b	114	r	
0011	3	3	♥	^C	ETX		正文结束	19	‼	^S	DC3		设备控制 3	35	#	51	3	67	C	83	S	99	c	115	s	
0100	4	4	♦	^D	EOT		传输结束	20	¶	^T	DC4		设备控制 4	36	$	52	4	68	D	84	T	100	d	116	t	
0101	5	5	♣	^E	ENQ		查询	21	§	^U	NAK		否定应答	37	%	53	5	69	E	85	U	101	e	117	u	
0110	6	6	♠	^F	ACK		肯定应答	22	▬	^V	SYN		同步空闲	38	&	54	6	70	F	86	V	102	f	118	v	
0111	7	7	•	^G	BEL	\a	响铃	23	↨	^W	ETB		传输块结束	39	'	55	7	71	G	87	W	103	g	119	w	
1000	8	8	◘	^H	BS	\b	退格	24	↑	^X	CAN		取消	40	(	56	8	72	H	88	X	104	h	120	x	
1001	9	9	○	^I	HT	\t	横向制表	25	↓	^Y	EM		介质结束	41	)	57	9	73	I	89	Y	105	i	121	y	
1010	A	10	◙	^J	LF	\n	换行	26	→	^Z	SUB		替代	42	*	58	:	74	J	90	Z	106	j	122	z	
1011	B	11	♂	^K	VT	\v	纵向制表	27	←	^[	ESC	\e	溢出	43	+	59	;	75	K	91	[	107	k	123	{	
1100	C	12	♀	^L	FF	\f	换页	28	∟	^\	FS		文件分隔符	44	,	60	<	76	L	92	\	108	l	124	|	
1101	D	13	♪	^M	CR	\r	回车	29	↔	^]	GS		组分隔符	45	-	61	=	77	M	93	]	109	m	125	}	
1110	E	14	♫	^N	SO		移出	30	▲	^^	RS		记录分隔符	46	.	62	>	78	N	94	^	110	n	126	~	
1111	F	15	☼	^O	SI		移入	31	▼	^-	US		单元分隔符	47	/	63	?	79	O	95	_	111	o	127	⌂	^Backspace 代码：DEL

注：表中的ASCII字符可以用“Alt + 小键盘上的数字键 ”方法输入。 2013/08/08

Language Representation with Semantics

- Token/ID based methods
  - Context2Center: CBOW
  - Center2Context: Skipgram
  - FeatureModel: ELMO, GPT, BERT
- Morphology based methods
  - Word Morphology: prefix, suffix, root
  - Sentence Morphology: grammar, syntax
  - Document Morphology: Paragraph

Language Representation Evaluation

- General Semantics Task
  - Word Similarity:
    - (消费者, 顾客)
    - (man, woman)
  - Word Analogy:
    - 国王 – 男人 = 女王 –女人
    - king – man = queen - woman
- Task Specific Evaluation
  - Classification: Precision, Recall, F1, …
  - Translation: BLEU
  - …

General Semantics Task

- Word Similarity
  - A list of pairs (word1, word2, score)
  - High similarity, high rank: $rank\_score = \cos(word1, word2)$
  - Spearman's correlation
- Word Analogy
  - A list of pairs (head1, tail1, head2, tail2)
  - Accuracy : $\mathrm{head2} = \underset{t \in \mathrm{W}}{\mathrm{argmax}} \big(\cos(\mathrm{head1} \ - \mathrm{tail1} + \mathrm{tail2}, \mathrm{t})\big)$
    - The above is called 3CosAdd
    - The other one expression, which is called 3CosMul, is $\mathrm{head2} = \underset{t \in \mathrm{W}}{\mathrm{argmax}} \frac{\cos(t, tail1) \cdot \cos(t, head2)}{\cos(t, head1) + \epsilon}$

Word Embedding: a demo case

- Semantics: similarity and analogy

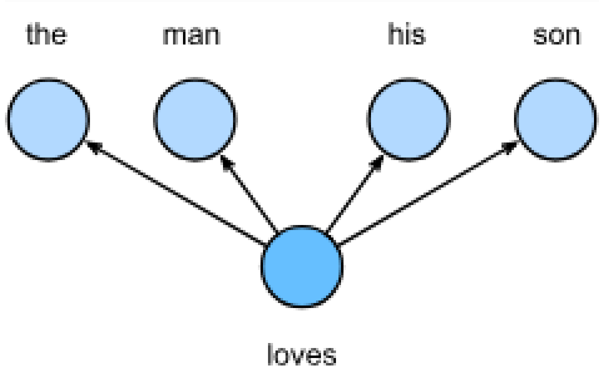
- Naïve way: one-hot
  - Hard to infer the semantics

- Distributed —— Word2vec
  - A sentence: the man loves his son
  - Assumption: similar words appear in similar contexts

