中文分词算法调研报告

TAL AI Lab 赖敏材

2018 年 4 月 8 日

摘要

本文中文分词做了简要的分析，分析了当前中文分词的一些难点，如歧义、粒度标准、分词标准等。接着本文总结了中文分词的一些常见方法，包括基于词典的分词技术、基于统计的机器学习的分词方法等。同时，也介绍了当前中文分词的研究进展和方向，如基于深度学习的分词方法、统计和词典结合的分词方法等。然后对主流的中文分词工具在权威数据集 SIGHAN Bakeoff 2005 MSR，SIGHAN Bakeoff 2005 PKU 上做了测评。最后，本文给出了相应的参考文献以及其他资料。

目录

1 导论

1.1 自然语言简介

众所周知，语言是人们日常生活的核心部分，任何与语言问题相关的工作都会显得非常有意思。在英语环境中，语言处理研究这一领域通常被简称为 NLP。对语言有深入研究的人通常被叫作语言学家，而“计算机语言学家”这个专用名词则指的是将计算机科学应用于语言处理领域的人。因此从本质上来说，一个计算机语言学家应该既有足够的语言理解能力，同时还可以用其计算机技能来模拟出语言的不同方面。虽然计算机语言学家主要研究的是语言处理理论，但 NLP 无疑是对计算机语言学的具体应用。

NLP 多数情况下指的是计算机上各种大同小异的语言处理应用，以及用 NLP 技术所构建的实际应用程序。在实践中，NLP 与教孩子学语言的过程非常类似。其大多数任务（如对单词、语句的理解，形成语法和结构都正确的语句等）对于人类而言都是非常自然的能力。但对于 NLP 来说，其中有一些任务就必须要转向标识化处理、语块分解、词性标注、语法解析、机器翻译及语音识别等这些领域的一部分，且这些任务有一大部分还仍是当前计算机领域中非常棘手的挑战[1]。

1.2 中文分词问题

中文信息处理是指自然语言处理的分支，是指用计算机对中文进行处理。在 NLP 任务中，分词是自然语言处理的基础过程，只有将句子段落分词，我们才能用文本去做其他任务。对于不同的语言，由于其语言结构的组成以及语义的不同，其分词方法也不相同。例如，世界上使用最广泛的语言—英语，其最基本的组成单位就是词，所以分词方法很简单，最常用的方法就是按指定的分隔符（空格）来分词。对于中文而言，最小的单位是字，但是具有语义的最小单位却是词，并且，中文文本是由连续的字符序列组成，词与词之间没有天然的分隔符，所以相对来说，中文分词困难很多。

中文自动分词的一个重要前提是以什么标准为词的分界，词是最小的能够独立运用的语言单位。词的定义非常抽象且不可计算。中文的词语有 2 字词语、3 字词语、4 字词语的分类，并且，有些短词语是长词语的前缀，故而，给定某文本，按照不同的标准的分词结果往往不同。词的标准成为分词问题一个很大的难点，没有一种标准是被公认的。但是，换个思路思考，若在同一标准下，分词便具有了可比较性。因此，只要保证了每个语料库内部的分词标准是一致的，基于该语料库的分词技术便可一较高下。

分词的难点主要在于分词粒度的选择和歧义。分词歧义主要包括以下几方面：

1. 交集歧义：A、X、B 分别为汉字串，如果其组成的汉字串 AXB 满足 AX 和 XB 同时为词，则汉字串 AXB 为交集型歧义字段。
   例如：“研究生命的起源”分词的结果为：
   研究/ 生命/ 的/ 起源
   研究生/ 命/ 的/ 起源
   此处，字符串” 研究生命” 是交集歧义字段
2. 组合歧义：汉字串 AB 满足 A、B、AB 同时为词，则该汉字串为组合型歧义字段。
   例如：“他从马上下来”分词结果为：
   他 / 从 / 马 / 上 / 下来

他 / 从 / 马上 / 下来
此句，字符串” 马上下来” 是组合歧义字段

3. 未登录词：字符串在字典中不存在，从未见过的词语
例如：”蔡英文和特朗普通话“分词结果为：
蔡英文 / 和 / 特朗普 / 通话
蔡英文 / 和 / 特朗 / 普通话
此句，字符串” 特朗普” 是未登录词

4. 上下文歧义：由自然语言的二义性所引起的歧义。
例如：“兵乓球拍卖完了”的分词结果为：
乒乓 / 球拍 / 卖 / 完 / 了
乒乓球 / 拍卖 / 完 / 了
此句，上下文歧义，必须根据上下文环境来分析

其中，第 4 种类型，是由自然语言的二义性所引起的歧义，就是人工分词也会产生歧义，只有结合上下文才能给出正确的切分。因此，又叫做第一类歧义; 第 1 和第 2 种类型，主要是由机器自动分词产生的特有歧义，称为第二类歧义; 第 3 种类型，是由于分词词典的大小引起的歧义，称为第三类歧义。

统计表明第一类歧义字段只占歧义字段总数的 5% 左右，剩下来的就都是第二类歧义字段和第三类歧义字段。故自动分词阶段对歧义的研究主要集中在对第二类、第三类歧义字段的研究。而对于第二类歧义，这其中交集型歧义又占了绝大多数，据统计达 94%，因此处理好交集型歧义在汉语分词中有着非常重要的地位[3]。

