AI科技大本营 | 腾讯云 | 微信

深度学习在NLP中的发展和应用

张金超 博士
微信模式识别中心高级研究员

目录

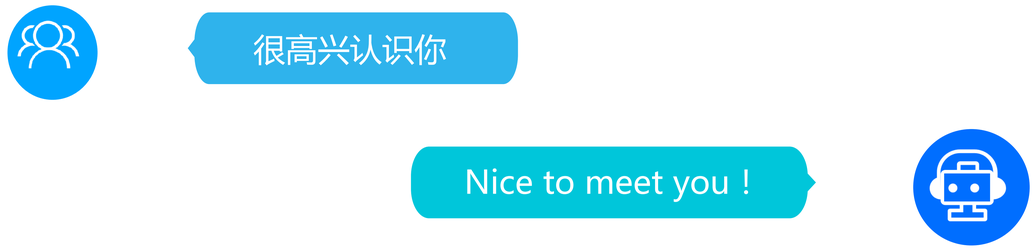
目录

自然语言处理基本概念

自然语言处理（Natural Language Processing，NLP）是指借助于计算机技术来分析、理解和生成人类的自然语言的过程。

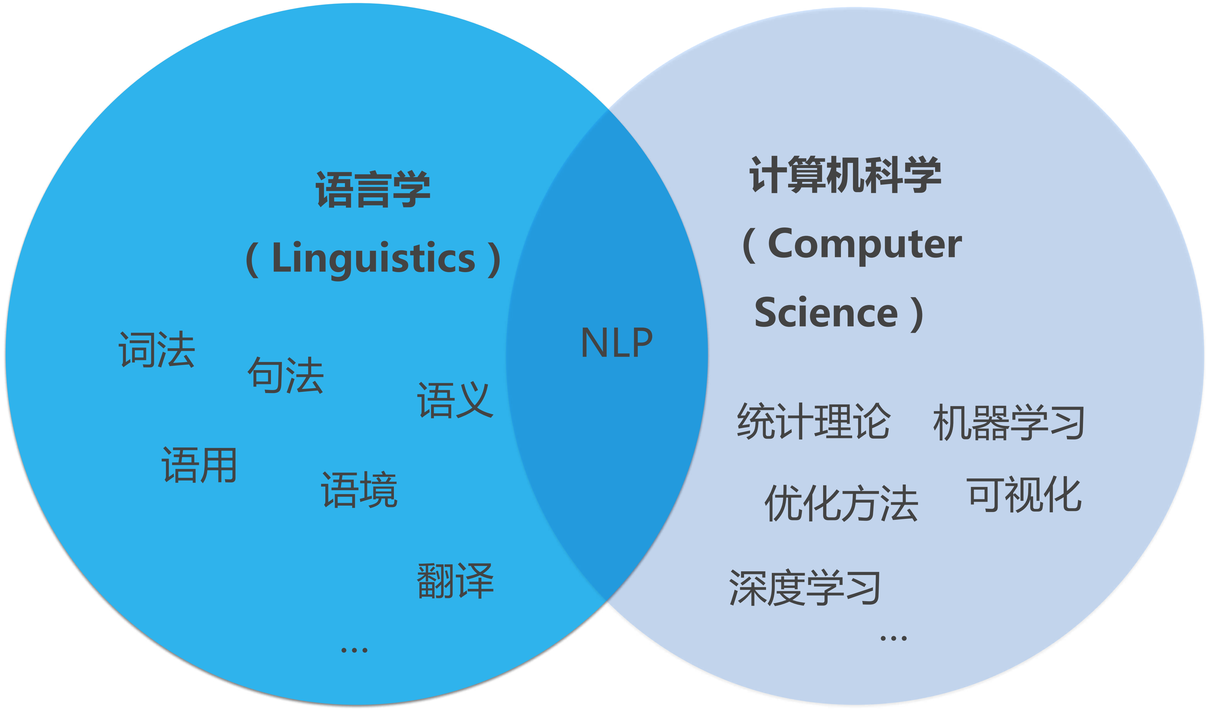
对话机器人（Chatbot）

机器翻译（Machine Translation）



自然语言处理基本概念

自然语言处理属于计算语言学科，是语言学和计算机科学的交叉学科，是人工智能的一个重要方面。



自然语言处理任务栈

顶层任务（High-level Tasks）

信息抽取（Information Extraction）

语义分析（Semantic Analysis）

句子分析（Sentence Analysis）

词法分析（Lexical Analysis）

自然语言处理任务栈—词法分析（ Lexical Analysis ）

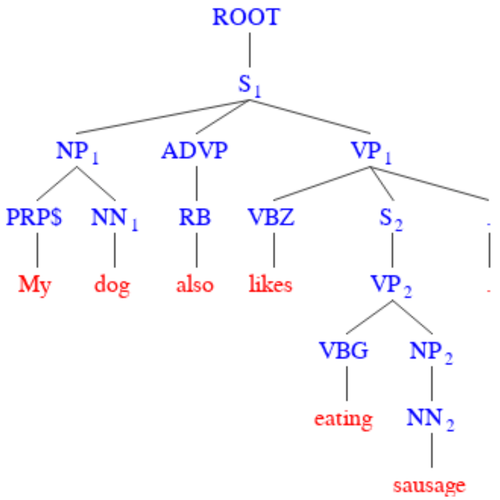
词法分析（Lexical Analysis）：对自然语言进行词汇层面的分析，是NLP基础性工作。

- 分词（Word Segmentation/Tokenization）：对没有明显词边界的文本进行切分，得到词序列。
  - “长江是中华民族的母亲河”
  - “长江　是　中华民族　的　母亲　河”
- 新词发现（New Words Identification）：找出文本中具有新形式、新意义或是新用法的词。
  - “活久见”，“十动然拒”，“十动然揍”
- 形态分析（Morphological Analysis）：分析单词的形态组成，包括词干(Stems)、词根(Roots)、词缀(Prefixes and Suffixes)等。
  - scored=score+d。
- 词性标注（Part-of-speech Tagging）：确定文本中每个词的词性，词性包括动词(Verb)、名词(Noun)、代词（pronoun）等。
  - “My/**PRPS** dog/**NN** also/**RB** likes/**VBZ** eating/**VBG** sausage/**NN**”
- 拼写校正（Spelling Correction）：找出拼写错误的词并进行纠正。

自然语言处理任务栈—句子分析（Sentence Analysis）

句子分析（Sentence Analysis）：对自然语言进行句子层面的分析，包括句法分析（Syntactic Parsing）和其他句子级别的分析任务。

- 组块分析（Chunking）：标出句子中的短语块，例如名词短语（NP），动词短语（VP）等。
- 超级标签标注(SuperTagging)：给每个句子中的每个词标注上超级标签，超级标签是句法树中与该词相关的树形结构。
- 成分句法分析（Constituency Parsing）：分析句子的成分，给出一棵由终结符和非终结符构成的成分句法树。
- 依存句法分析（Dependency Parsing）：分析句子中词之间的依存关系，给一棵由词语依存关系构成的依存句法树。



图：成分句法树

自然语言处理任务栈—句子分析（Sentence Analysis）

句子分析（Sentence Analysis）：对自然语言进行句子层面的分析，包括句法分析（Parsing）和其他句子级别的分析任务。

- 语言模型（Language Modelling）：对给定的一个句子进行打分，该分数代表句子合理性 (流畅度)的程度。
- 语种识别（Language Identification）：给定一段文本，确定该文本属于哪个语种。
- 句子边界检测（Sentence Boundary Detection）：给没有明显句子边界的文本加边界。

自然语言处理任务栈—语义分析（Semantic Analysis）

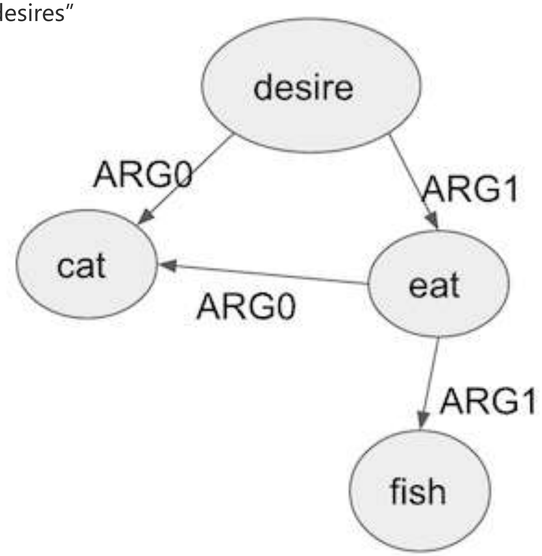
语义分析（Semantic Analysis）：对给定文本进行分析和理解，形成能够表达语义的形式化表示或是分布式表示。

- 词义消岐（Word Sense Disambiguation）：对有歧义的词，确定其准确的词义。
- 语义角色标注（Semantic Role Labeling）：标注句子中的语义角色类标，语义角色包括施事、受事、影响等。
- 抽象语义表示分析（Abstract Meaning Representation Parsing）：AMR是一种抽象语义表示形式，AMR parser把句子解析成AMR结构。
- 一阶谓词逻辑演算（First-Order Predicate Calculus）：使用一阶谓词逻辑系统表达语义。
- 框架语义分析（Frame Semantic Parsing）：根据框架语义学的观点，对句子进行语义分析。
- 词汇/句子/段落的向量化表示（Word/Sentence/Paragraph Vector）：研究词汇/句子/段落的向量化方法，向量的性质和应用。

自然语言处理任务栈—语义分析（Semantic Analysis）

抽象语义表示分析（AMR）：

- “The cat wants to eat the fish”
- “The cat’s desire is to eat fish”
- “Eating fish is what the cat desires”



