北京大学学报(自然科学版)
Acta Scientiarum Naturalium Universitatis Pekinensis
doi: 10.13209/j.0479-8023.2014.002

C-TERN：一种基于 CFSA 的军事新闻文本时间信息处理算法

王伟[1,2]　赵东岩[2,†]　苏婷婷[1]

1. 武警工程大学信息安全重点实验室，西安 710086; 2. 北京大学计算科学与技术研究所，北京 100871;
† 通信作者, E-mail: zdy@pku.edu.cn

摘要　提出一种基于层叠有限状态自动机(CFSA)的中文军事文本时间表达式识别与规范化算法 C-TERN。C-TERN 首先利用成熟的分词工具识别出文本中的时间词，然后将从通用语言和军事语言中提取的时间表达式规则分成多层，逐层进行时间信息的精细识别；在规范化过程中，通过 4 个步骤分别对特殊时间表达式、简单时间表达式、时间段表达式和绝对/相对时间表达式进行推理计算和规范化。算法注意了规则集提取的正确性、规则之间冲突的消解以及匹配方式的合理性。在多个数据集上的实验结果显示，C-TERN 不但能有效地识别出标准时间、偏移时间和不确定性时间表达式，而且能完成对简单、特殊以及隐含的时间点、时间段和偏移时间的推理与规范化，能够满足军事文本时间信息的处理的需要。

关键词　自然语言理解；有限状态自动机；时间表达式；识别与规范化
中图分类号　TP391

C-TERN: A Temporal Information Processing Algorithm of Chinese Military News Story Based on Cascade Finite State Automata

WANG Wei[1,2], ZHAO Dongyan[2,†], SU Tingting[1]

1. Key Laboratory on Information Security, Engineering University of CAPF, Xi'an 710086; 2. Institute of Computer Science and Technology, Peking University, Beijing 100871; † Corresponding author, E-mail: zdy@pku.edu.cn

Abstract　The authors propose a new method C-TERN to recognize and normalize the temporal expression in military story based on cascade finite state automata. Firstly, C-TERN recognizes the temporal expression in military story, and layers the temporal information extracted from general language and military language, and recognizes the temporal by layer. Then, in the procedure of temporal expression normalization, C-TERN ratiocinates and normalizes the simple/specify time, duration time, absolute and relative temporal expression in four steps. The method pays special attention to the correctness of the regulation extraction, the dispelling of the collision between regulations, and the reasonability of the matching method. The experimental results on multi-information show that proposed method can recognize and normalize the absolute and relative temporal expression as well as the simple/specify time and duration time effectively. It can better meets the temporal information processing needs in military applications.

Key words　natural language processing; finite state automata; temporal expression; recognition and normalization

时间表达式是指表示时间信息的自然语言短语。时间表达式的识别是命名实体识别领域一项基础而重要的任务，在主题检测与跟踪、自动问答、机器翻译等领域具有广泛的应用。时间信息抽取是信息抽取的重要内容之一，在 MUC(Message Understanding Conferences) 和 ACE (Automatic

中国博士后科学研究基金(2012M521944)、国家自然科学基金(2012AA011101)和武警工程大学军事基础研究基金(WJY 201314)资助
收稿日期: 2013–06–16; 修回日期: 2013–09–18; 网络出版时间: 2013-11-11 10:25

Content Extraction)评测中, 都有相关子任务。2004年 ACE 的 TERN (Temporal Expression Recognition and Normalization)评测[1], 将这项任务分为两个子任务: 时间表达式识别和规范化, 并制定相应的标准——TIMEX2[2−3]。时间表达式的识别任务就是在文本中正确的检测和划分时间表达式的边界。时间表达式的规范化则是为其标注上相应的 TIMEX2 归一化属性值, 即进行时序语义标注。