中文的最小单位是字，其分词粒度最小也是词，但是分词粒度越小，其所蕴含的语义也就越不清晰，但是分词粒度越大，容易过度表达语义，因此，分词粒度也是难题。研究者们往往把” 结合紧密、使用稳定” 视为分词单位的界定准则，然而人们对于这种准则理解的主观性差别较大，受到个人的知识结构和所处环境的很大影响。故而，在面对不同的任务的时候，选择不同的分词粒度，例如，对于机器翻译，有以下的信息来做参考：

1. 一般情况下，对于需要进行语义语法分析的中文片段来说，分词的粒度越大它所能表达的含义也就越确切。
2. 切分结果中单字数和非字典词越少越好。
3. 分词结果中，总体词数越少越好。

然而，在中文搜索引擎中，小的颗粒度比大的颗粒度好[4]。

2 中文分词技术详解

在前一章中，已经详细的介绍了中文分词的难点，在本章中，接下来详细介绍，现在比较重要的中文分词技术：

1. 基于词典的中文分词技术
2. 基于统计的机器学习技术
3. 基于词典和统计的分词技术
4. 基于深度学习的分词技术

2.1 基于词典的中文分词技术

基于词典分词算法也称作字符串匹配分词算法，该算法是按照一定的策略将待匹配的字符串和一个已建立好的“充分大的“词典中的词进行匹配，若找到某个词条，则说明匹配成功，识别了该词。梁南元在 1983 年发表的论文《书面汉语的自动分词与另一个自动分词系统 CDWS》提到，苏联学者 1960 年左右研究汉俄机器翻译时提出的 6-5-4-3-2-1 分词方法。其基本思想是先建立一个最长词条字数为 6 的词典, 然后取句子前 6 个字查词典，如查不到, 则去掉最后一个字继续查, 一直到找着一个词为止。梁南元称该方法为最大匹配法——MM 方法 (The Maximum Matching Method)。由 MM 方法自然引申，有逆向的最大匹配法。它的分词思想同 MM 方法，不过是从句子（或文章）末尾开始处理的，每次匹配不成词时去掉最前面的字。双向最大匹配法即为 MM 分词方法与逆向 MM 分词方法的结合。梁南元等人首次将 MM 方法应用于中文分词任务，实现了我国第一个自动汉语自动分词系统 CDWS[5]。

现在常见的基于词典的分词算法分为以下几种：正向最大匹配法、逆向最大匹配算法、双向匹配分词法和复杂最大匹配法等。

2.1.1 正向最大匹配算法

每次从头开始读取最大长度的字符串，然后，去词典中检查该词是否存在，如果不存在，则减少一个长度单位，再次读取字符串，检查该词是否在词典中，依次类推，直到找到一个词为止，然后，读取下一个词。

例如：假设存在词典：香港大学 | 香港 | 大学 | 校庆 | 典礼。对句子”香港大学校庆典礼“使用正向最大匹配算法分词。假设最大匹配长度为 5，从句子的开始位置读取 5 个长度的字符，得到”香港大学校“，查询词典，该词不存在，接着读取 4 个长度的字符，得到”香港大学“，查询词典，该词存在，则保留该词，接着，依次将剩余的字符串分词。最终的结果为：香港大学 | 校庆 | 典礼

2.1.2 逆向最大匹配算法

逆向最大匹配算法和正向匹配算法基本一致，唯一不同的地方在于，其匹配顺序从字符串的尾部开始匹配。

例如：假设存在词典：香港大学 | 香港 | 大学 | 校庆 | 典礼。对句子”香港大学校庆典礼“使用逆向最大匹配算法分词。假设最大匹配长度为 5，从句子的结束位置读取 5 个长度的字符，得到”学校庆典礼“，查询词典，该词不存在，接着读取 4 个长度的字符，得到”校庆典礼“，查询词

典，该词不存在，接着读取 3 个长度的字符串，得到”庆典礼“，查询词典，该词不存在，接着读取两个长度的字符串，得到”典礼“，查询词典，该词存在，则保留该词，接着，依次将剩余的字符串分词，最终的结果为：香港大学 | 校庆 | 典礼

2.1.3 双向最大匹配算法

在工程应用中，工程师会将正向最大匹配算法和逆向最大匹配算法结合，即：双向匹配算法，具体的过程是：对句子，采用两种方法分别进行分词处理，然后将两者结果，进行比对，如果分词结果一致，则直接输出，否则的话，按照以下原则输出：

1. 分词结果中，词越少，越优先输出。
2. 分词结果在词典中匹配的词越多，越优先输出。

如果正向和逆向都未通过两条原则，则优先输出逆向最大匹配的结果。

2.1.4 复杂最大匹配算法

复杂最大匹配算法，由 Chen 和 Liu 在《Word identification for Mandarin Chinese sentences》提出[6]。该文提出了三词语块（three word chunks）的概念。三词语块生成规则是：在对句子中的某个词进行切分时，如果有歧义拿不定主意，就再向后展望两个汉语词，并且找出所有可能的三词语块。在所有可能的三词语块中根据如下四条规则选出最终分词结果。