Token/ID based methods

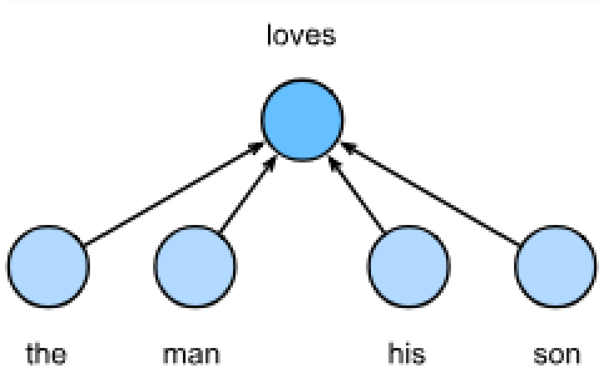
Center2Context: Skipgram

- $P(\text{"the"}|\text{"loves"}) \cdot P(\text{"man"}|\text{"loves"}) \cdot \cdots$
- $P(w_o|w_c) = \frac{\exp(u_o^T v_c)}{\sum_{i \in \mathcal{V}} \exp(u_i^T v_c)}$
- $\prod_{t=1}^{T} \prod_{-m \leq j \leq m, j \neq 0} P\left(w^{t+j}|w^t\right)$
- $-\sum_{t=1}^{\mathrm{T}} \sum_{-m \leq j \leq m, j \neq 0} \log P(w^{t+j}|w^t)$



Context2Center: CBOW

- $\mathrm{P}(\mathrm{loves}|\text{"the"}, \mathrm{man}\text{"}, \mathrm{his}\text{"}, \mathrm{son})$
- $P\left(w_c|w_{o_1,\ldots o_{2m}}\right) = \frac{\exp\left(u_c^T \cdot \frac{1}{2m} \sum_i^{2m} v_{o_i}\right)}{\sum_{i \in \mathcal{V}} \exp(u_i^T \cdot \frac{1}{2m} \sum_j^{2m} v_{o_j})}$
- $P(w_c|W_o) = \frac{\exp\left(u_c^T \bar{v}_o\right)}{\sum_{i \in \mathcal{V}} \exp(u_i^T \cdot \bar{v}_o)}$
- $\prod_{t=1}^{T} P(w^t|w^{t-m}, \ldots, w^{t+m})$



Problem

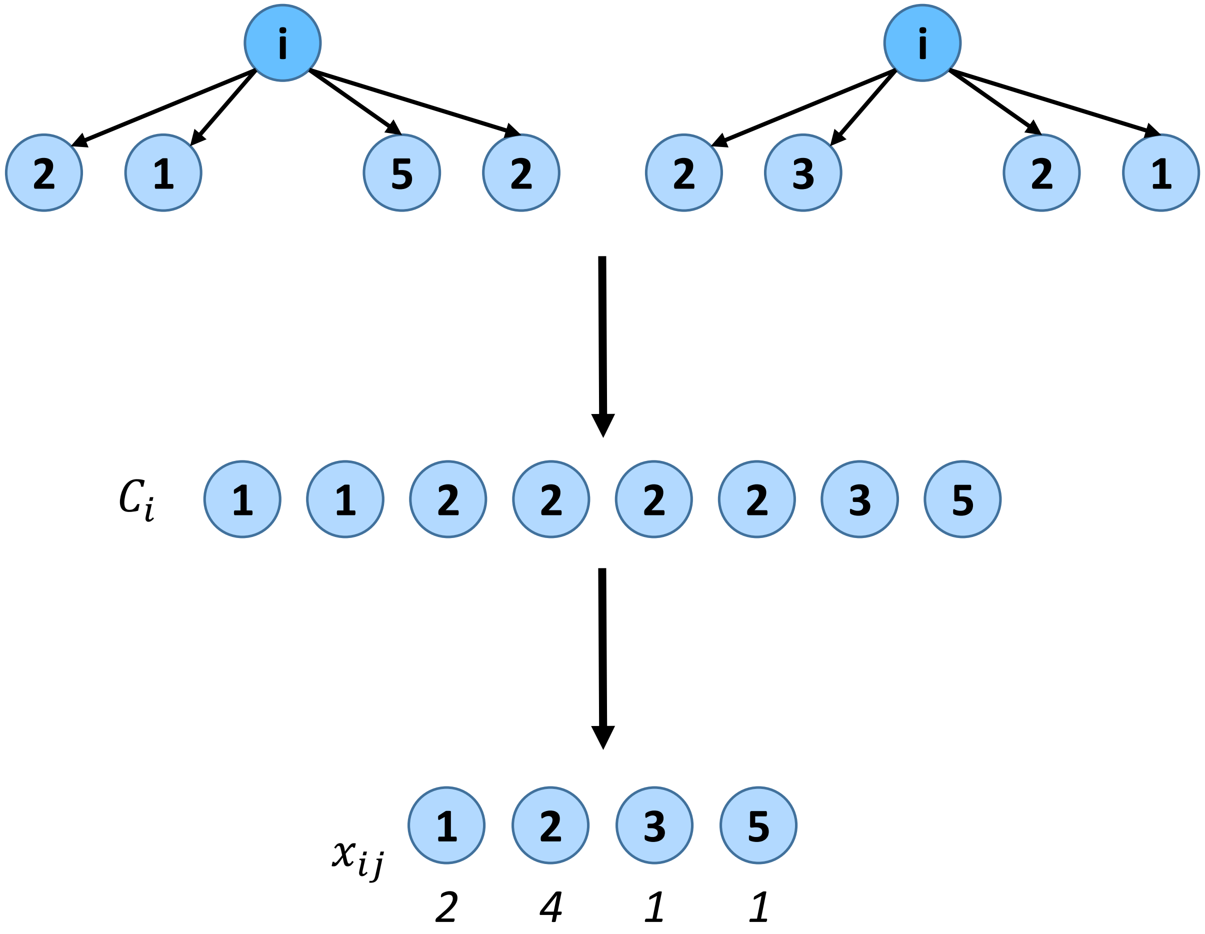
- Bias from Frequency
  - "the" vs "microprocessor"
  - Solution
    - Subsample based on frequency
      - Dropout Probability: $P(w_i) = \max(1 - \sqrt{\frac{c}{f(w_i)}}, 0)$
      - $f(w_i)$ is the word frequency, and c is a constant, usually 10^{-3} or 10^{-4}
- High Time Complexity
  - $\mathrm{O}(|\mathcal{V}|)$
    - $P(w_o|w_c) = \frac{\exp(u_o^T v_c)}{\sum_{i\in\mathcal{V}} \exp(u_i^T v_c)}$
    - $P(w_c|W_o) = \frac{\exp(u_c^T \bar{v}_o)}{\sum_{i\in\mathcal{V}} \exp(u_i^T \cdot \bar{v}_o)}$