英文时间信息抽取研究已经进行了多年[4−5], 技术比较成熟。与英文时间信息研究内容类似, 中文时间信息抽取也需要能识别中文文本中的时间表达式, 计算出该表达式所蕴含的时间信息, 以计算机能理解的结构及方式储存。但是与英文相比中文在语法和语义上有非常大的区别, 一些对于英文时间识别比较有效的方法并不能直接用于中文时间的处理。因此, 目前仍有许多工作致力于研究适用于中文环境的时间处理方法[6−10]。

在军事应用领域, 军事文本的自动处理、军事情报的自动获取、检索等通常都与时间相关。军事文本中的时间, 既有绝对时间(例如“12 月 22 日 21 时 40 分”), 也有相对时间(例如“两个月前”), 既有点时间(“上午 8 时”), 也有段时间(“不少于 2 小时”)。另外, 有些情况下还需要根据上下文环境利用时间逻辑推理技术为缺少时间状语的事件确定时间信息。例如, 给出偏移时间表达式(如“上个星期”、“一年之后”)和不确定性时间词(如“日前”, “入冬以来”)的准确时间。这些时间信息的表述既根植于日常语言, 又具有军事语言的特点。因此需要基于中文自然语言时间信息抽取技术, 研究适用于军事文本的时间处理方法, 通过识别、规范化以及推理, 准确理解其中的时间信息。

本文参照 TIMEX2 规范, 实现了一个能够同时适用于通用语言、军事语言的中文时间表达式标注、规范化和推理的工具 C-TERN。C-TERN 通过提取军、地新闻文本中的特殊时间、简单时间、时间段和绝对/相对时间的表达规则与特征, 基于层叠有限状态自动机实现时间表达式识别与规范化。在“人民日报标注语料库”、“TempEval”数据集和利用军事新闻构造的数据集上的实验结果显示, C-TREN 不但能够准确地识别出多种类型的时间信息, 而且具有规范化和初步推理能力, 可支持多种军事文本自动处理相关的应用。

1 相关工作

TERN 任务是 ACE 组织赞助的一项关于时间表达式识别和规范化的测评任务。TERN 任务主要分为两个部分: 时间表达式的识别和时间表达式的规范化。目前 TERN 测评涉及的语言包括阿拉伯语、英语和汉语。在英文时间信息处理方面, 以 Mani 等[4]为代表, 主要利用基于规则的方法和基于序列标注的机器学习方法对英文时间的识别和规范化做深入研究。Ahn 等[5]将识别与规范化任务相分离, 用机器学习方法进行识别, 并在机器学习识别的结果上进行规范化, 提高了识别和规范化的准确率与召回率。也有人将 TIMEX2 标签方案扩展到韩语[11]、法语和西班牙语[12], 并进行初步的探索性研究。

中文时间信息处理的研究也主要利用上述方法。早期的研究以 Li 等[13]和 Wu 等[14]为代表, 对金融类的新闻事件的时间信息进行研究, 主要利用基于规则的方法进行时间分析, 但并没有把识别任务和规范化任务分开。在中文词法分析系统 ICTCLAS[15] (Institute of Computing Technology, Chinese Lexical Analysis System)中, 时间词的识别是未登录词(命名实体)识别中的一部分, 系统利用层叠隐马尔科夫模型[16](Multi-layer Hidden Markov Model, MHMM), 能够识别常见的时间词, 但由于 ICTALAS 并非专用时间信息识别工具, 因此不能识别复杂的时间表达式。

在利用语言特征方面, 贺瑞芳等[6]提出基于依存分析和错误驱动的方法。他们首先以时间触发词为切入点, 利用依存关系递归地识别时间表达式, 然后采用错误驱动学习进一步增强识别效果, 根据错误识别结果和人工标注的差异自动地获取和改进规则。刘莉等[7]将语义角色(Semantic Roles, SR)特征用于构建特征向量, 然后采用条件随机场模型进行识别, 在 SemEval-2010 评测的 TempEval-2 任务数据上实验的结果显示, *F*-score 值达到 85.6%, 比未加入语义角色特征提高了 5.2%。朱莎莎等[8]通过分析中文时间短语的词法、句法和上下文信息等语言学特征, 将时间短语分为日期型和事件型两种, 并构造常用词表作为外部特征, 采用条件随机场(CRFs)整合不同层面的特征, 将识别问题转化为序列标注问题, 取得一定效果。