1. 最大匹配 (Maximum matching)
   其核心的假设是：最可能的分词方案是使得三词语块 (three-word chunk) 最长。
2. 最大平均词长（Largest average word length）
   在句子的末尾，很可能得到的” 三词语块” 只有一个或两个词（其他位置补空），这时规则 1 就无法解决其歧义消解问题，因此引入规则 2：最大平均词长，也就是从这些语块中找出平均词长最大的语块，并选取其第一词语作为正确的词语切分形式。这个规则的前提假设是：在句子中遇到多字词语的情况比单字词语更有可能。
3. 最小词长方差（Smallest variance of word lengths）
   还有一些歧义是规则 1 和规则 2 无法解决的。因此引入规则 3：最小词长方差，也就是找出词长方差最小的语块，并选取其第一个词语作为正确的词语切分形式。在概率论和统计学中，一个随机变量的方差描述的是它的离散程度。因此该规则的前提假设是：句子中的词语长度经常是均匀分布的。
4. 最大单字词语语素自由度之和（Largest sum of degree of morphemic freedom of one-character words）
   有可能两个” 三词语块” 拥有同样的长度、平均词长及方差，因此上述三个规则都无法解决其歧义消解问题。规则 4 主要关注其中的单字词语。直观来看，有些汉字很少作为词语出现，而另一些汉字则常常作为词语出现，从统计角度来看，在语料库中出现频率高的汉字就很可能是一个单字词语，反之可能性就小。计算单词词语语素自由度之和的公式是对” 三词语块” 中的单字词语频率取对数并求和。规则 4 则选取其中和最大的三词语块作为最佳的词语切分形式。

最大匹配算法以及其改进方案是基于词典和规则的。其优点是实现简单，算法运行速度快，缺点是严重依赖词典，无法很好的处理分词歧义和未登录词。因此，很长的一段时间内研究者都在对基于字符串匹配方法进行优化，比如，未登录词的识别、最大长度设定、采用 TEIR 索引树等。

2.2 基于统计的机器学习技术

针对基于词典的机械切分所面对的问题，尤其是未登录词识别，使用基于统计模型的分词方式能够取得更好的效果。基于统计模型的分词方法，简单来讲就是一个序列标注问题。在一段文字中，我们可以将每个字按照他们在词中的位置进行标注，常用的标记有以下四个 label：B，Begin，表示这个字是一个词的首字；M，Middle，表示这是一个词中间的字；E，End，表示这是一个词的尾字；S，Single，表示这是单字成词。分词的过程就是将一段字符输入模型，然后得到相应的标记序列，再根据标记序列进行分词。举例来说：“达观数据位是企业大数据服务商”，经过模型后得到的理想标注序列是：“BMMESBEBMEBME”，最终还原的分词结果是“达观数据/是/企业/大数据/服务商”。

随着基于统计的分词方法理论的发展，形成了不少相关的分词模型互信息、N 元文法模型、条件随机场、隐马尔可夫模型等。这些统计模型充分利用词与词的共现概率作为分词的依据，分词的有效性和准确率能够最大可能地满足文本分类需要[7]。

统计的分词模型有以下优点：

1. 有较强的数学理论做基础
2. 运用大规模语料库更容易，大规模语料库能提供足够的实例模型化知识。
3. 如果训练的语料足够大，能更客观反映语言学中的规律。
4. 一致性好。
5. 统计方法处理自然语言的健壮性好，能够覆盖的范围较大。

但是统计的方法也并不是十全十美的，其有三点不足：

1. 对自然语言的处理和表示相对比较浅薄。
2. 需要大规模的准确涵盖面广的语料库来作为其数据支持，如果没有大的语料库，其效果比较差。
3. 表达的知识难理解。

2.2.1 隐马尔科夫模型

隐马尔科夫模型 (HMM) 属于统计模型的一种，它是由马尔科夫过程扩展的一种随机过程，其基本理论是由数学家 Baum 及其同事在上世纪六十年代末到七十年代初建立起来的，七十年代中后期应用于语音处理，广泛应用则是在八十年代，八十年代之后，人们将其应用到文本处理。在马尔可夫模型中每个状态代表了一个可观察事件，所以马尔可夫模型有时又称作是可视马尔可夫模型。这在某种程度上限制了模型的适应性。而在隐马尔可夫模型中，我们并不知道模型所经过的状态序列，只知道状态的概率函数，也就是说，观察到的事物是状态的随机函数，因此，改模型是一个双重随机过程。其中，模型的状态转换过程是不可观察的，即隐蔽的，可观察时间的随机过程是隐蔽的状态转换过程的随机函数[8]。

隐马尔科夫模型是关于时序的概率模型，描述由一个隐藏的马尔科夫链随机生成不可预测的状态随机序列，再由哥哥状态生成一个观测而产生观测随机序列的过程，隐藏的马尔科夫链随机生成的状态的序列，称为状态序列，每个状态生成一个观测，而由此产生的观测的随机序列，称为观测序列，序列的每一个位置又可以看作是一个时刻。