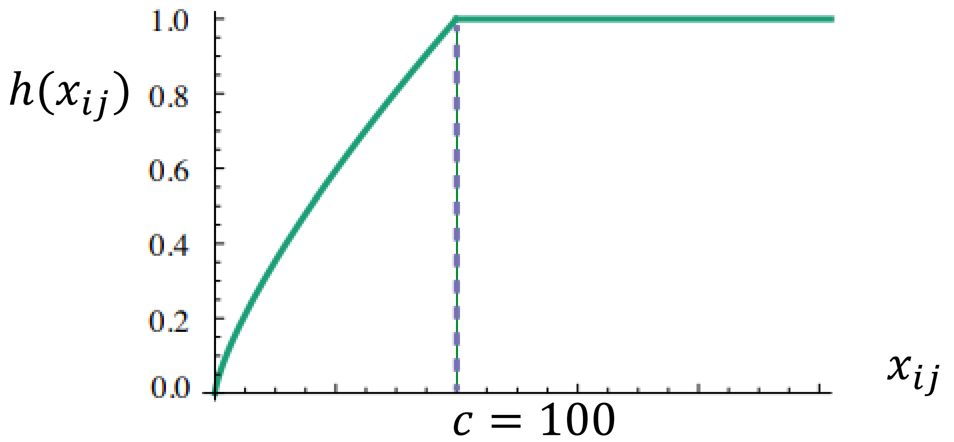
High Time Complexity

- $\mathrm{O}(|\mathcal{V}|)$
  - $P(w_o|w_c) = \frac{\exp(u_o^T v_c)}{\sum_{i \in \mathcal{V}} \exp(u_i^T v_c)}$
- Solution
  - Negative sampling
    - $P(w_o|w_c) \to \max$
    - $P(D = 1|w_o, w_c) = \sigma(u_o^T v_c) \to max.$ σ is the sigmoid function.
    - $P(w_o|w_c) \to P(D = 1|w_o, w_c) \prod_{k=1, w_k \sim U(w)}^{K} P(D = 0|w_k, w_c)$
    - $U(w) = \frac{f^{\alpha}(w)}{\sum_i f^{\alpha}(w_i)}$, is the powered unigram distribution. α is a constant (e.g., $\frac{3}{4}$) and K usually is set as 5.
  - Hieratical softmax
    - Space for time

Global Vectors

- Skipgram
  - Standard
    - $P(w_o|w_c) = \frac{\exp(u_o^T v_c)}{\sum_{i \in \mathcal{V}} \exp(u_i^T v_c)}$
    - $-\sum_{t=1}^{\mathrm{T}} \sum_{-m \le j \le m, j \ne 0} \log P\left(w^{t+j} \middle| w^t\right)$
  - Another View
    - $p_{ij} = P\left(w_j \middle| w_i\right) = \frac{\exp\left(u_j^T v_i\right)}{\sum_{k \in \mathcal{V}} \exp\left(u_j^T v_i\right)}$
    - $-\sum_{i \in \mathcal{V}} \sum_{j \in \mathcal{V}} x_{ij} \log q_{ij}$
    - $x_i = |C_i|, p_{ij} = \frac{x_{ij}}{x_i}$
- GloVe
  - $p'_{ij} = x_{ij}, q'_{ij} = \exp\left(u_j^T v_i\right)$
  - $\left(\log q'_{ij} - \log p'_{ij}\right)^2 = \left(u_j^T v_i + b_i + c_j - \log x_{ij}\right)^2$
  - $\sum_{i \in \mathcal{V}} \sum_{j \in \mathcal{V}} h\left(x_{ij}\right) \left(u_j^T v_i + b_i + c_j - \log x_{ij}\right)^2$
  - $h\left(x_{ij}\right) = \begin{cases} \left(x_{ij}/c\right)^\alpha & x_{ij} < c \\ 1 & otherwise \end{cases}$