邬桐等[9]对基于机器学习的序列标注方法和基于规则的方法进行比较, 研究表明, 虽然序列标注

方法在命名实体识别领域占据主流地位, 但是条件最大熵和条件随机场模型都无法有效完成时间表达式识别任务。在中文时间表达式识别领域, 经典的基于规则的方法仍然是最为广泛使用和最有效的方法。他们基于“时间基元”进行规则构建, 并利用错误驱动思想进行规则剪枝。在 ACE07 中文语料上的实验结果显著超过了现有水平, *F*-score 达到 89.8%。

在时间规范化表示方面, 清华大学的林静等[10]依据 TIMEX2 的中文标注草案, 开发了一个基于正则表达式的时间信息自动标注的系统。系统采用三层候选确定参考时间, 并行使用多个模块识别输入句子中的时间短语, 通过排序冲突消解确定最终结果。正确识别和规范化后的时间信息, 能够支持基于时间的语义计算[17]、新闻事件间的时序关系、逻辑关系的分析和推理[18]等研究。

现有中文时间信息研究工作表明, 单独使用基于机器学习或规则匹配的方法通常不能得到满意结果, 应该探索两种途径的结合。同时, 由于汉语的复杂性, 所提取规则集的合理性、规则之间冲突的消解、错误的纠正等都是影响时间信息处理准确性的重要因素。因此, 本文利用成熟的基于机器学习算法的中文分词工具的标注结果, 结合提取的多种时间表达规则, 采用反向匹配、上下文检查等错误纠正策略, 实现时间表达式的准确识别与规范化。

2 基于 CFSA 的时间信息处理算法

有限状态自动机(finite state automata, FSA)具有概念简单、运行速度快、开发周期短等优点, 在基于规则的系统中有广泛的应用。本文基于层叠有限状态自动机(cascade FSA, CFSA), 将 ICTCLAS 分词与词性标注结果作为初始输入, 并逐步引入时间信息识别与规范化规则, 以第 n–1 层自动机的输出作为第 n 层自动机的输入, 逐层识别和规范化符合规则的时间短语, 实现一个基于CFSA的中文时间表达式识别与规范化算法 C-TREN, 框架如图 1 所示。

在编制 CFSA 中所使用的规则时, 我们遵循以下 3 个原则以保证正确性与合理性: 1) 分类设定语法规则, 根据时间短语的类型将规则分为多个子类, 例如简单时间、特殊时间、偏移时间等, 这样能够通过减小复杂性提高正确率。2) 利用上下文消解规则歧义, 针对一些容易混淆的语言环境, 提取专门表示时间信息的规则。3) 对于可能存在冲突的规则, 采用正、反向匹配相结合的方式, 并通过人工判断以概率更高的方式产生结果。

2.1 中文分词系统中的时间表达式识别

经过多年发展, 中文分词技术已经非常成熟, 以 ICTCLAS 为例, 2002 年 973 评测结果显示其分词的正确率高达 97.58%, 在 2003 年 ACL SIGHAN 国际汉语分词评测大赛中, ICTCLAS 取得了多个项目第一名的好成绩。但对时间表达式的识别来说, 仅仅基于机器学习的分词结果是不够的。

ICTCLAS 在词性标注的过程中, 会识别出有时间属性的词语。我们把这些被标注为时间属性的词语或短语初步作为含有时间信息的表达式。但 ICTCLAS 标注的结果并不能有效识别偏移时间的表达式和不确定性时间的表达式。例如“在 21 世纪的时候”的分词结果为“在/p 21/m 世纪/n 的/uj 时候/n”, 其中“21 世纪”并没有被识别为时间词。