隐马尔科夫模型由初始概率分布（π）、状态转移概率分布 (A) 以及观测概率分布确定 (B)。而这三者 (π，A，B) 又称作隐马尔科夫模型的三要素。隐马尔科夫模型的基本模型可以由下图表示：

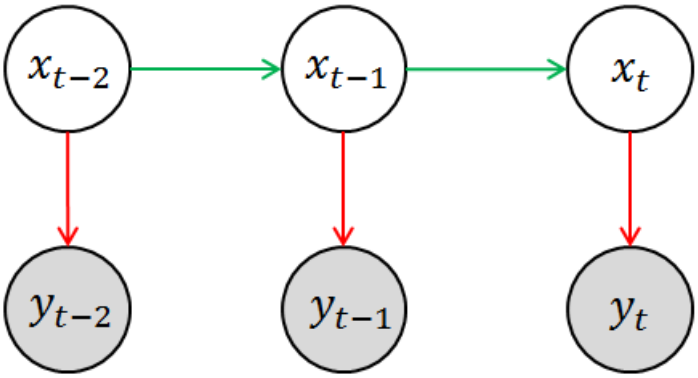

图 1: HMM 基本模型

在中文分词中，一段文字的每个字符可以看作是一个观测值，而这个字符的词位置 label（BEMS）可以看作是隐藏的状态。使用 HMM 的分词，通过对切分语料库进行统计，可以得到模型中 5 大要要素：起始概率矩阵，转移概率矩阵，发射概率矩阵，观察值集合，状态值集合。在概率矩阵中，起始概率矩阵表示序列第一个状态值的概率，在中文分词中，理论上 M 和 E 的概率为 0。转移概率表示状态间的概率，比如 B->M 的概率，E->S 的概率等。而发射概率是一个条件概率，表示当前这个状态下，出现某个字的概率，比如 $P(\ |B)$ 表示在状态为 B 的情况下人字的概率。有了三个矩阵和两个集合后，HMM 问题最终转化成求解隐藏状态序列最大值的问题，求解这个问题最长使用的是 Viterbi 算法，这是一种动态规划算法。

2.2.2 条件随机场

条件随机场（CRF）是以最大熵马尔科夫模型和隐马尔科夫模型为基础的一种判别式概率无向图学习模型，是一种用户标注和切分有序数据的条件模型。CRF 自提出以来，就被广泛的应用到自然语言处理和图像处理等领域。CRF 是用来标注和划分序列结构数据的概率化结构模型。言下之意, 当对于给定的输入观测序列 X 和输出序列 Y，CRF 通过定义条件概率 P(Y|X)，而不是联合概率分布 P(X，Y) 来描述模型。

常见的统计模型可以分为以隐马尔可夫模型为代表的生成模型, 和以最大熵模型为代表的判别模型。隐马尔可夫模型的假设条件为某时刻的观测值依刻的隐藏状态, 同时各个观察值是相互独立的, 但实际条件下这一假设通常很难满足。相对于隐马尔可夫模型,CRF 的主要优点在于它的条件随机特征, 只需要考虑当前已经出现的观测状态特征, 有独立性的严格要求。而相对于最大熵模型和其它针对线性序列模型条件隐马尔可夫模型会出现条件偏置 (label bias) 赖于该时问题。CRF 避免了这个问题，对于整个序列内部的信息和外部观测信息都可以有效地应用。CRF 具有最大熵模型的一切优点, 两者的关键区别在于, 最大熵模型使用每一个当前状态的指

数模型来计算给定状态的下一个状态的条件概率，而 CRF 用单个指数模型来计算给定观察的整个标记序列的联合概率。因此在不同的状态不同特征的权重下可以相互交替代换。

条件随机场模型相对于隐马尔可夫模型有着明显的优点。隐马尔可夫的主要缺点在于，在含有诸多特征信息的任务中，不能对输入序列中的任意、相互依赖性的特征进行建模，由于模型本身的假设，观测变量 Xi 只依赖于其对应的标记变量 Yi。所以我们在建模的过程中也统计了和 (Xi,Yi)、Yi 相关的信息，这意味着在进行模型推理、对标记序列预测的时候，我们不会用到 Xi 之外的 X 的变量信息。而条件随机场模型可以融合各种特征到模型中，而且特征容易更换，可以灵活地选择特征供进行分词应用，从而克服了隐马尔可夫模型强独立性假设条件。同时克服了最大熵模型的标记偏置。

CRF 具有那么多的优点，其缺点也不能忽视，CRF 模型与以往模型相比的缺点主要集中在两个方面，首先模型的训练十分复杂，相比于隐马尔科夫模型模型，其参数更多，故而其时间、空间复杂度都远超过以往的模型。其次，模型相较于以往的最大熵模型，缺乏特征选择机制，目前对于由语料库中产生的特征一般通过根据其出现频率设定阀值以进行筛选。