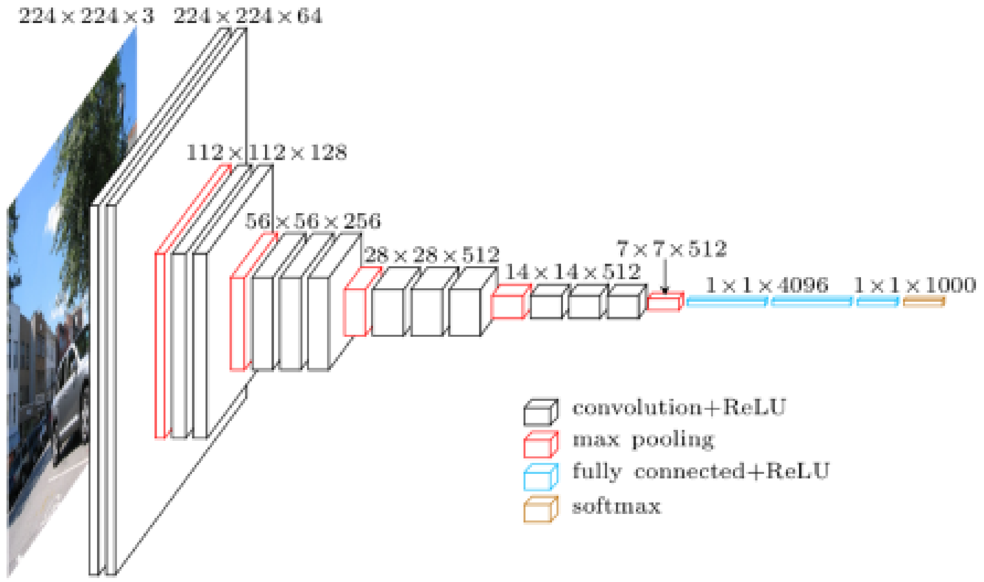


Problem

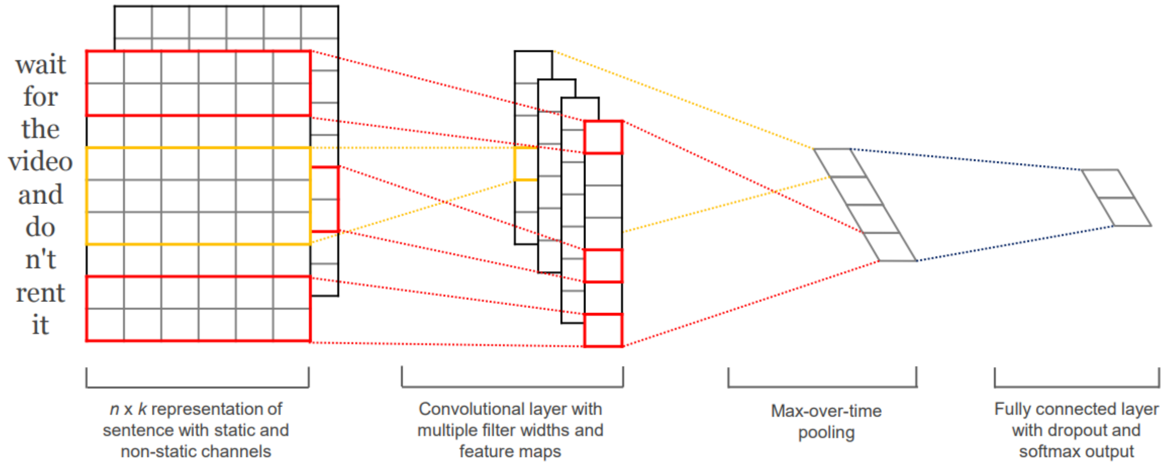
- Polysemy
  - Bank & Bank
  - He walks on the bank
  - The bank is robbed