值得注意的是, 在新闻文档中相对偏移时间与不确定性时间占将近一半的比重, 如果不处理这一类的时间表达式, 则很大一部分时间表达式就不能被正确识别和规范化。对于这种情况, 我们加入触发词识别的处理过程。根据整理好的相关词表, 利用正则表达式的技术进行进一步的识别。最后综合中文分词的结果与触发词的结果, 一起作为时间表达式识别的结果。

2.2 基于规则的时间表达式识别

通过对时间表达式的观察与分析, 整理出 4 类时间表达式的表示模式, 并根据不同的表示模式设计出相应的正则表达式进行识别。如果文本中有一段字符串能匹配上这 4 种正则表达式的任何一种, 则认为该字符串为时间表达式。

1) 简单(标准)时间表达式。

这种形式的正则表达式为“(凌晨|半夜|今天|早上|中午|上午|下午|晚上|昨天|明天|前天|大前天|后天|大后天)(?<hour>\d+?)(:|:)(?<min>\d+)分?”。这种表达式的特点是用通用的形式表达小时与分钟的概念。“21:12 分”这种时间表达式是国际通用的小时时间表达式。

2) 偏移时间表达式。

这种形式的正则表达式为“(未来|将来|之前|之后|以后|以前|已经有|第|已经|公元前|每|前|后)?([一二两三四五六七八九十零几元 0-9 0 - 9]+个?((年代|世纪|年|月|日|天|号|点|时|小时|钟头|分钟|分|秒|星期|礼拜)(?![钱局])))+((前|后|之前|之后|以前|以后|初|