图：AMR结构表示语义

http://www.akornilo.com/computational-semantics/

自然语言处理任务栈—信息抽取（Information Extraction）

信息抽取(Information Extraction，IE）：从无结构文本中抽取结构化的信息。

- 命名实体识别（Named Entity Recognition）：从文本中识别出命名实体，实体一般包括人名、地名、机构名、时间、日期、货币、百分比等。
- 实体消岐（Entity Disambiguation）：确定实体指代的现实世界中的对象。
- 术语抽取（Terminology/Glossary Extraction）：从文本中确定术语。
- 共指消解（Coreference Resolution）：确定不同实体的等价描述，包括代词消解和名词消解等
- 关系抽取（Relationship Extraction）：确定文本中两个实体之间的关系类型。
- 事件抽取（Event Extraction）：从无结构的文本中抽取结构化的事件。

6月29日起，北京市在已实现外贸领域“十五证合一”的基础上，将再整合市财政局、市人力社保局、市住建委、市城市管理委、市农业局、市旅游委、市新闻出版广电局、市粮食局、市气象局等部门涉及信息采集、记载公示、管理备查类的涉企证照事项，实现“二十四证合一”。此举意味着将有更多领域的企业享受到改革带来的便利。

自然语言处理任务栈—信息抽取（Information Extraction）

信息抽取(Information Extraction，IE）：从无结构文本中抽取结构化的信息。

- 情感分析（Sentiment Analysis）：对文本中的主观性情绪进行提取。
- 意图识别（Intent Detection）：对话系统中的一个重要模块，对用户给定的对话内容进行分析，识别用户意图。
- 槽位填充（Slot Filling）：对话系统中的一个重要模块，从对话内容中分析出与用户意图相关的有效信息。

自然语言处理任务栈—顶层任务（High-level Tasks）

顶层任务（High-level Tasks）：直接面向普通用户，提供自然语言处理产品服务的系统级任务，会用到多个层面的自然语言处理技术。

- 机器翻译（Machine Translation）：通过计算机自动化的把一种语言翻译成另外一种语言。
- 文本摘要（Text summarization/Simplication）：对较长的文本进行内容梗概的提取。
- 问答系统（Question-Answering System）：针对用户提出的问题，系统给出相应的答案。
- 对话系统（Dialogue System）：能够与用户进行聊天对话，从对话中捕获用户的意图，并分析执行。
- 阅读理解（Reading Comprehension）：机器阅读完一篇文章后，给定一些文章相关问题，机器能够回答。
- 自动文章分级（Automatic Essay Grading）：给定一篇文章，对文章的质量进行打分或分级。

自然语言处理任务的难点

- 歧义问题（Ambiguity）
- 知识问题（Knowledge）
- 离散符号计算问题（Symbol-based Computation）
- 语义本质问题（Essence of Semantics）

自然语言处理任务的难点—歧义问题

歧义问题（Ambiguity）：词汇层面，句子层面都会存在歧义问题。

词汇层面的歧义：“

“我们研究所/有东西”，“我们研究/所有东西”。

“这个人手里拿了一个苹果”，“我们公司目前缺人手”。

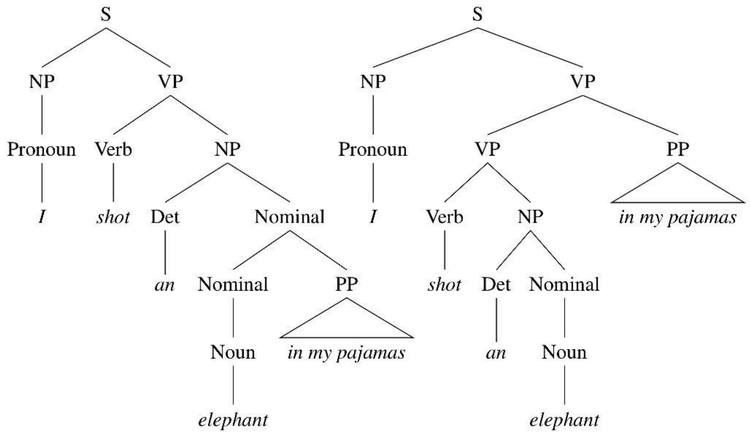
“乒乓球/拍卖/完了”，“乒乓球拍/买完了”。

多义词：“山上的杜鹃开了”，“树上有一只杜鹃在叫”。

词性兼类：一个词在不同的上下文环境中，体现的词性不同。“向雷锋同志学习”“他非常勤奋，学习很好”

结构性歧义：不同的结构视角下的句子，可能具有不同的语义。结构性歧义包括附着（attachment ）歧义、并列（coordination）歧义、名词短语括号（noun-phrase bracketing）歧义等。

I shot an elephant in my pajamas



图：结构性歧义

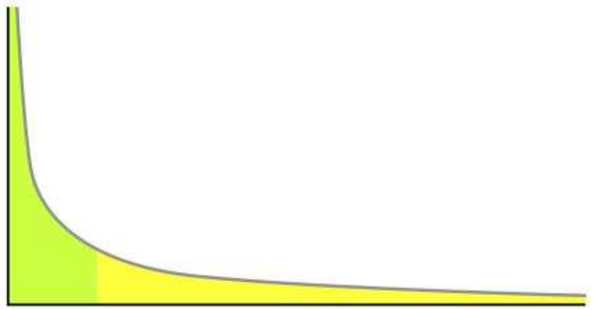
自然语言处理任务的难点—知识问题

知识稀疏（Knowledge Sparse）：词汇稀疏，搭配稀疏，语义稀疏。

齐夫定律（ Zip‘f law ）：自然语言语料库中，一个单词出现的频率与它在频率表中的排名基本成反比关系。

Word	Frequency	Rank	Zipf Frequency
“the”	69971	1	C
“of”	36411	2	C/2
“and”	28852	3	C/3

表：Brown 数据集中的zipf现象



图：长尾现象

齐夫定律是实验性定律，反映了语言中存在了严重的长尾现象（Long Tail）。这种词汇层面上的知识稀疏，会最终导致语料中搭配稀疏和语义稀疏，给模型学习造成困难。

知识依赖（Knowledge dependency）：对语言的精确理解和生成，多数情况下需要背景知识的支持。

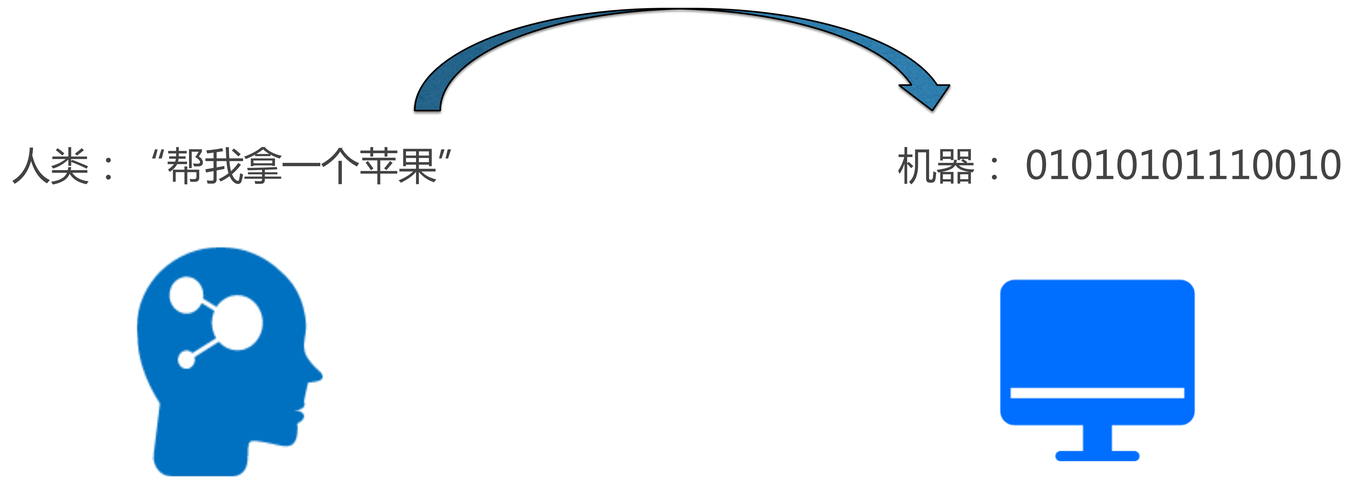
自然语言处理任务的难点—离散符号计算问题

面向符号的计算（Towards Symbol Computation）：

语言的问题其实是符号的问题，而计算机擅长数值计算，弱于逻辑运算和推理，并不适合针对符号的计算。需要把文本符号映射成数值化的特征，才能利用算机的强大的数值处理能力。

传统的基于one-hot正交的符号数值化方法，一方面面临高维的问题，另一方面难以建模符号之间的相关关系.。

基于分布式表示的符号数值化方式，可以绕过高维的问题，同时也可以建模符号之间的相关关系，但是可解释性差，可组合性也差。



自然语言处理任务的难点—语义本质问题

语义本质（Essence of Semantics）：到底什么是语义？

自然语言处理的终极目标是使得机器能够完全准确的理解人类语言中的语义，但是：

- 语义究竟是什么？
- 应该以什么方式在计算机中表达语义，是一种结构化的符号表示形式，还是数值型的表示形式，或是其他的表示形式？

What's in Languages ?