2.2.3 结构化感知器

结构化感知器（Structured Perceptron, SP）是由 Collins 在 EMNLP'02 上提出来的，用于解决序列标注的问题。中文分词工具 THULAC、LTP 所采用的分词模型便是基于此。CRF 全局化地以最大熵准则建模概率 P(Y|X)；其中，X 为输入序列 x_1^n，Y 为标注序列 y_1^n。不同于 CRF 建模概率函数，SP 则是以最大熵准则建模 score 函数：$S(Y,X) = \sum_s \alpha_s \Phi_s(Y,X)$ SP 解决序列标注问题，可视作为：给定 X 序列，求解 score 函数最大值对应的 Y 序列：$\arg\max_Y S(Y,X)$ 为了避免模型过拟合，保留每一次更新的权重，然后对其求平均。具体流程如下所示：

The Structured Perceptron with Averaging

- Input: labeled examples, $\{(\underline{x}^i, \underline{s}^i)\}_{i=1}^n$.
  Initialization: $\underline{w} = \underline{0}$, $\underline{w}_a = \underline{0}$
- For $t = 1 \ldots T$, for $i = 1 \ldots n$:
  - Use the Viterbi algorithm to calculate

$$\underline{s}^* = \arg\max_{\underline{s} \in \mathcal{Y}} \underline{w} \cdot \underline{\Phi}(\underline{x}^i, \underline{s}) = \arg\max_{\underline{s} \in \mathcal{Y}} \sum_{j=1}^{m} \underline{w} \cdot \underline{\phi}(\underline{x}, j, s_{j-1}, s_j)$$

  - Updates:

$$\begin{aligned} \underline{w} &= \underline{w} + \underline{\Phi}(\underline{x}^i, \underline{s}^i) - \underline{\Phi}(\underline{x}^i, \underline{s}^*) \\ &= \underline{w} + \sum_{j=1}^{m} \underline{\phi}(\underline{x}, j, s_{j-1}^i, s_j^i) - \sum_{j=1}^{m} \underline{\phi}(\underline{x}, j, s_{j-1}^*, s_j^*) \\ \underline{w}_a &= \underline{w}_a + \underline{w} \end{aligned}$$

- Return $\underline{w}_a / nT$

图 2: 结构化感知器具体算法

2.3 基于词典和统计的分词技术

基于统计的方法在分词性能方面有了很大的提升，但是在跨领域方面都有很大的不足，它们需要针对不同的领域训练不同的统计分词模型。这样导致在领域变换后，必须为它们提供相应领域的分词训练语料，但是分词训练语料的获得是需要大量人工参与的，代价昂贵。而基于词典的方法却在领域自适应方面存在着一定优势，当目标分词领域改变时，只需要利用相应领域的词典即可。领域词典的获取相比训练语料而言要容易很多。如果把这两种方法结合起来，使得统计的方法能够合理应用词典，则可实现中文分的领域自适应性。

2012 年，张梅山等人提出了通过在统计中文分词模型中融入词典相关特征的方法，使得统计中文分词模型和词典有机结合起来。一方面可以进一步提高中文分词的准确率，另一方面大大改善了中文分词的领域自适应性。

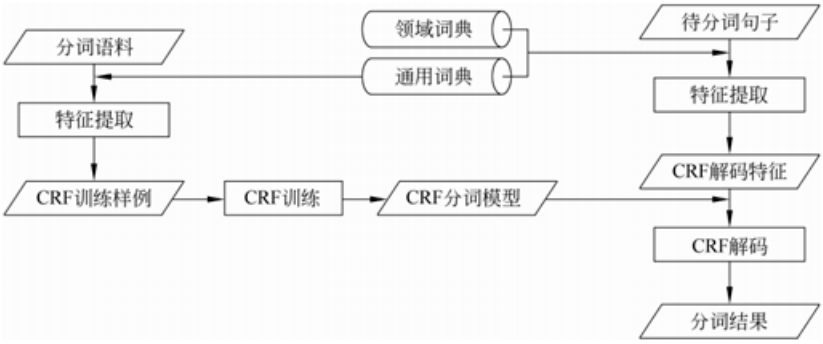

图 3: 领域自适应性框架图

2.4 基于深度学习的分词技术

近年来，深度学习在人工智能的多个领域取得了显著成绩。微软使用的 152 层深度神经网络在 ImageNet 的比赛上斩获多项第一，同时在图像识别中超过了人类的识别水平。百度在中文语音识别上取得了 97% 的准确率，已经超过了人类的识别能力。

随着深度学习在越来越多的领域中取得了突破性进展，自然语言处理这一人工智能的重要领域吸引了大批的研究者的注意力。最近谷歌发布了基于深度学习的机器翻译 (GNMT)，和基于短语的机器翻译相比，错误率降低了 55%-85% 以上，从而又引发了深度学习在自然语言处理上的热潮。

使用深度学习的方法进行中文分词。分词的基础思想还是使用序列标注问题，将一个句子中的每个字标记成 BEMS 四种 label。模型整的输入是字符序列，输出是一个标注序列，因此这是一个标准的 sequence to sequence 问题。因为一个句子中每个字的上下文对这个字的 label 类型影响很大，因此考虑使用 RNN 模型来解决。在传统的神经网络中，从输入层到隐藏层到输出层，层之间是全连接的，但是每层内部的节点之间是无连接的。因为这样的原因，传统的神经网络不能利用上下文关系，而在自然语言处理中，上下文关系非常重要，一个句子中前后词并不独立，不同的组合会有不同的意义，比如”优秀”这个词，如果前面是”不”字，则意义完全相反。RNN 则考虑到网络前一时刻的输出对当前输出的影响，将隐藏层内部的节点也连接起来，即当前时刻一个节点的输入除了上一层的输出外，还包括上一时刻隐藏层的输出。RNN 在理论上可以储存任意长度的转态序列，但是在不同的场景中这个长度可能不同。比如在词的预测例子中：1,“他是亿万富翁，他很?”; 2,“他的房子每平米物业费 40 元，并且像这样的房子他有十