FeatureModel

- Image
  - VGG16 & VGG19
  - ResNet
- NLP
  - TextCNN
  - ELMO
  - GPT
  - BERT



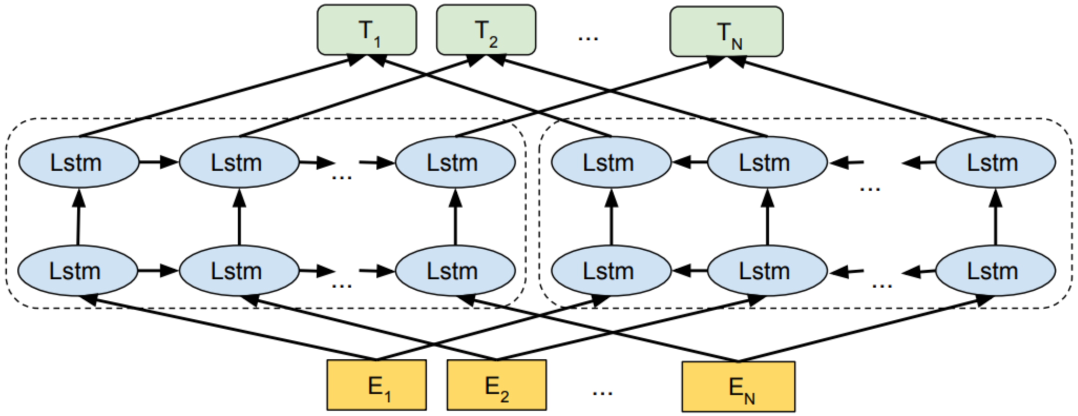
VGG16



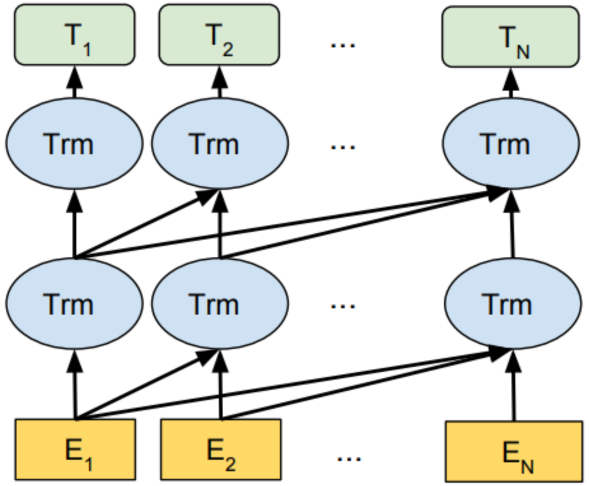
TextCNN

FeatureModel

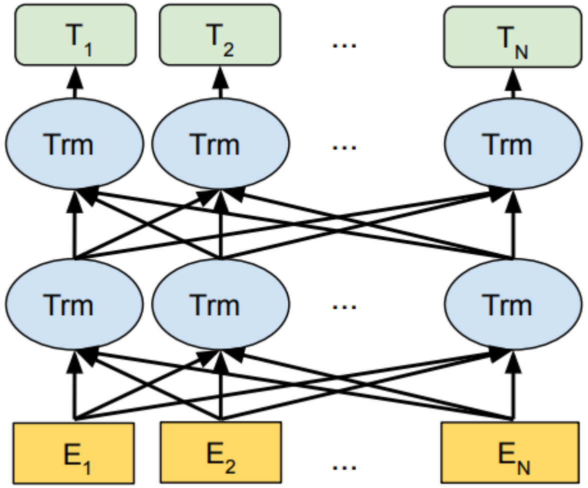
- ELMO



- GPT



- BERT



Drawback

- Ignore the abundant information in morphology
  - Word Morphology: prefix, suffix, root
  - Sentence Morphology: grammar, syntax
  - Document Morphology: Paragraph
- OOV Word

Morphology based methods

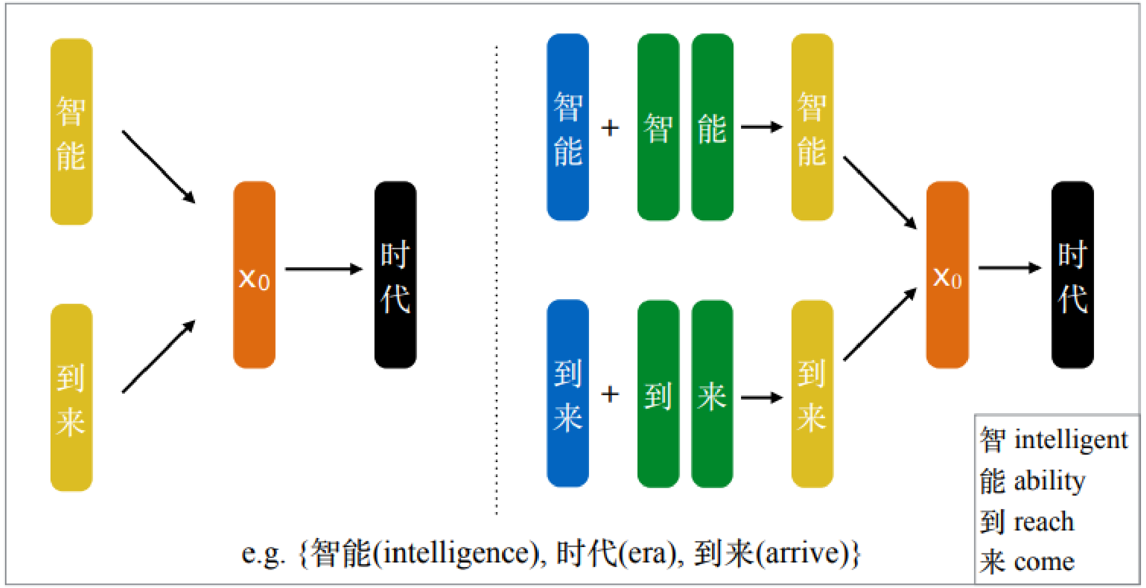
- Alphabetic
  - Subword: prefix, suffix, root
- Logogram
  - Character
  - Stroke
  - Glyph