在语义本质问题解决之前，我们能做的事情是针对具体应用需求解决好自然语言处理中的各个子问题。

目前，几乎全部的自然语言处理方法都是基于数据驱动（Data-Driven）的，数据质量+模型能力（拟合能力和泛化能力）决定了最后的任务表现，而非机器真正能够全面准确的理解人类语言中的语义。

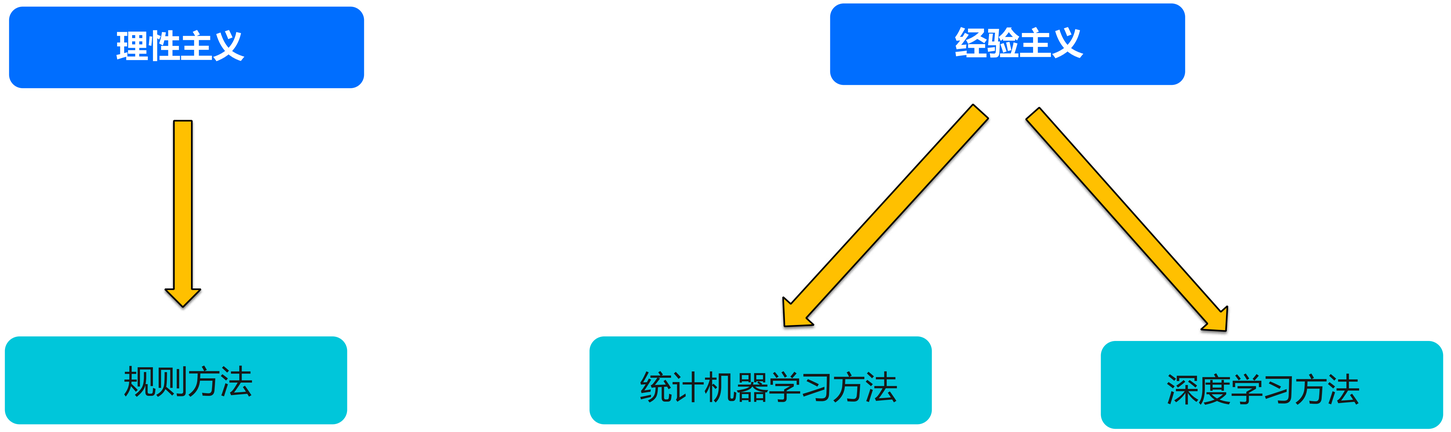
小结

本章介绍了自然语言处理的基本概念，学科范畴，对NLP任务和研究难点进行了梳理。要点包括：

- 自然语言处理任务背靠语言学理论、利用统计机器学习方法和深度学习方法等对文本进行分析、理解和生成。
- 自然语言的内在结构性，使得自然语言任务存在着层级和递进的关系。
- 自然语言处理任务会使用语言学理论，但一般不会从学术的角度研究语言学。
- 自然语言处理任务会使用统计机器学习和深度学习方法，很少会深入研究算法理论知识。
- 自然语言处理任务面向应用，重点关注应用场景分析、问题建模和算法使用。

目录

自然语言处理方法的演化



- 1950s-1980s
- 语言学家主导
- 语言学知识驱动
- 编写符合语言学意义的处理规则
- 方法可解释性好
- 规则构造成本高，易冲突

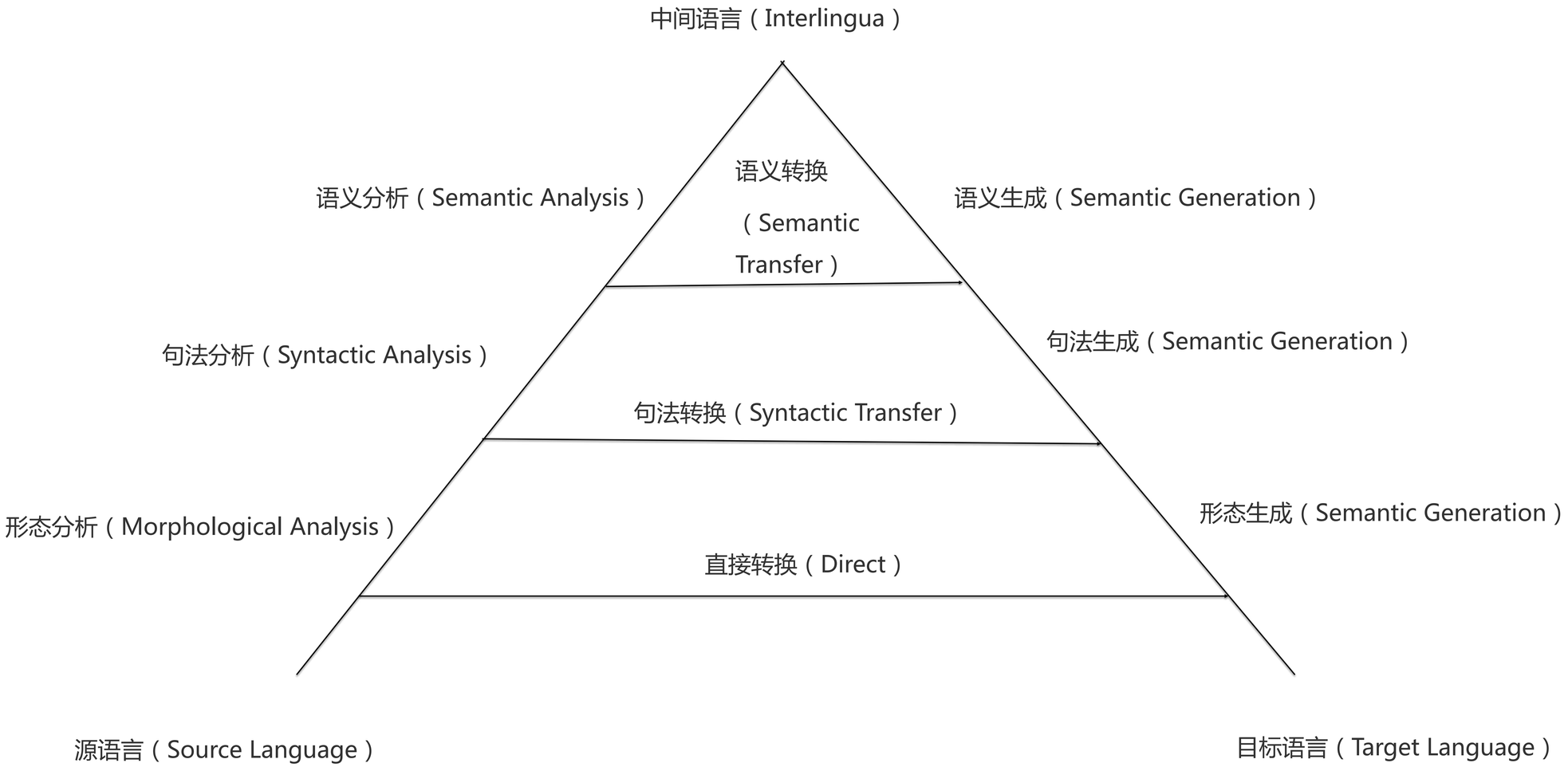
- 1990s-now
- 计算机科学家主导
- 基于数据驱动的机器学习模型
- 少量语言学知识

- 基于符号特征的计算
- 统计机器学习模型

- 基于分布式表示特征的计算
- 以各种神经网络为核心模型

自然语言处理中的理性主义模型

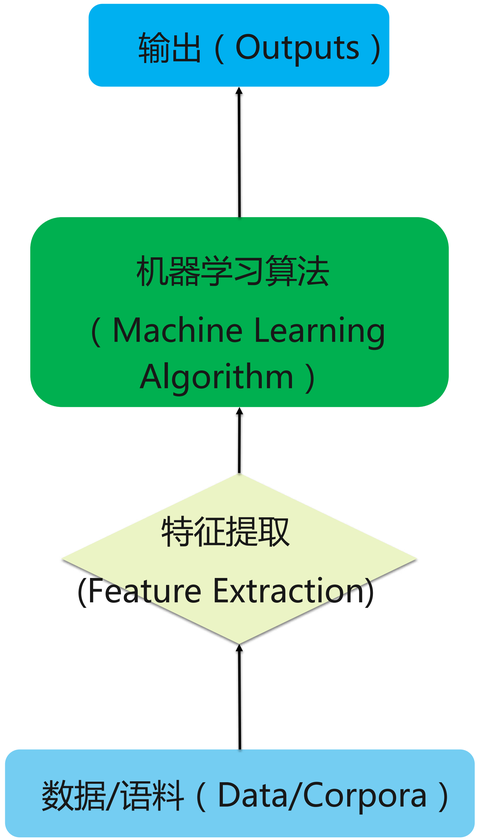
B.Vauquois 教授提出的机器翻译金字塔（Machine Translation Pyramid）：



理性主义视角下的自然语言处理的**核心**问题是**语言学知识的表达。**

自然语言处理中的常用模型

经验主义模型：



图：机器学习流程

经验主义是将自然语言处理任务建模成数据驱动的机器学习问题，一般的流程包括：

- 构建训练语料
- 特征提取
- 训练机器学习模型

经验主义视角下的自然语言处理的**核心**问题是**任务的建模和机器学习算法的求解。**

自然语言处理中的常用问题模型

问题模型与算法模型：

问题模型指将实际的自然语言处理任务建模成一个形式化的问题，然后对该问题进行求解。

算法模型是指求解某个问题模型时使用的具体的机器学习算法。

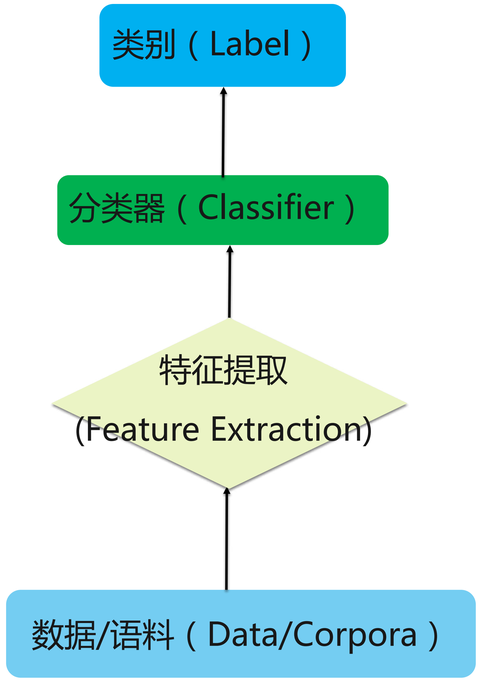
NLP中常用的问题模型：

- 分类模型（Classification Model）
- 序列标注模型（Sequence Labeling Model）
- 序列生成模型（Sequence Generation Model）