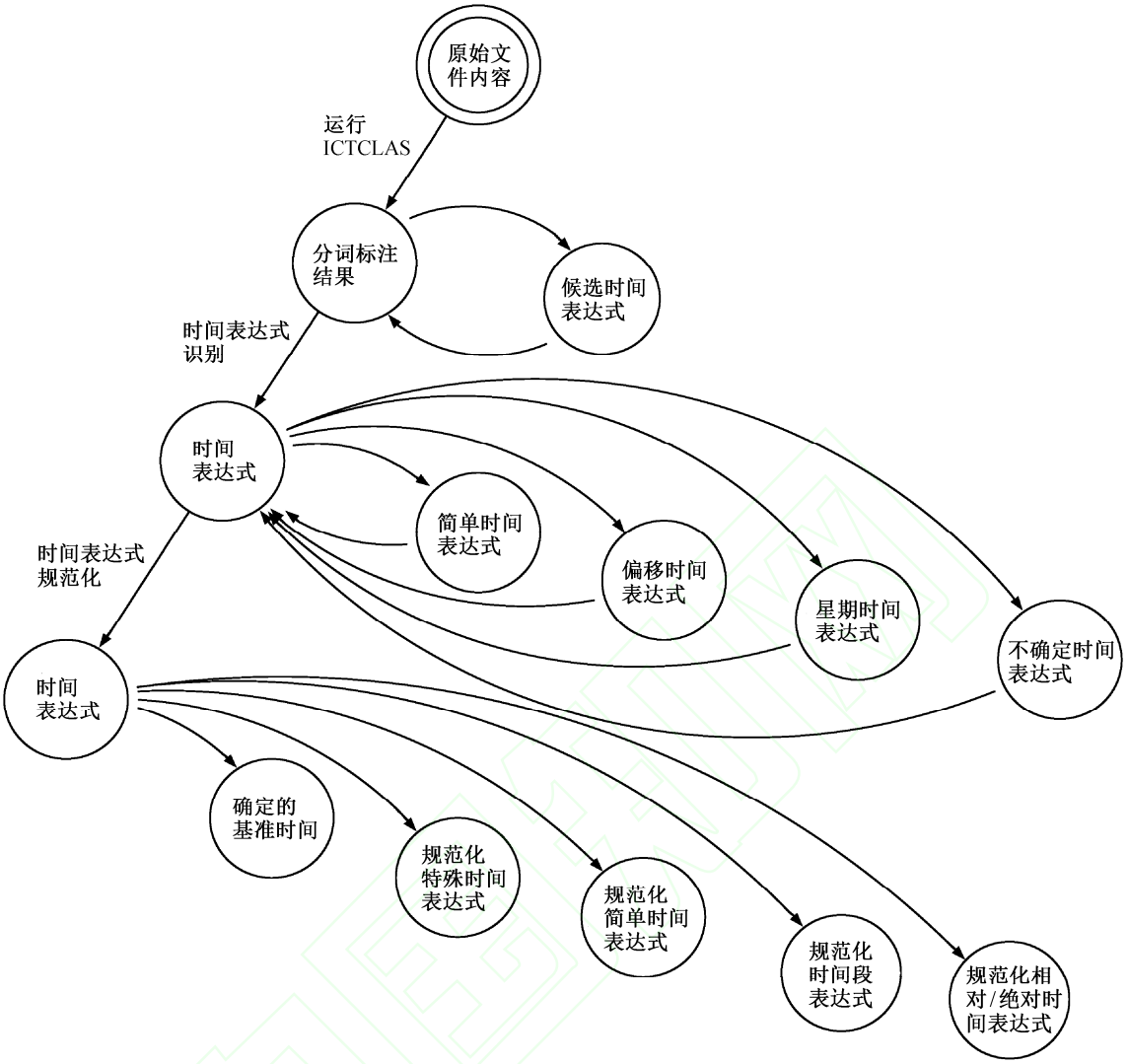

图 1　基于 CFSA 的时间信息处理算法框架

Fig. 1　CFSA based temporal information processing algorithm

末)(?!往))?”。该正则表达式能识别含有具体数字和单位的时间的偏移时间表示形式。

根据对新闻数据集的分析，发现能表明时间单位的词有年代、世纪、年、月、日、天、号、点、时、小时、钟头、分钟、分、秒、星期、礼拜。同时，这些时间单位前必须有具体的数字才能构成完整的表达时间含义的表达式，于是在这些时间单位词前面匹配数量词。

本规则在匹配数量词的时候考虑了中文形式的“一二三”也考虑了阿拉伯数字形式的“123”。这里可能会产生歧义，例如在“工人们在十分努力的干活”这句话中，正则表达式会把“十分”这个词识别为时间表达式。很容易判断这句话中的“十分”说的并不是十分钟这个概念，而是修饰“努力”这个词。为了处理这种歧义，我们在正则表达式识别出时间短语之后，又根据上下文的相关信息加入了错误判断机制。即当识别出的时间表达式只有“十分”这两个字的时候则不是时间表达式，因为很少有只表达“十分钟”这个概念的时候，一般“分钟”都与“小时”同时出现。当确实要表达“十分钟”这个概念的时候也会用“十分钟”而不是只用“十分”这两个字表达。另外还要考虑“分”这个字后面不能跟“钱”“局”这种名字，避免将“一分钱”、“公安三分局”识别为时间词。错误判断机制能够保证识别出的时间表达式的正确性。

3) 表示星期的时间表达式。

这种形式的正则表达式为“(上个|下个|这个)?(周|星期|礼拜)[一二两三四五六天日 1-6](之前|

之后|以前|以后)?”。之所以把星期的表达方式单独拿出来, 而不并入第二种时间表示形式的原因在于, 对于普通的时间表达式, 数量词在时间单位之前(比如“三天”)。但星期的表示方式为数量词在时间单位之后(比如“星期四”)。这两种形式在数量词和时间单位词的顺序上有所区别因此单独分开处理, 使识别精度更高。

4) 不确定性时间表达式。

有一些时间短语并没有具体的数量词和时间单位词, 但他们依然能表达时间信息(比如“元旦”、含有特殊符号的表达式中已经做了介绍。还有一些不确定性的时间短语, 它们表达的并不是一个具体的时间而是一个很大的时间范围(比如“近期”、“初春”)。这些词语都需要识别出来, 但它们并没有一个非常类似的特征能用一个比较短的正则表达式概括。我们用到的方法是总结这些有特殊含义的时间表达式, 然后采用枚举的方法, 看待处理的文本中是否含有相应的特殊时间表达式。