几套，他很?”。从这两个句子中我们已经能猜到? 代表“有钱”或其他类似的词汇，但是明显，第一句话预测最后一个词时的上线文序列很短，而第二段话较长。如果预测一个词汇需要较长的上下文，随着这个距离的增长，RNN 将很难学到这些长距离的信息依赖，虽然这对我们人类相对容易。在实践中，已被证明使用最广泛的模型是 LSTM（Long Short-Term Memory，长短时记忆）很好的解决了这个问题。

LSTM 最早由 Hochreiter 及 Schmidhuber 在 1997 年的论文中提出。首先 LSTM 也是一种 RNN，不同的是 LSTM 能够学会远距离的上下文依赖，能够存储较远距离上下文对当前时间节点的影响。所有的 RNN 都有一串重复的神经网络模块。对于标准的 RNN，这个模块都比较简单，比如使用单独的 tanh 层。LSTM 拥有类似的结构，但是不同的是，LSTM 的每个模块拥有更复杂的神经网络结构：4 层相互影响的神经网络。在 LSTM 每个单元中，因为门结构的存在，对于每个单元的状态，使得 LSTM 拥有增加或减少信息的能力[10]。

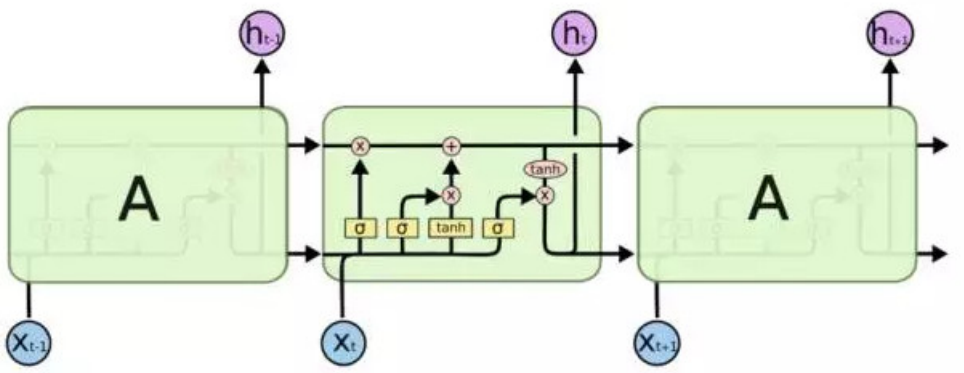

图 4: LSTM 标准模型

得益于模型和算法上的不断进步，如今 90% 乃至 95% 以上的中文分词准确率已不是什么难题。在传统 CRF 中，特征需要人工设定，因此大量繁杂的特征工程将不可避免。基于 LSTM 的网络发展给很多研究问题带来了全新的解决方案。在中文分词上，基于深度学习的方法，往往使用「字向量 + 双向 LSTM + CRF」模型，利用神经网络来学习特征，将传统 CRF 中的人工特征工程量降到最低，如图 5 所示，其中：

1. 字向量层（对应 Embedding Layer）能够把离散的汉字符号转化为连续的向量表示。
2. 双向 LSTM 网络（对应 Feature Layer）能够在考虑时序依赖关系的同时抽取有用的文本特征。
3. CRF 模型（对应 Inference Layer）则建模了两个相邻输出的概率制约关系强大的样本表示、特征抽取和概率建模能力，使它成为如今最主流的中文分词模型。

最近也有不少学者对这种框架作出了改进，进一步提高分词效果。比如复旦大学 NLP 团队于 2017 年发表在 ACL 上的论文提出了使用对抗神经网络（GAN）来利用多语料集进行中文分词的方案[11]。以往的分词方法是针对一个语料用一种分割标准进行分词，但是在一般情况下一个预料遵循一种分割标准。比如在两个语料对句子‘姚明进入总决赛’的分割标准就不同。如图 6 所以此文作者提出利用 GAN 与 LSTM 使同时利用多个分词标准的语料集来进行分词，该方案在 8 个语料库上均提高了准确率。框架代码已开源（https://github.com/FudanNLP/adversarial-multi-criteria-learning-for-CWS）。深度学习方法虽然训练需要较长的时间，但可以一次训练好