自然语言处理中的常用问题模型—分类模型

分类模型（Classification）：

使用分类器将文本进行类别标注，一些自然语言处理任务可以建模成分类问题，比如，文本分类，意图识别、情感分类等。



图：分类模型

分类模型常用的统计机器学习算法包括：逻辑回归模型（Logistic Regression，LR），贝叶斯模型（Bayes Model），支持向量机（Support Vector Machine，SVM），决策树（Decision Tree）等。

自然语言处理中的常用问题模型—序列标注模型

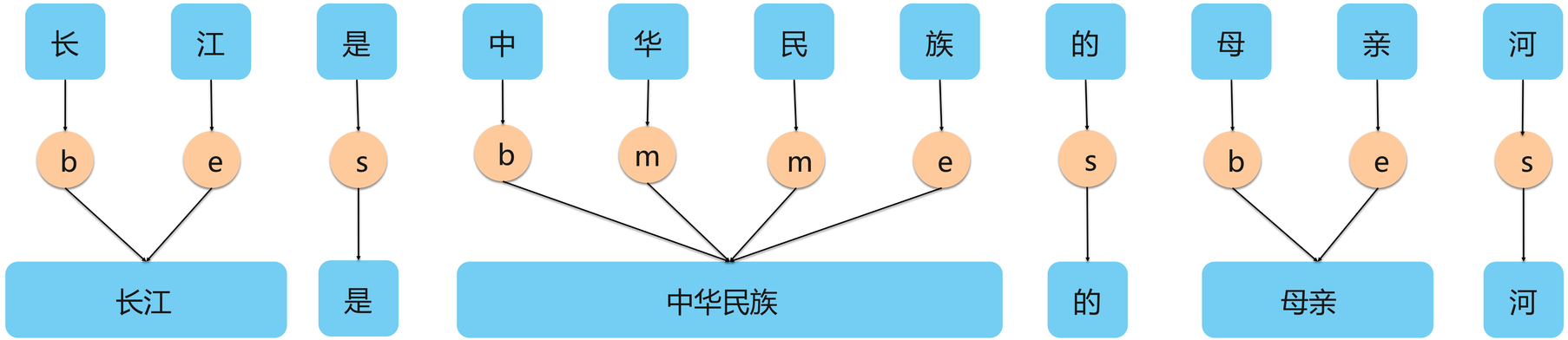
序列标注模型（Sequence Labeling Model）：

自然语言处理任务中最常用的问题模型。给定一个序列，对序列中的每个元素进行标注（分类）。

该模型可以建模大量的自然语言处理任务，包括：分词、词性标注、命名实体标注等，句法分析和语义角色标注等也可以使用该问题建模。

$$[x_1, x_2, ..., x_n] \longrightarrow [y_1, y_2, ..., y_n]$$

字的标签集：b（begin），m（middle），e（End），s（Single）。

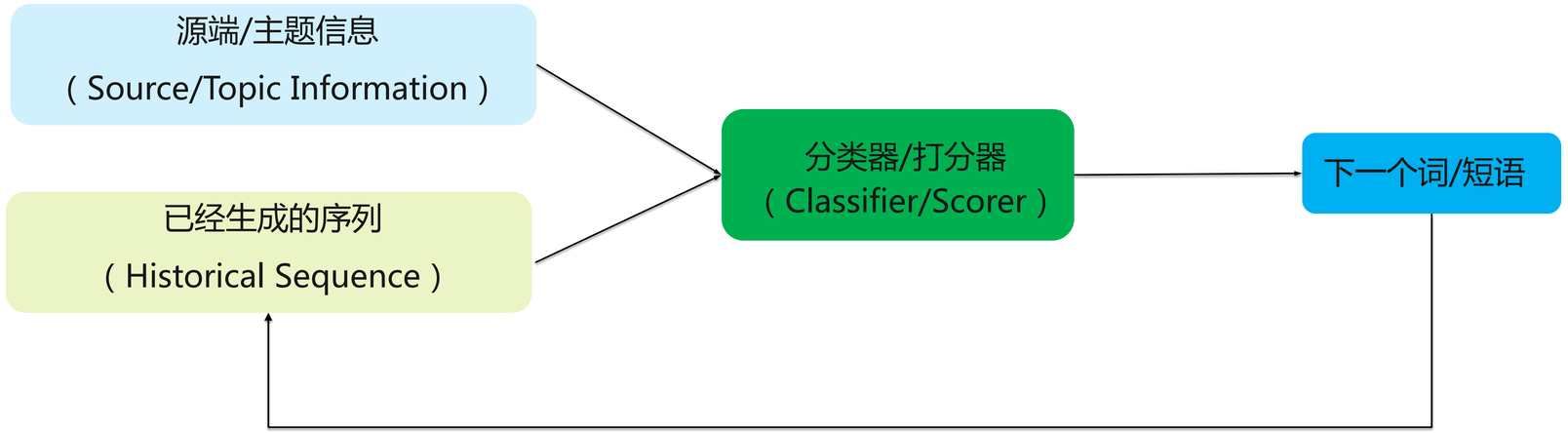


图：使用序列标注模型解决中文分词问题

序列标注模型常用的统计机器学习算法包括：隐马尔科夫模型（Hidden Markov Model，HMM），最大熵模型（Maximum Entropy Model，MEM），条件随机场（Conditional Random Fields，CRF），平均感知机模型（Average Perceptron，AP）等。

自然语言处理中的常用问题模型—序列生成模型

序列生成模型（Sequence Generation Model）：



图：序列生成模型

在深度学习方法应用到自然语言处理领域中之前，并没有很好的方法来建模序列生成问题。一般使用语言模型（Language Model）来做单语的序列生成，使用统计机器翻译模型(Statistical Machine Translation)实现双语的序列生成。

分类器或打分器要根据多个特征来确定分数/概率最高的下一个词/短语，这个过程通常会使用束解码（beam search）的方式，以约束搜索空间。

统计机器学习算法模型的不足

- 繁琐的特征工程（Complex Feature Engineering）
- 算法模型对序列建模能力弱（Weak Sequence Modeling）
- 流水线式的搭建导致错误传播（Error Propagation）

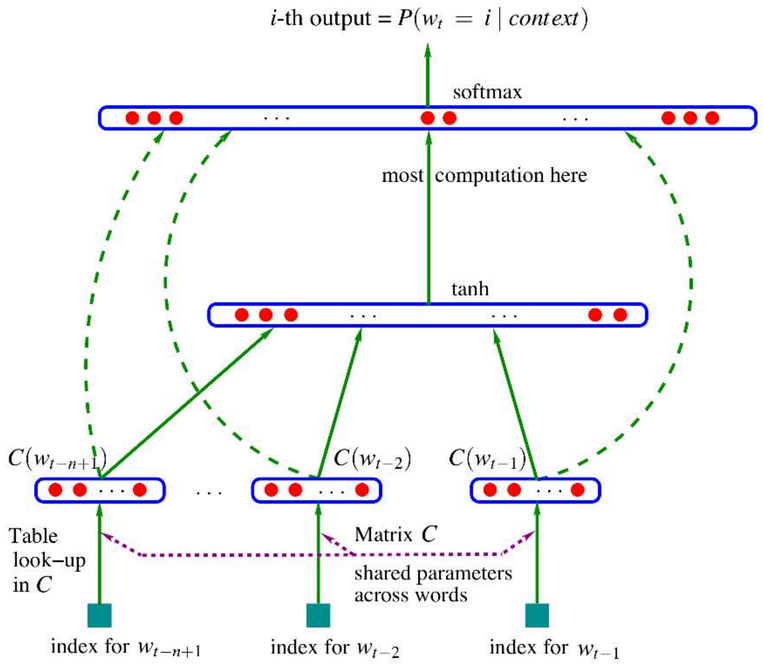
深度学习基本组件（Components）

- **前向神经网络（Feedforward Neural Network）**
- **循环神经网络（Recurrent Neural Network，RNN）**
  - 长短期记忆网络（Long short-Term Memory Neural Network，LSTM）
  - 门控循环单元网络（Gated Recurrent Unit Neural Network，GRU）
- **卷积神经网络（Convolutional Neural Network, CNN）**
- **注意力机制（Attention Mechanism）**