2.3 时间表达式的规范化

识别出时间表达式后, 并不能得到这些词语所表达的具体时间是什么, 因此需要为识别出来的时间表达式赋予一个标准格式, 即规范化表示。只有有了统一的格式, 机器才能处理这些时间信息, 为后续的应用提供支持。

TIMEX2 是时间信息的抽取与规范化的一套详细方案, 标注的对象是时间短语, 其任务在于解释时间短语的含义, 回答某件事情“什么时候发生”“持续了多长时间”或“发生的频率如何”等问题。TIMEX2 规范使用具有 6 种可能属性(VAL, MOD, ANCHOR_VAL, ANCHOR_DIR, SET, COMMENT)的 XML 标记< TIMEX2>标注时间表达式。

本文参考 TIMEX2, 采用基于规则的方法进行时间表达式的规范化。规范化部分的主要功能是识别输入字符串中隐涵的时间信息, 转化为 TIMEX2 的形式进行标注。该模块的输入为时间表达式识别模块的输出, 对于每个字符串单独进行处理, 标注其所蕴含的时间信息。

1) 基准时间的确定。

新闻文本中的时间表达式很多时候都是以相对时间形式表述, 比如“两个月以前”。尽管算法识别出了“两个月以前”这个信息, 但如果没有基准时间, 则不能确定“两个月以前”是从什么时候开始的。在这种情况下, 算法首先利用该新闻的发表时间作为基准时间来计算相对的偏移时间。比如发表时间是“2013 年 5 月 1 日”, 则从 5 月份开始向前推两个月, 得到最终的绝对时间为“2013 年 3 月 1 日”。

有些情况下使用新闻文档的发表时间作为基准时间来计算相对偏移时间会得出错误的结果。比如文档是 5 月 7 日发表的, 但讲述的事件发生在 5 月 6 日, 如果文档中出现了“之前两天”的时间信息, 则应该是从 5 月 6 日开始向前计算两天而不是从 5 月 7 日开始向前计算。因此算法在处理新闻文档时, 基准时间不是固定不变的新闻文档发表时间, 而是会根据推理结果动态调整。

算法在开始处理文档时, 会以文档的发表时间作为基准时间。但如果识别出一个绝对时间, 则以该绝对时间作为基准时间, 直到又识别出另一个绝对时间表达式, 则把基准时间调整为新识别的绝对时间。这种算法可以很好地解决前面例子中事件发生时间与文档创建时间不一致的问题。碰到相对偏移时间表达式的时候, 算法会以最近的前一个绝对时间作为基准时间, 如果没有则用文档时间。假设新闻中首先出现“5 月 6 号”的时间表达式并被算法正确识别及规范化, 则再出现“之前两天”的时候, 会以“5 月 6 号”作为基准时间向前推两天, 得到“5 月 4 号”的最终绝对时间。

2) 时间规范化过程。

第 1 步, 算法调用特殊时间表达式规范化子模块进行识别。该子模块主要处理类似于“元旦”、“国庆节”这种蕴涵特殊时间信息的字符串。

虽然该类型的字符串在没有背景知识的情况下很难从字面获得所蕴涵的时间信息, 但这种有特殊含义的时间表达式的个数是有限的, 因此可以枚举每一个时间表达式和其所表示的时间信息。当遇到同样字符串的时候, 算法可以正确识别其隐涵的时间信息。比如“元旦”代表的就是“1 月 1 日”, “国庆节”代表的是“10 月 1 日”, “情人节”代表的是“2 月 14 日”等。这些特殊时间表达式会记录在一个词表中, 当输入的文本能匹配上词表中的一个表达式时, 则赋予文本该表达式的时间信息。