Morphology Word Embedding for Alphabetic

- sisg
  - fasttext
  - “declare”, “clarify” → root word “clar”
  - (“dog”, “dogs”), (“interest”, “interesting”)
  - Subword n-gram
    - where → <where>
    - 3-gram → <wh, whe, her, ere, re>
    - $3 \leq n \leq 6$ and attach a special subword “where” or “<where>”
    - $v_w = \sum_{g \in \mathcal{G}_w} z_g$
    - $P(w_o|w_c) = \frac{\exp(u_o^T v_c)}{\sum_{i \in \mathcal{V}} \exp(u_i^T v_c)}$

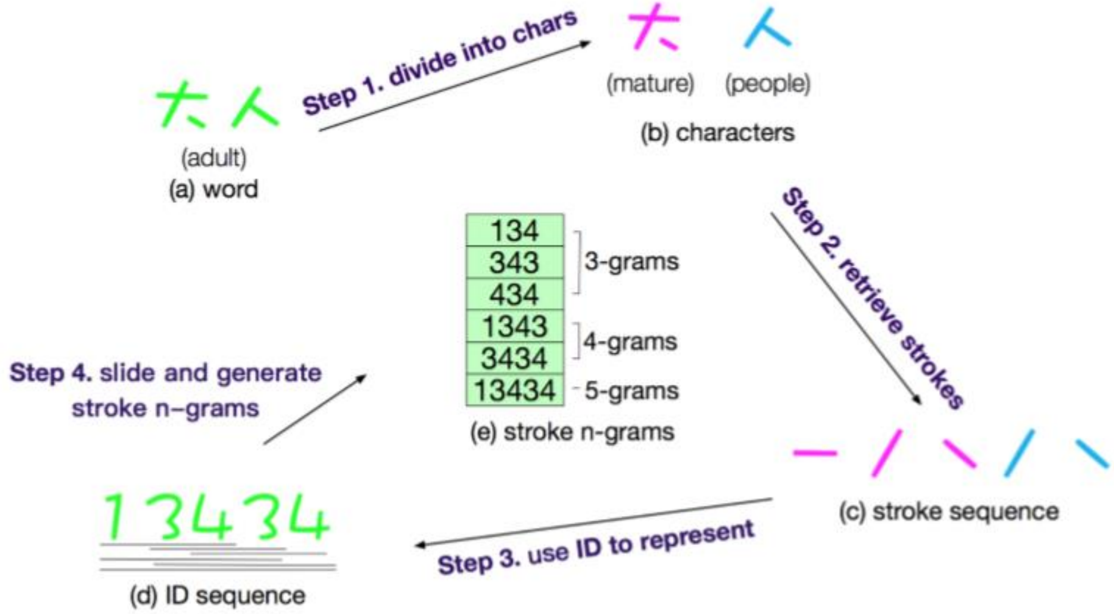
Morphology Word Embedding for Logogram

- Character
  - $u_j = w_j \oplus \frac{1}{N_j}\sum_{k=1}^{N_j} c_k$



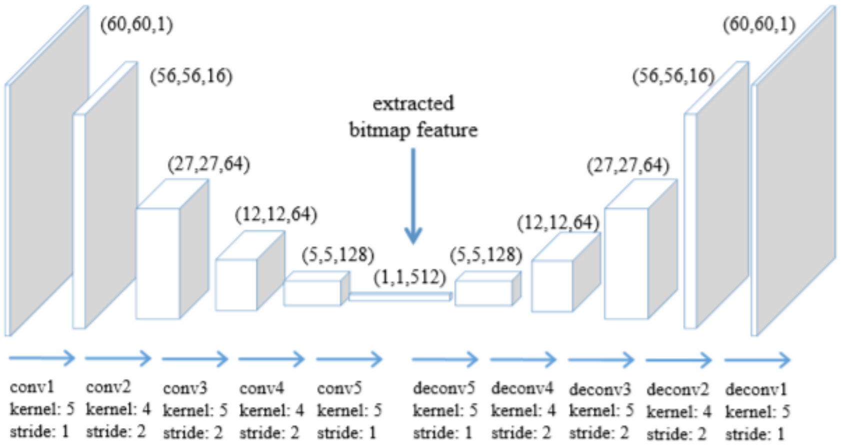
(A) CBOW (B)Character-enhanced Word Embedding