强大的深度学习方法—分布式表示

使用分布式表示绕开复杂的特征工程：

词的分布式表示（Distributed Representation/Word Embedding/Word Representation）



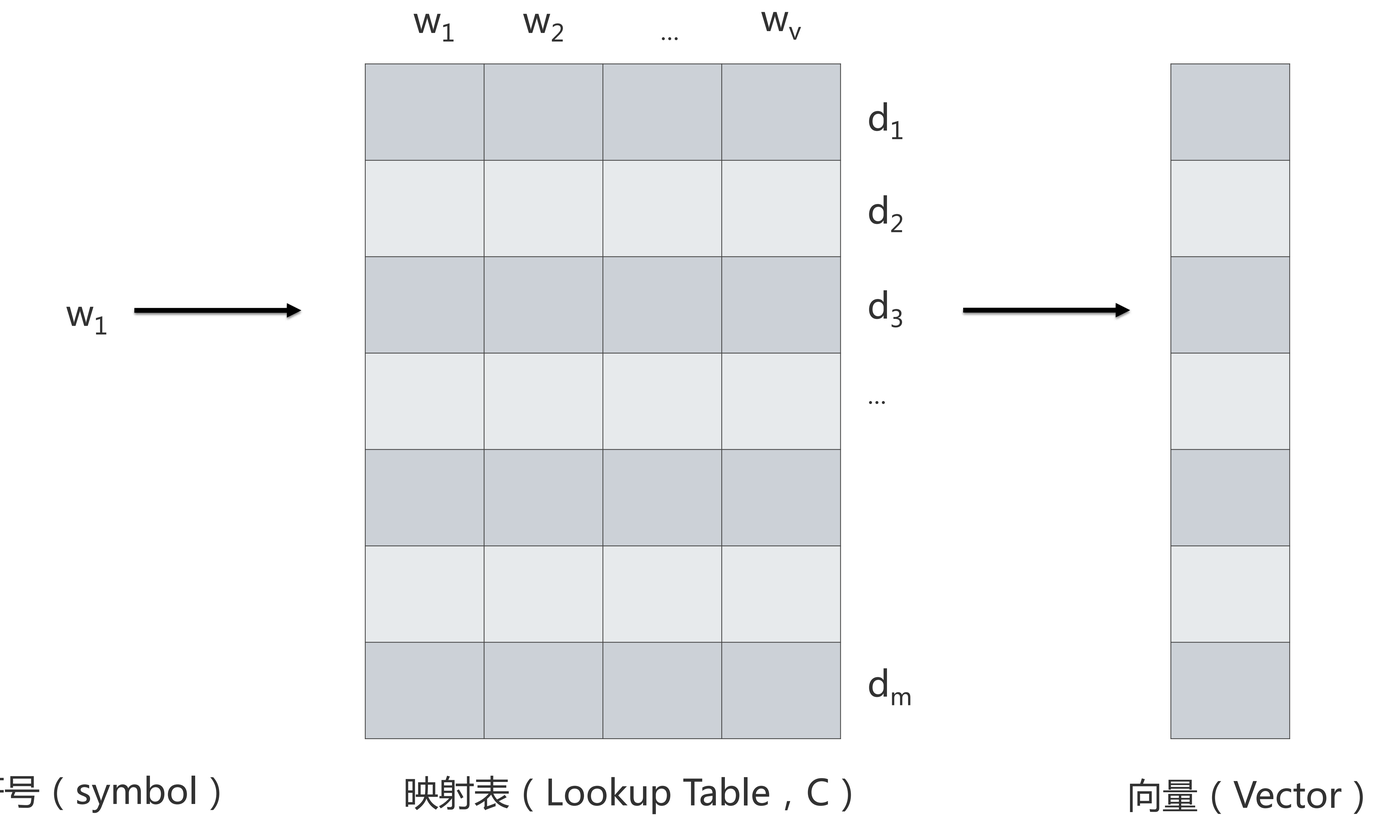
图：Bengio 提出的神经网络语言模型

Bengio Y, Ducharme R, Vincent P, et al. A neural probabilistic language model[J]. Journal of machine learning research, 2003, 3(Feb): 1137-1155.