第 2 步, 算法调用简单时间表达式规范化子模块处理同样的输入字符串。该子模块主要处理标准的时间表达式, 例如“2013 年 5 月 29 日”、 “下午 5:28 分”, 这一类表达式有明显的特征并且带有数量词及时间单位词。

一篇新闻中有时出现“星期几”的时间表达方

式，有时又出现“几月几号”的表达方式，因此需要处理星期到日期的转换。比如“2013 年 5 月 30 日”发表的新闻中出现了“星期二”这个表达式的时候，算法首先会将“2013 年 5 月 30 日”转化为“星期四”，然后比较“星期二”与“星期四”，可以得到事件时间比报道时间早两天，然后再在“2013 年 5 月 30 日”的基础上向前推两天，得到最终“2013 年 5 月 28 日”的时间信息。

第 3 步，算法调用时间段表达式规范化子模块。该模块识别蕴涵时间段信息的时间表达式。例如“20 个小时”、“一年以前”。

时间段表达式与简单时间表达式的特征非常类似，都有数量词与时间单位词。不同的是时间段表达式没有复杂的隐涵信息并且在数量词的约束上没有那么的严格，在一些时间单位的表达上也有一定的区别。比如“天”这个概念，在时间点类型的表达式中会以“日”、“号”这种的形式出现(比如“3 月 3 号”、“4 月 2 日”)，在时间段类型的表达式中会以“天”这种形式出现(比如“用了 3 天”)，没有人会说“用了 3 号”来表示时间段的概念。

第 4 步，算法调用相对时间/绝对时间判断子模块，来判断待处理的字符蕴涵的时间信息是绝对时间信息还是相对偏移时间信息。

相对时间是指偏移时间，一般由一个段时间加上方位词构成。因此我们利用正则表达式识别这种由段时间和方位词构成的字符。正则表达式分为两类：一类识别方位词在时间词后面的字符串模式，正则表达式为“([一二两三四五六七八九十几 0-9 0 - 9]+ 个 *(年|月|日|天|小时|钟头|分钟|星期|礼拜))+(前|后|之前|之后|以前|以后)”。另一类识别方位词在时间词前面的字符串模式，正则表达式为“(未来|将来|之前|之后|以后|以前|已经有|已经|每|前|后).*?([一二三四五六七八九十几 0-9 0 - 9]+个*(年|月|日|小时|钟头|分钟|星期|礼拜))+”。

第 5 步，在得出以上 4 个子模块结果后，我们可以得到最终规范化的结果。如果相对时间与绝对时间模块判断的结果为时间点，则识别出的时间为特殊时间与简单时间综合的结果。如果判断的结果为相对时间，则算法会根据字符串判断时间信息相对于基准时间是向前偏移还是向后偏移。如果是向前偏移，则用当前的基准时间减去第 3 步中识别出的时间段时间作为最终的时间。如果是向后偏移，则用当前的基准时间加上第 3 步中识别出的时间段时间作为最终的时间。如果判断的结果既不是向前偏移也不是向后偏移，则算法认为字符串中所表达的时间并不是一个固定的时间点，而是一个时间段，比如“用了一年的时间”。这个时候我们会把第 3 步中的时间段时间作为最后的时间段，并在 TIMEX2 中的 value 属性值前面标注“P”，表示该属性值所表达的时间为时间段时间。

通过上述 5 步操作，我们既可以规范化绝对时间、点类型的时间表达式，也可以规范化相对偏移类型的时间表达式，还可以规范化时间段类型的时间表达式。

3 实验结果及分析

3.1 数据集

为检测本文时间识别与规范化算法的有效性，我们采用 3 个数据集进行了评测：1)“人民日报标注语料库”1 月新闻文本中的前 200 篇；2) SemEval-2010 中 TempVal 数据集的 33 篇文本；3) 我们手工标注的 2012 年 200 篇“人民武警报”新闻报道。数据集的统计信息如表 1 所示。