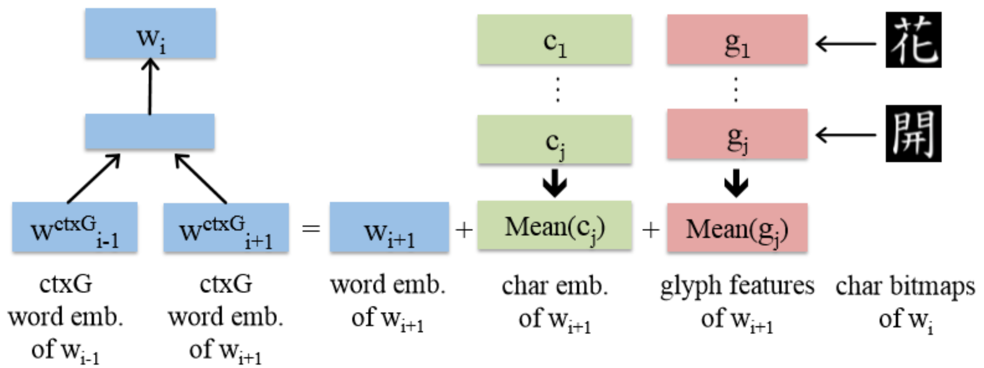
- Stroke



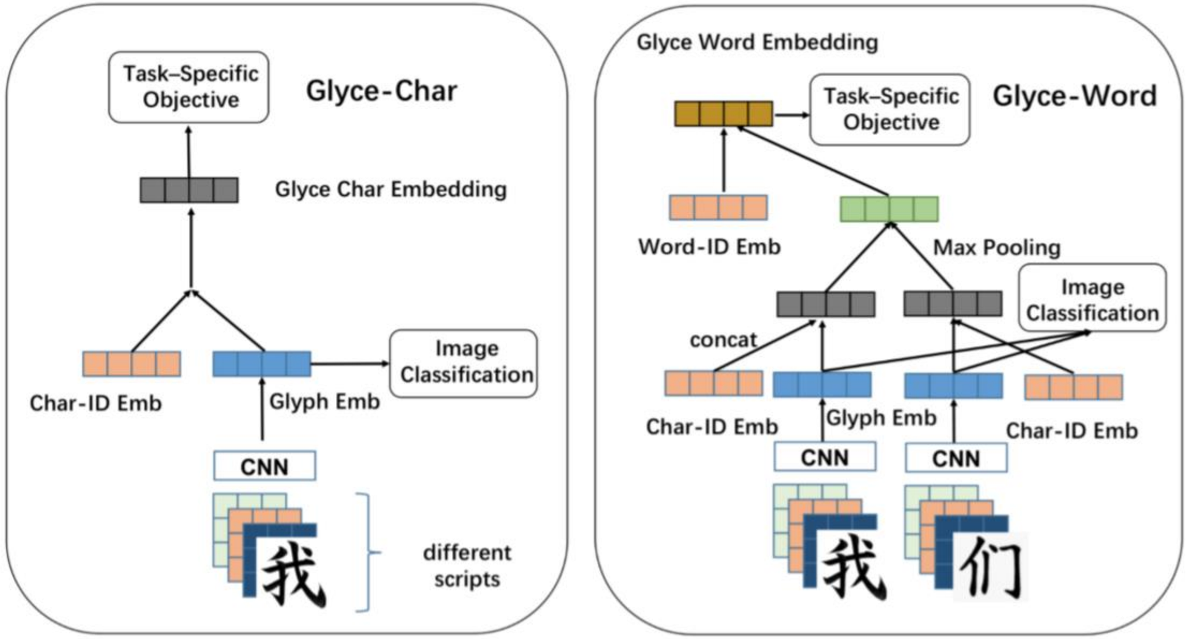
Glyph

- GWE





- Glyce



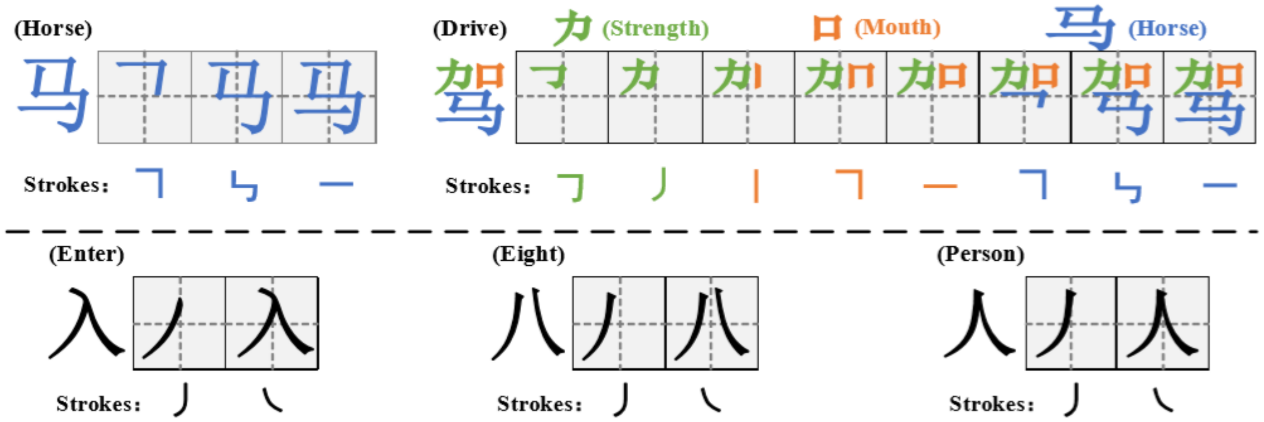
Dual channel view for Morphology Logogram