强大的深度学习方法—分布式表示

使用分布式表示绕开复杂的特征工程：

词的分布式表示（Distributed Representation/Word Embedding/Word Representation）



符号向量化的好处：

- 克服维度爆炸的问题。
- 可以直接进行数值计算。
- SGD自动特征学习。

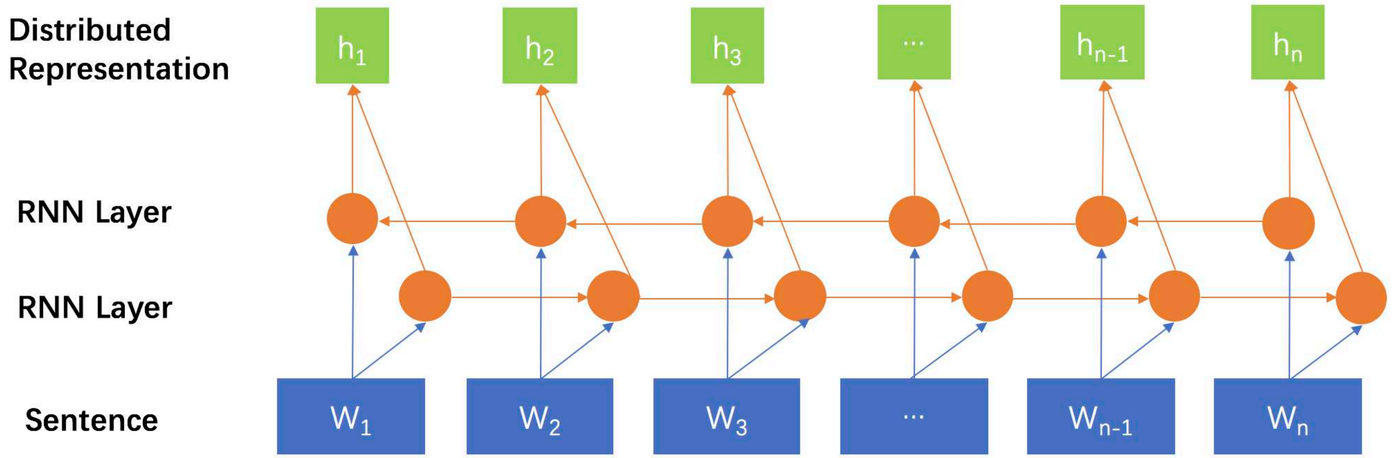
图：符号的分布式表示

强大的深度学习方法—序列建模方法

腾讯云 | 微信

RNN/CNN/Self-attention等的序列建模方法：

以词汇层面的分布式表示为基础，可以构建句子(序列)的分布式表示。这个过程中句子中的词汇表示发生交互运算，最后的序列分布式表示可以认为是句子的数值化的全局特征。



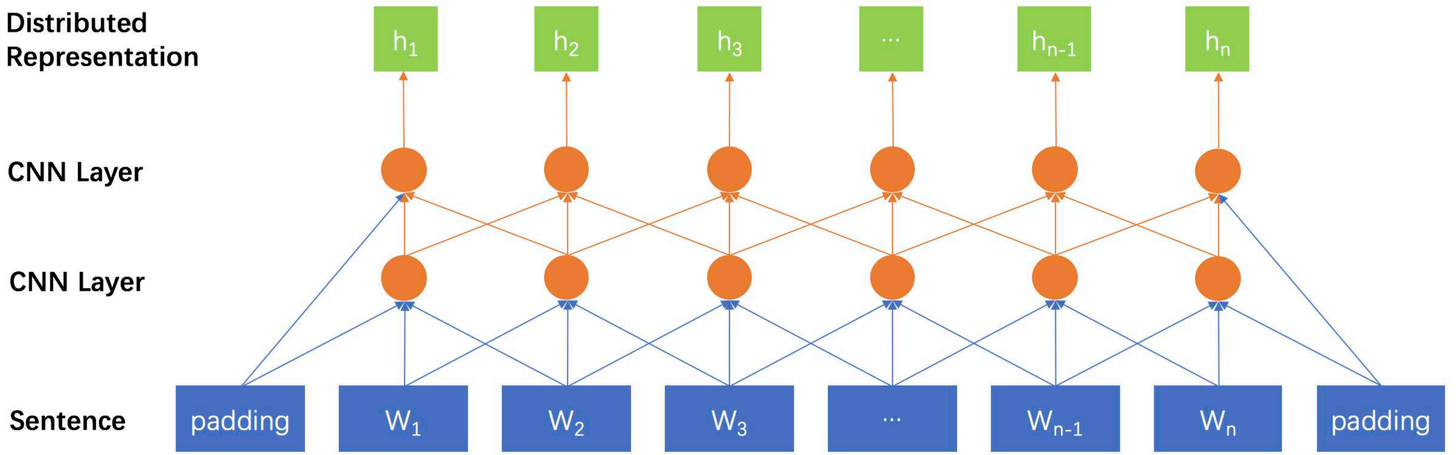
图：基于双向RNN的序列建模过程

RNN结构非常适合建模序列问题，双向RNN可以捕获词左边和右边的语义环境。

强大的深度学习方法—序列建模方法

RNN/CNN/Self-attention等的序列建模方法：

以词汇层面的分布式表示为基础，可以构建句子(序列)的分布式表示。这个过程中句子中的词汇表示发生交互运算，最后的序列分布式表示可以认为是句子的数值化的全局特征。



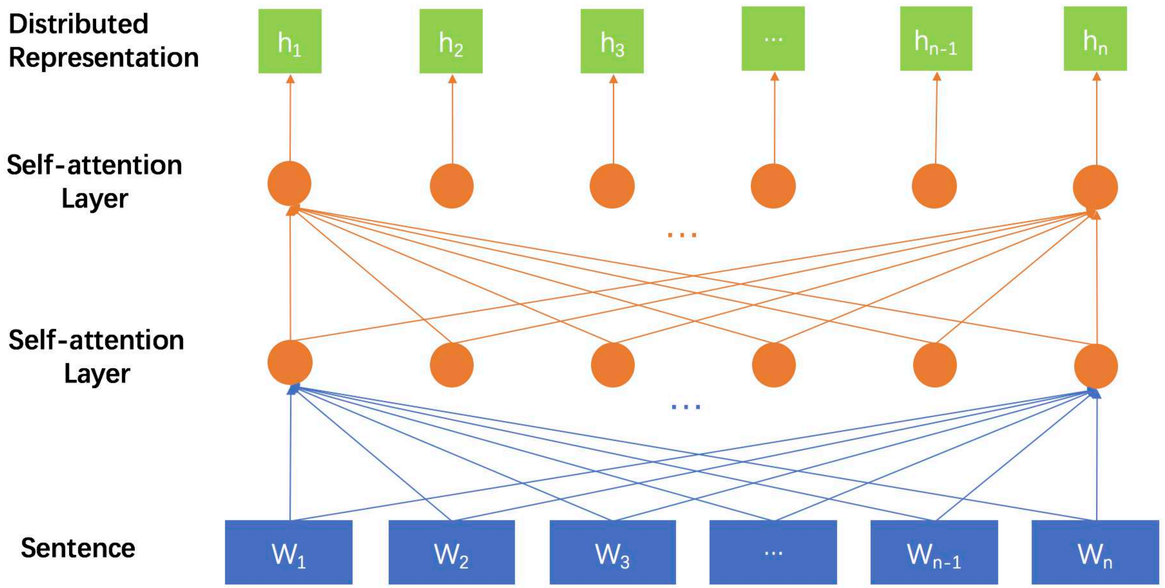
图：基于CNN的序列建模过程

强大的深度学习方法—序列建模方法

腾讯云 | 微信

RNN/CNN/Self-attention等的序列建模方法：

以词汇层面的分布式表示为基础，可以构建句子(序列)的分布式表示。这个过程中句子中的词汇表示发生交互运算，最后的序列分布式表示可以认为是句子的数值化的全局特征。

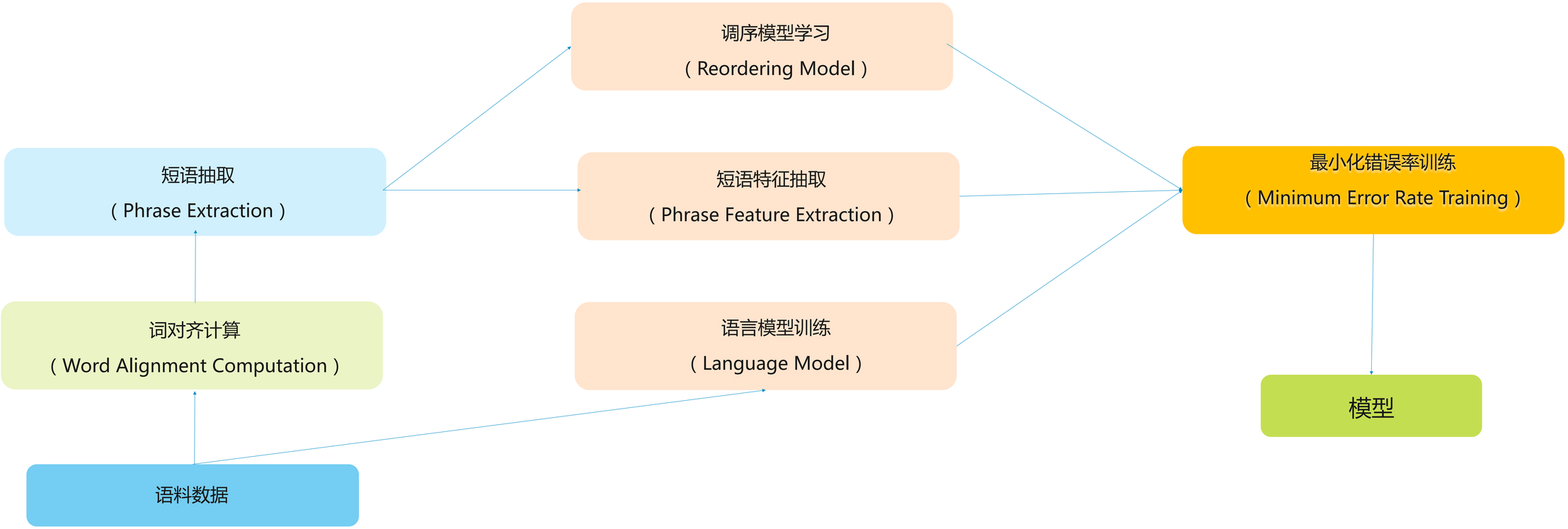


图：基于Self-Attention的序列建模过程

强大的深度学习方法—参数统一优化

参数统一优化：

端到端（End-to-End）的问题建模方式把所有的参数统一到一个优化优化目标下，缓解了流水线式搭建系统导致的错误传播问题。

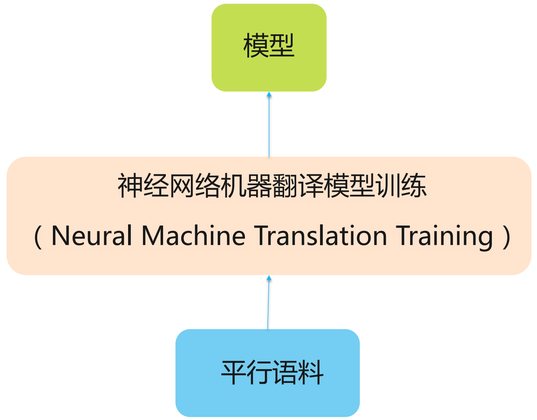


图：统计短语机器翻译模型搭建流程

强大的深度学习方法—参数统一优化化

参数统一优化：

端到端（End-to-End）的问题建模方式把所有的参数统一到一个优化优化目标下，消除了流水线式搭建系统导致的错误传播问题。

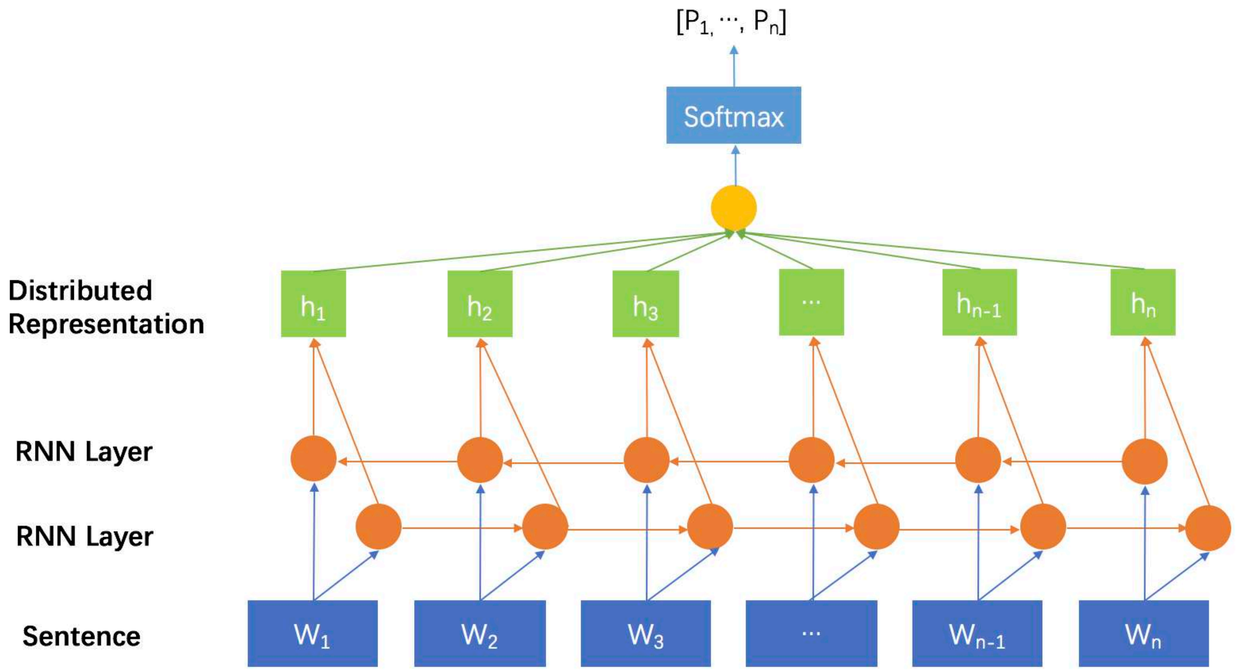


图：神经网络机器翻译模型搭建流程

使用深度学习方法解决分类问题

源端表示作为分类器的输入特征：

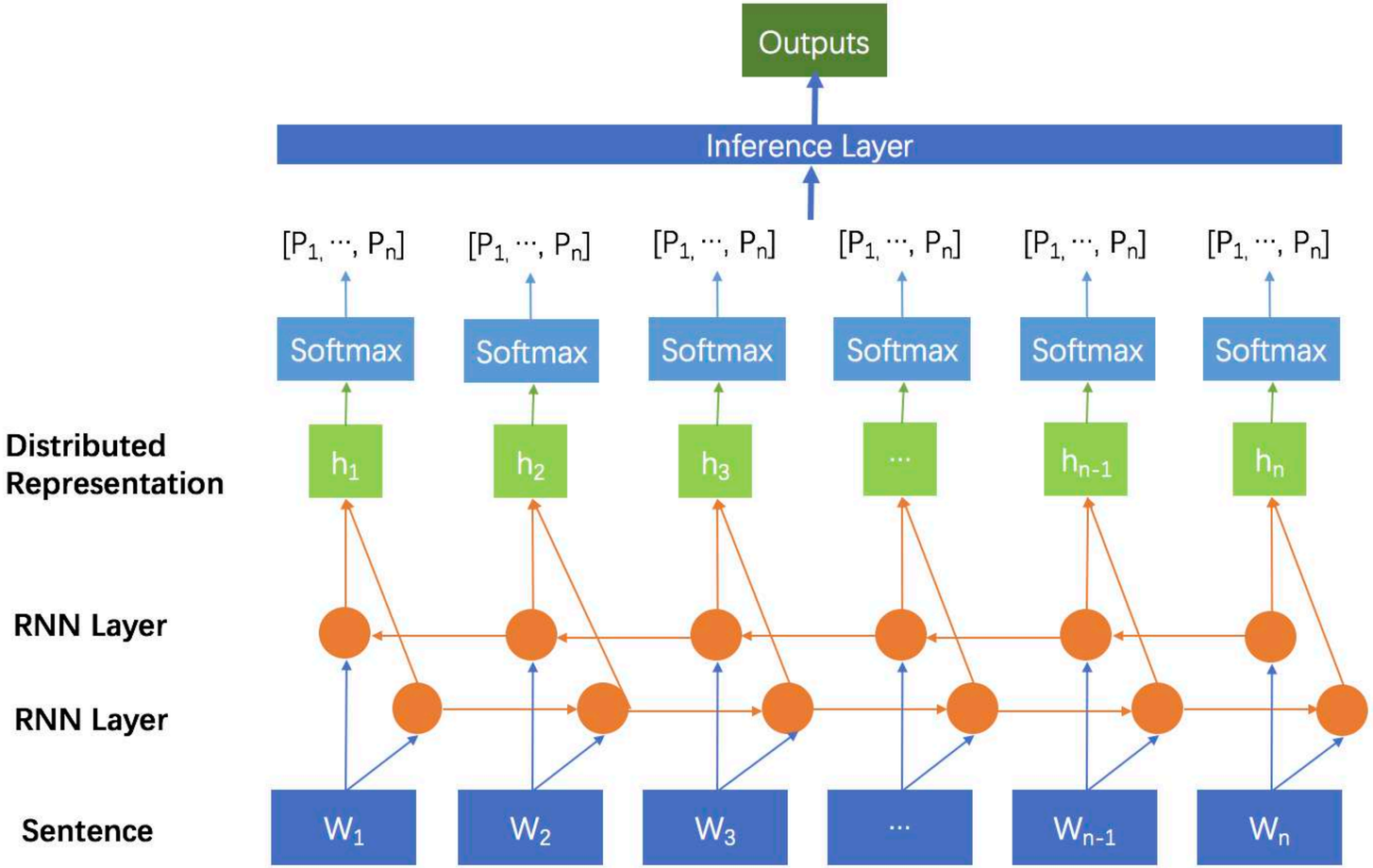
对序列良好的建模可以使得特征更符合当前的任务。



图：双向RNN解决分类问题

使用深度学习方法解决序列标注问题

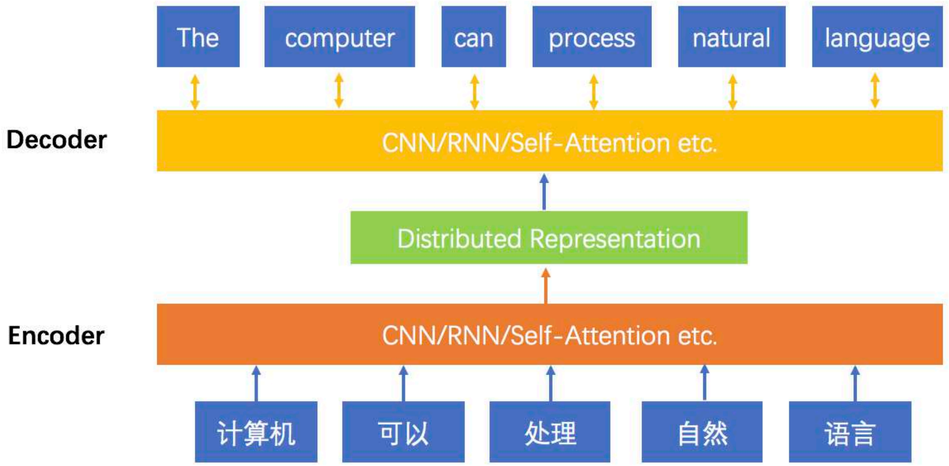