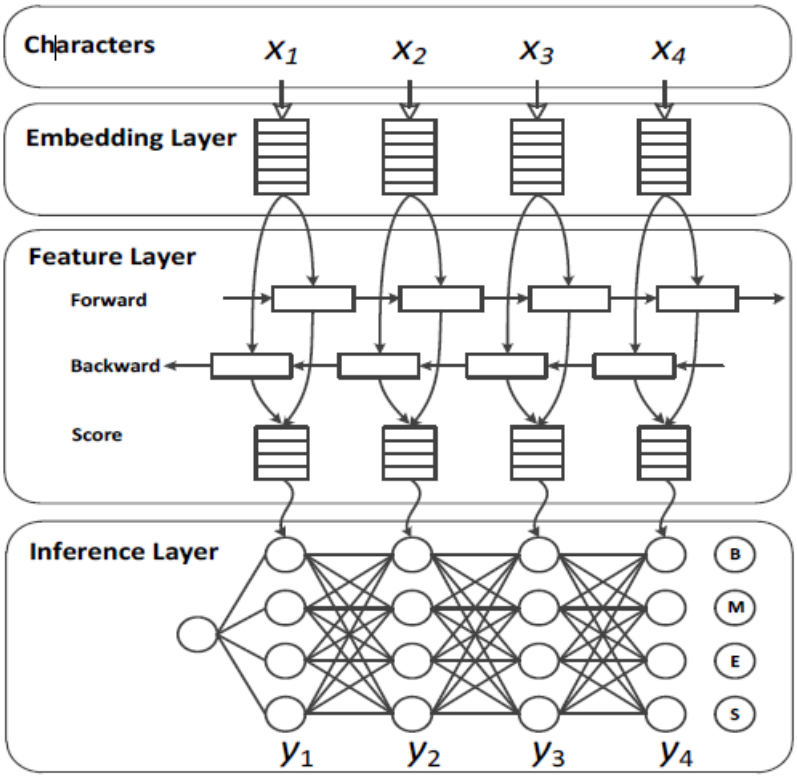

图 5: 基于深度学习的中文分词框架

Corpora	Yao	Ming	reaches	the final	
CTB	姚明		进入	总决赛	
PKU	姚	明	进入	总	决赛

图 6: 不同分割标准下的分词差异

存储模型进行调用，在速度上也不会低于传统的方法，未来深度学习在分词方面预计还会大放光彩。

2.4.1 一个 Bi-lstm 网络的实现

使用 bi-directional LSTM 模型训练后对句子进行预测得到一个标注的概率，再使用 Viterbi 算法寻找最优的标注序列。网络模型如下：该模型分词效果不错，分词精确率与召回率不逊色

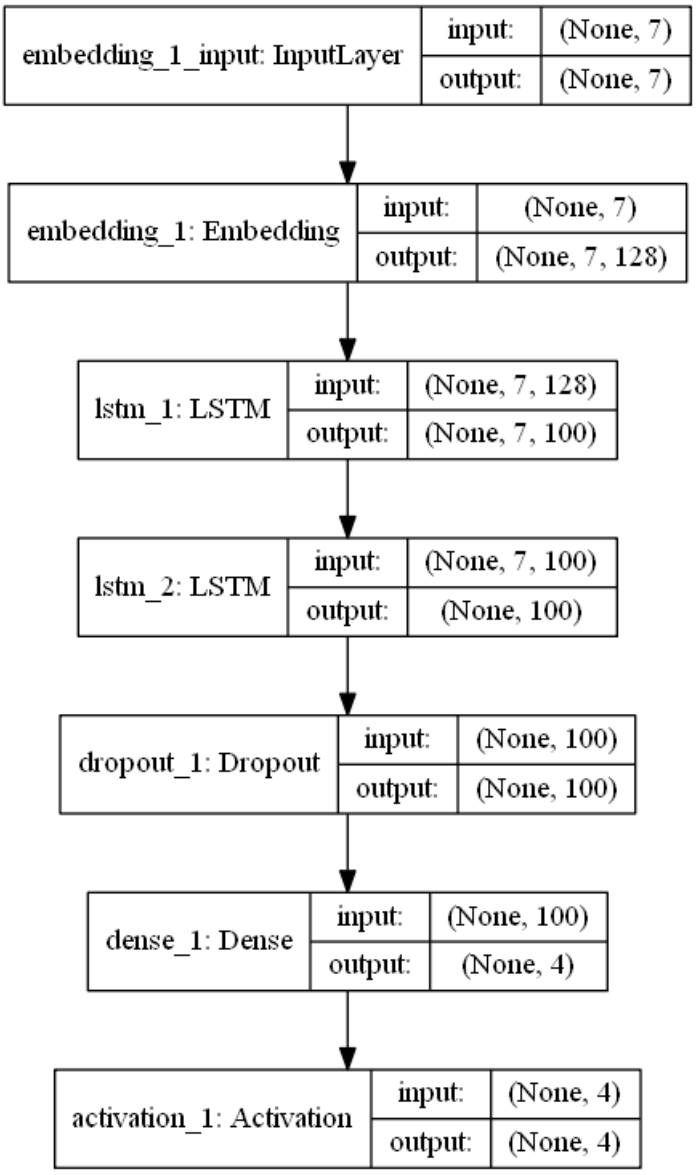

图 7: 基于深度学习的中文分词框架

于主流分词框架，但速度较慢。

3 中文分词工具简介

现在市面上有很多的中文分词工具，不同的分词器，由于其实现的算法不一样，故而，其分词效果也就不同。其中，LTP 与 THULAC 是基于 SP 算法，Jieba 与 NLPIR 是基于 HMM 算法，FoolNltk 是基于 bi-lstm 深度神经网络。