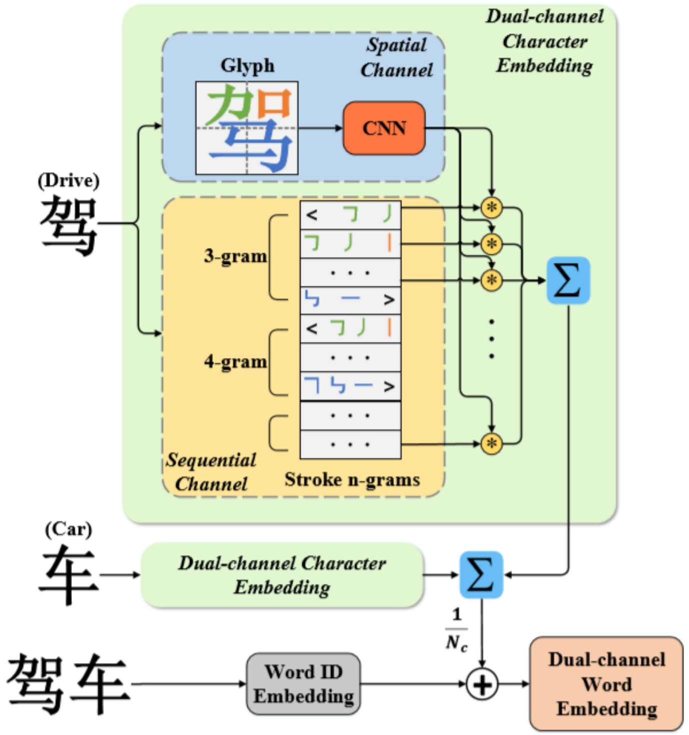
Sequential

- Character
- Stroke

Spatial

- Glyph





Problem

- High Time Complexity
- High Computing Resource
- Weak Interpretability
  - Only can infer similarity and analogy

Future Work

- Try to compress corpus into knowledge graph
  - Reduce time complexity
  - Reduce computing Resource
  - Strong interpretability
  - High interactivity

Summary

- Challenge
  - Semantics
    - Distributed Embedding
  - Polysemy
    - Feature Model
  - OOV
- Quality Evaluation
  - General semantics task: word similarity, word analogy
  - Task specific evaluation: classification, translation
- Practical
  - Time Complexity
    - Negative Sampling
  - Interpretability
    - ?

Reference

- Mikolov T, Sutskever I, Chen K, et al. Distributed representations of words and phrases and their compositionality[C]//Advances in neural information processing systems. 2013: 3111-3119.
- Mikolov T, Chen K, Corrado G, et al. Efficient estimation of word representations in vector space[J]. arXiv preprint arXiv:1301.3781, 2013.
- Pennington J, Socher R, Manning C. Glove: Global vectors for word representation[C]//Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). 2014: 1532-1543.
- Kim Y. Convolutional neural networks for sentence classification[J]. arXiv preprint arXiv:1408.5882, 2014.
- Peters M E, Neumann M, Iyyer M, et al. Deep contextualized word representations[J]. arXiv preprint arXiv:1802.05365, 2018.
- Radford A, Narasimhan K, Salimans T, et al. Improving language understanding by generative pre-training[J]. URL https://s3-us-west-2. amazonaws. com/openai-assets/research-covers/languageunsupervised/language understanding paper. pdf, 2018.
- Devlin J, Chang M W, Lee K, et al. Bert: Pre-training of deep bidirectional transformers for language understanding[J]. arXiv preprint arXiv:1810.04805, 2018.

Reference

- Bojanowski P, Grave E, Joulin A, et al. Enriching word vectors with subword information[J]. Transactions of the Association for Computational Linguistics, 2017, 5: 135-146.
- Chen X, Xu L, Liu Z, et al. Joint learning of character and word embeddings[C]//Twenty-Fourth International Joint Conference on Artificial Intelligence. 2015.
- Cao S, Lu W, Zhou J, et al. cw2vec: Learning chinese word embeddings with stroke n-gram information[C]//Thirty-Second AAAI Conference on Artificial Intelligence. 2018.
- Su T R, Lee H Y. Learning chinese word representations from glyphs of characters[J]. arXiv preprint arXiv:1708.04755, 2017.
- Wu W, Meng Y, Han Q, et al. Glyce: Glyph-vectors for Chinese Character Representations[J]. arXiv preprint arXiv:1901.10125, 2019.

Q&A