深度学习方法解决序列标注问题：



图：双向RNN解决序列标注问题

使用深度学习方法解决序列生成的问题

Encoder-Decoder模型：



图：Encoder-Decoder模型

- 编码器（Encoder）对源句子进行分布式表示。
- 解码器（Decoder）根据源句子的分布式表示生成目标句子。

深度学习方法的缺点

- 模型可解释性低，较难结合语言学/人类知识。
- 计算资源需求较重。
- 模型的表现除了依赖于本身的结构，还依赖于较多的训练技巧。

小结

本章介绍了自然语言处理中的常用问题模型和算法模型，对比了统计机器学习方法和深度学习方法。要点包括：

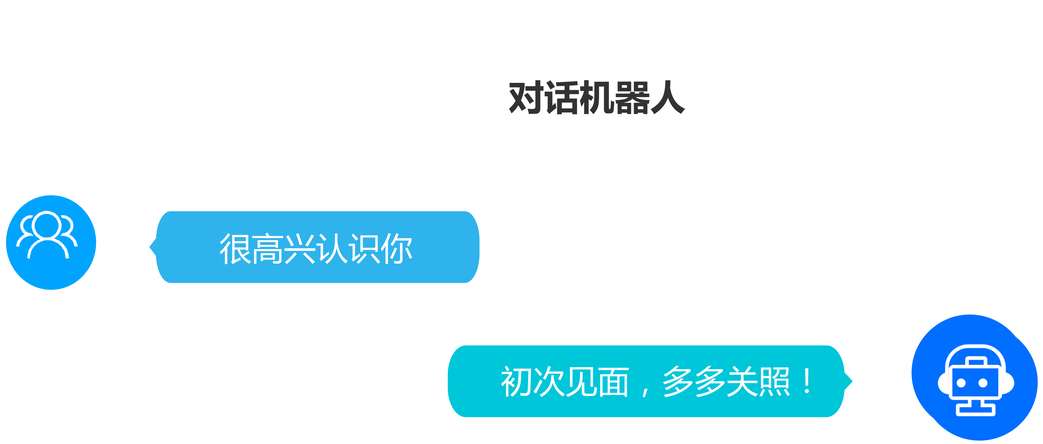
- 自然语言处理方法从规则式方法演进到统计机器学习方法，到现在的深度学习方法。
- 分类模型、序列标注模型、序列生成模型是自然语言处理中常用的问题模型。
- 深度学习方法与统计机器学习方法相比，存在诸多优势。

目录

机器翻译和对话系统中的序列到序列模型

机器翻译（Machine Translation）&对话系统中的闲聊模块（Chatbot）：

- 序列到序列问题是序列生成问题中的一种。
- 机器翻译任务是给出一个句子，返回该句子的译文。闲聊任务是给出一个用户话语句子，返回对该句子的回复。
- 可以使用编码器-解码器框架来建模机器翻译和闲聊问题。



机器翻译和对话系统中的序列到序列模型

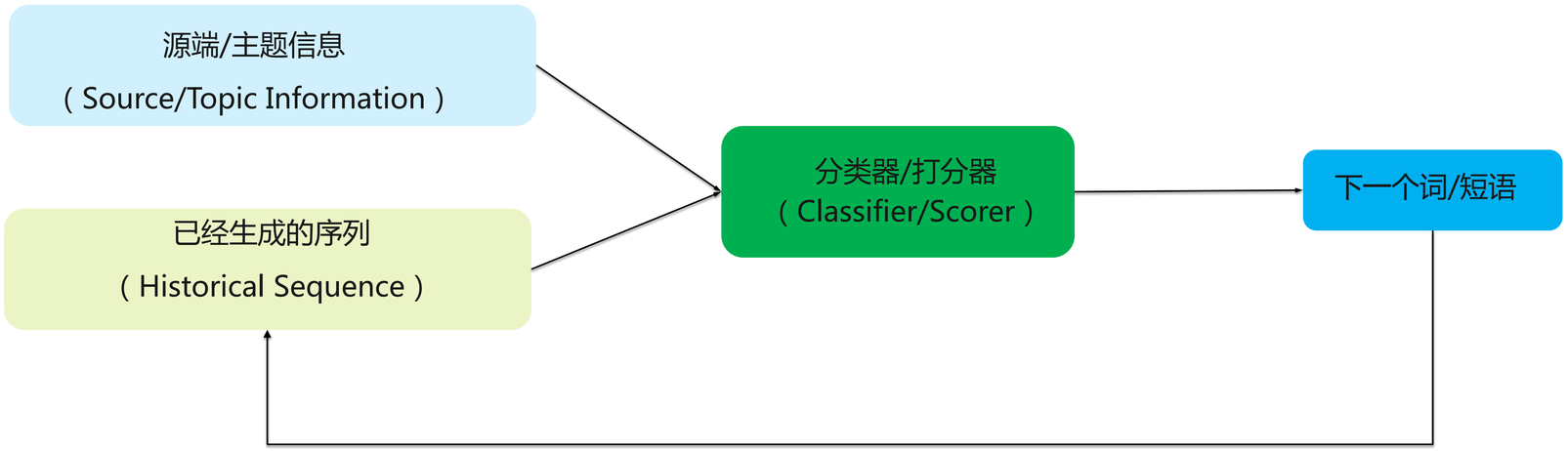
问题形式化：

源句子（Source Sentence）：$X = [x_1, x_2, ..., x_m]$

目标句子（Target Sentence）：$Y = [y_1, y_2, ..., y_n]$

翻译/回复概率（Translation/Rely Probability）：$P(Y|X) = \prod_{t=1}^{n} p(y_t|y_{<t}, X)$

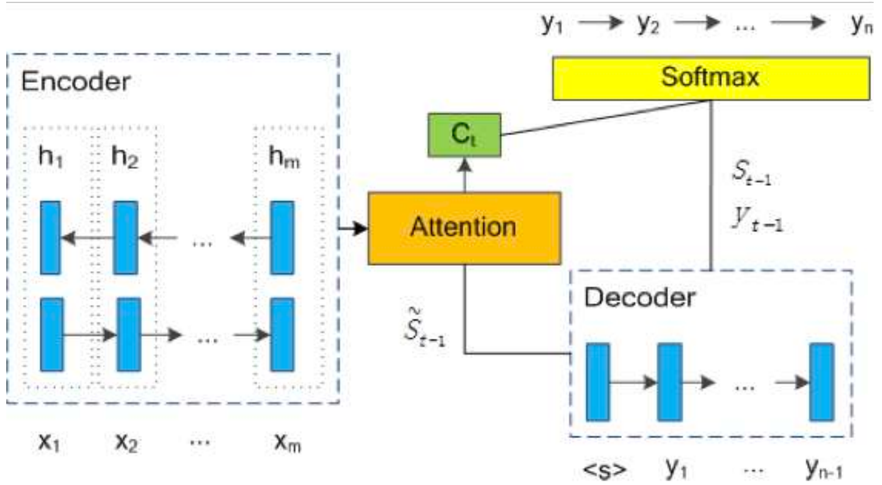
目标函数是最大化训练语料中所有句对的对应概率



图：序列生成模型

机器翻译和对话系统中的序列到序列模型

基于注意力机制的Encoder-Decoder模型：



图：RNNSearch模型

对源句子进行表示： $[h_1, h_2, ..., h_m] = Encoder([x_1, x_2, ..., x_m])$

Attention计算： $c_t = \sum_{j=1}^{m} \alpha_{t,j} h_j$

目标端词的预测： $p(y_t|y_{<t}, X) = g(y_{t-1}, c_t, s_t)$

Bahdanau D, Cho K, Bengio Y. Neural Machine Translation by Jointly Learning to Align and Translate. Iclr 2015 . 2014;1–15. Available from: http://arxiv.org/abs/1409.0473