1. 哈工大 LTP https://github.com/HIT-SCIR/ltp
2. 中科院计算所 NLPIR https://github.com/NLPIR-team/NLPIR
3. 清华大学 THULAC https://github.com/thunlp/THULAC
4. jieba https://github.com/yanyiwu/cppjieba
5. FoolNltk https://github.com/rockyzhengwu/FoolNLTK
6. bilstm https://github.com/clayandgithub/rnn_cws

3.1 评测方法

就目前来说，对于中文分词器的评测还没有一个非常完整的评测评价指标，常见的评价指标有准确率，召回率和分词速度等，除此之外，还需要考虑分词器本身的一些评价，如功能完备性，可维护性，以扩充性和可移植性等。在本次测评中，主要测试分词器的准确率，召回率，未登录词 OOV 的召回率，已登录词 IV 的召回率和分词速度，测评数据集如下：

1. SIGHAN Bakeoff 2005 MSR, 560KB
2. SIGHAN Bakeoff 2005 PKU, 510KB

该数据集是 SIGHAN 于 2005 年组织的中文分词比赛所用的数据集，也是学术界测试分词工具的标准数据集，用于测试各大分词工具的准确性。测评方法为 SIGHAN Bakeoff 2005 比赛中所自带的 score 脚本、test gold 数据和 training words 数据对 4 个工具进行准确性测试[12]。

3.2 测评环境

1. 处理器：Intel(R) Core(TM) i7-7700 CPU 3.60GHZ
2. 内存：12.0 GB
3. 系统类型：64 位
4. IDE: jupyter notebook
5. Python 版本：python 3.6.4

注：所有库均使用其 python 版本，运行时间包括按行读入测试集与写入分词结果的时间

3.3 测评结果

3.3.1 MSR 测试结果

表 1: MSR 测评结果

TOOLS	PRECISION	RECALL	F MEASURE	OOV Recall Rate	IV Recall Rate	TIME(s)
bilstm	0.946	0.952	0.949	0.952	0.995	26.381
foolnltk	0.846	0.885	0.865	0.884	1.000	15.104
jieba	0.817	0.812	0.815	0.810	1.000	0.555
pyltp	0.868	0.899	0.883	0.898	1.000	1.184
pynlpir	0.914	0.869	0.891	0.913	1.000	0.404
thulac-python	0.933	0.925	0.929	0.877	1.000	2.180

3.3.2 PKU 测试结果

表 2: PKU 测试结果

TOOLS	PRECISION	RECALL	F MEASURE	OOV Recall Rate	IV Recall Rate	TIME(s)
bilstm	0.901	0.902	0.901	0.897	0.976	23.381
foolnltk	0.926	0.924	0.925	0.919	0.988	12.529
jieba	0.853	0.787	0.818	0.781	0.870	0.656
pyltp	0.960	0.946	0.953	0.943	0.990	1.992
pynlpir	0.940	0.944	0.942	0.941	0.990	0.463
thulac-python	0.921	0.923	0.922	0.919	0.984	3.230

3.4 测试结论

1、一个好的分词工具不应该只能在一个数据集上得到不错的指标，而应该在各个数据集都有很不错的表现。从这一点来看，thulac、ltp、foolnltk、nlpir 都表现非常不错

2、基于 bi-lstm 模型的分词方法运行时间最长，其次是基于 sp 算法的 ltp、thulac，耗费时间最少的是基于 hmm 的 jieba、nlpir

3、大家都知道，基本的分词依赖模型，但真正想用分词工具来解决应用层面上的问题，都需要借助于词库，本文测试的 5 个工具均支持用户自定义词库。

4、深度学习方法对 OOV 的召回率高于传统机器学习算法

参考文献

[1] llhthinker, 中文分词研究入门, 博客地址:http://www.cnblogs.com/llhthinker/p/6323604.html.

[2] zoohua, 中文自动分词歧义类型, 博客地址:http://blog.csdn.net/zoohua/article/details/4567802.

[3] 张瑞, 教学资源自动文摘系统的研究与设计 [M], 北京交通大学.

[4] 梁南元, 书面汉语的自动分词与另一个自动分词系统 CDWS, 中国汉字信息处理系统学术会议, 桂林, 1983.

[5] Chen, K. J. and Liu S.H. Word identification for Mandarin Chinese sentences. Proceedings of the 14th International. Conference on Computational Linguistics. 1992.

[6] 颜军, 基于条件随机场的中文分词研究与应用 [M], 武汉理工大学.

[7] 李航, 统计机器学习方法.

[8] 张梅山, 统计与字典相结合的领域自适应中文分词, 中文信息学报, 2012 年第 02 期.

[9] ResysChina, 高翔, 深度学习将会变革 NLP 中的中文分词, https://www.leiphone.com/news/201608/IWvc75oJglAIsDvJ.html.

[10] Xinchi Chen, Zhan Shi, Xipeng Qiu, Xuanjing Huang. 2017. Adversarial Multi-Criteria Learning for Chinese Word Segmentation. arXiv preprint arXiv:1704.07556 .

[11] 黄翼彪, 开源中文分词器的比较研究 [M], 郑州大学.