机器翻译和对话系统中的序列到序列模型

检索视角下的注意力机制（Attention Mechanism）：

- attention是一种query机制，即用一个query来检索一个memory区域。
- query可以表示为key_q，memory是一个M项的键值对集合，其中的第i项我们表示为（key_m_i, value_m_i）。
- relation函数计算query和key_m_i的相关度，决定查询结果中value_m_i的权重。
- 注意，这里的key_q，key_m，value_m都是vector。

相关度计算： $relation(query, memory) = [relation(key\_q, key\_m_1), ..., relation(key\_q, key\_m_M)]$

相关度归一化： $$P = [p_1, ..., p_M] = softmax(relation(query, memory))$$

memory加权平均： $$result = \sum_{i=1}^{M} p_i * value\_m_i$$

注意：Encoder-Decoder结构中attention，query是Decoder的隐状态，memory是对源端的表示，memory中的key和value是相同的。

机器翻译和对话系统中的序列到序列模型

广义的Encoder-Decoder框架：

- Encoder和Decoder可以分别选用RNN、CNN、Self-attention结构等。
- Attention机制可以选用基于加法的或是基于乘法的，全局的或是局部的。
- 可以融合更多的输入特征。
- 可以增加额外的记忆空间。
- 可以在框架中融入更多的功能模块。
- 不变的是，Encoder用来做表示，Decoder用来做生成。
- 目前自然语言处理的泛文本生成任务都基于该框架来做。

机器翻译系统VS对话系统

机器翻译：

- 机器翻译任务知识较为固定，源端句子和目标端句子在语义上具有强一致性关系，是一个非常标准的序列到序列任务。
- 可以通过学习大规模的平行语料来覆盖近乎全量的翻译现象。

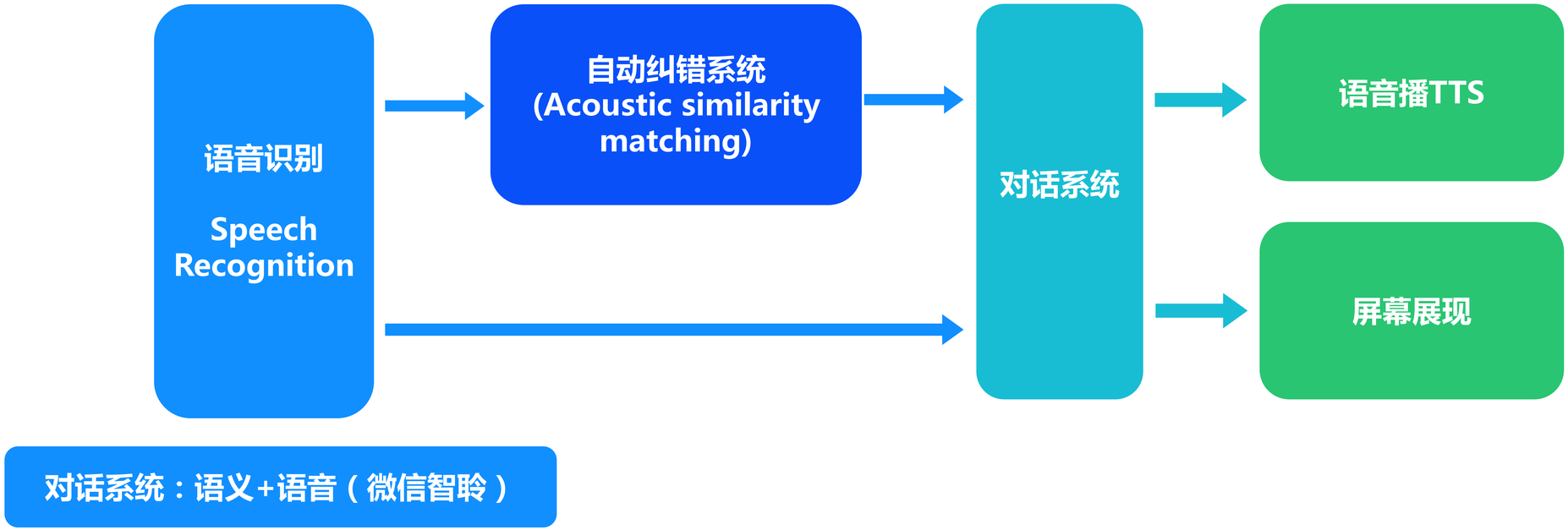
对话系统：

- 对话任务，场景复杂，源端句子和目标端句子在语义上只是存在相关性关系。
- 大规模语料只能覆盖部分对话现象。
- 语料规模变大时会因为回复多样性导致知识冲突。
- 使用序列到序列的建模方式，可以搭建一个chatbot，但没法搭建一套实用的面向任务的对话系统（Task-oriented Dialogue system）。
- 对话系统需要知识库、多媒体内容平台等支持。

一套实用的任务对话系统—小微对话系统

图：小微中的部分自然语言处理任务

一套实用的任务对话系统—小微对话系统



图：对话系统及其周边支持系统

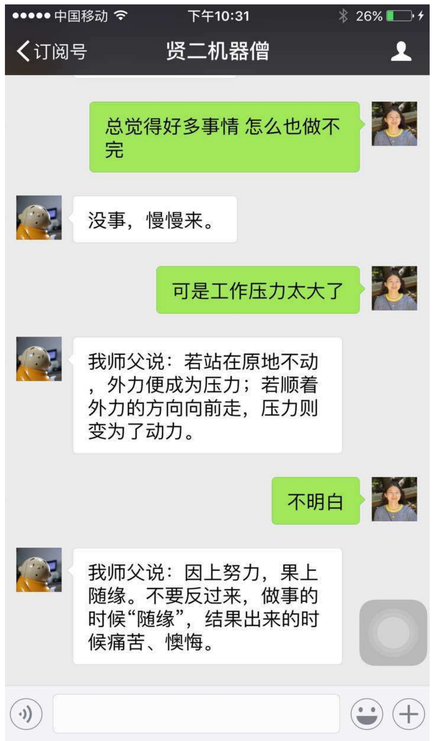
标杆案例

基于FAQ的自动客服

Messaging
(Can be automated)

Account management

基于知识的对话机器人：贤二机器僧



标杆案例

车载：音乐与导航

外交部12308助手

外交部 12308 微信版

标杆案例


Asus Zenbo
海美迪视听机器人
优必选QRobot Alpha
Harman Kardon Allure
叮咚二代
出门问问
若琪月石
长安A800

腾讯云AI能力免费接入

腾讯云 | 微信

语音识别类

1.语音识别
2.语音合成
3.机器翻译
4.一句话识别
5.录音文件识别
等...

计算机视觉类

1.人脸检测与分析
2.人脸验证
3.五官定位
4.活体检测
5.文字识别类
6.图片标签
等...

扫码进入免费领用页面

目录

开发者技能进阶建议—基础篇

打牢数学基础：

- 线性代数（矩阵运算）
- 概率论（概率计算）
- 高等数学（函数、导数，级数，公式推导）

熟练使用一种深度平台：

- Python
- TensorFlow
- PyTorch
- …

开发者技能进阶建议—基础篇

一些公开课和Blogs：

- Dan Jurafsky & Chris Manning：Natural language Processing
- Standford CS224d：Deep Learning for Natural Language Processing
- Coursera: Introduction to Natural Language Processing
- …

https://blog.csdn.net/CoderPai/article/details/71195143

开发者技能进阶建议—进阶篇

研读现有优秀系统：

- Word2Vec
- GNMT
- Tensor2Tensor
- …

培养问题建模能力：

- 对问题和模型比较熟悉。
- 能够根据问题和数据情况选择合适的模型。

模型实现：

- 在深度学习平台上实现模型。
- 模型性能分析与调优。

开发者技能进阶建议—创新篇

培养创新能力：

- 阅读论文并思考，对领域的的研究现状和方法有清晰的认识。
- 看清问题的本质，并尝试提出新的解决方法。
- 以合理的实验验证新方法的能力。
- ...

THANKS !

自然语言处理任务在产品应用中的难点

有限的标注数据（Restricted Labeled Data）：

1. 基于数据驱动的有监督自然语言处理方法，极度的依赖于标注数据。
2. 人工数据标注的成本较高，导致训练数据有限，对问题覆盖程度低。
3. 标注的过程中可能因为标注者对任务理解的偏差，存在不一致（噪音）。
4. 针对某一垂直领域的特定任务进行标注的数据，很难迁移到其他任务或领域上去使用。

复杂的应用场景（Complex Scenario）：

1. 自然语言处理技术应用在人机交互场景里时，需要面对比较复杂甚至异常的情况。
2. 基于数据驱动的机器学习模型往往对场景覆盖不足，一定概率会返回低质量的结果。
3. 规则式的方法是对机器学习模型很好的补充，能够提升产品鲁棒性，但需要更多的人力成本。