本文采用准确率、召回率和 F-measure 评估 C-TERN 算法性能，各项参数的学习均在上述数据集上进行。时间表达式识别与规范化准确率、召回率定义如下，F-measure 定义为 $2PR/(P+R)$。

$$\text{识别准确率RP}=\frac{\text{本算法正确识别出的表达式个数}}{\text{本算法识别出的时间表达式个数}},$$

$$\text{识别召回率RR}=\frac{\text{本算法识别出已标注为时间词的个数}}{\text{数据集中标注为时间词的个数}},$$

$$\text{规范化准确率NP}=\frac{\text{本算法正确赋值的时间表达式个数}}{\text{本算法赋值的时间表达式个数}},$$

$$\text{规范化召回率NR}=\frac{\text{本算法赋值的时间表达式个数}}{\text{可以赋予具体数值的时间表达式个数}}。$$

3.2 标注结果样式

以“人民日报标注语料库”为例，经分词标注后新闻文本形式如下：

19980101–01–001–001/m 迈向/v 充满/v 希望/n 的/u 新/a 世纪/n ——/w 一九九八年/t 新年/t 讲话/n (/w 附/v 图片/n 1/m 张/q)/w

表 1 实验数据集统计

Table 1 Statistics of news data set

数据集	数据来源	时间	数量
RenminDaily	人民日报标注语料库	1998	200
TempEval	SemEval-2010	2010	33
CAPFnews	人民武警报新闻文本	2012	200

19980101–01–001–002/m 中共中央/nt 总书记/n 、/w 国家/n 主席/n 江/nr 泽民/nr

19980101–01–001–003/m (/w 一九九七年/t 十二月/t 三十一日/t)/w

19980101–01–001–004/m 12月/t 31日/t ，/w 中共中央/nt 总书记/n 、/w 国家/n 主席/n 江/nr 泽民/nr 发表/v 1998年/t 新年/t 讲话/n 《/w 迈向/v 充满/v 希望/n 的/u 新/a 世纪/n 》/w 。/w (/w 新华社/nt 记者/n 兰/nr 红光/nr 摄/Vg)/w

……

其中每段的开始部分是该新闻的编号，例如“19980101–01–001–001”表示该篇新闻发表在1998年1月1日第一版的第一篇新闻，并且该行后面的内容是这篇新闻的第一段。我们能够从该编号中提取出新闻发表时间作为时间规范化起点。C-TERN 对时间信息标注规范化后，可得到如下结果:

<STORY_REF_TIME>19980101</STORY_REF_TIME>

迈向充满希望的新世纪——<TIMEX2 val="1998–00–00T00:00:00">一九九八年新年</TIMEX2>讲话(附图片1张)

中共中央总书记、国家主席江泽民

(<TIMEX2 val="1997–12–31T00:00:00">一九九七年十二月三十一日</TIMEX2>)

<TIMEX2 val="1997–12–31T00:00:00">12月31日</TIMEX2>，中共中央总书记、国家主席江泽民发表<TIMEX2 val="1998–00–00T00:00:00">1998年新年</TIMEX2>讲话《迈向充满希望的新世纪》。(新华社记者兰红光摄)

……

示例中被<TIMEX2>及</TIMEX2>标签所包含的字符串为系统识别出来的时间表达式。<TIMEX2>标签中属性“val”的值即为规范化后得到的时间值。

3.3 结果统计与分析

在数据集 RenminDaily 中，对于 200 篇新闻文档，中文分词系统识别出 1244 个时间表达式，手动标注的结果识别出 1301 个时间表达式，而 C-TERN 算法识别出 1455 个时间表达式。我们对于 3 种方法分别统计其所能正确识别的时间表达式的个数，并递增进行比较，结果如图 2 所示。

由图 2 的数据可以看出，“人民日报语料库”手动标注的结果好于只用中文分词处理的结果，而本系统识别出时间表达式的个数又多于“人民日报语料库”手动标注的个数。由于中文分词的方法是面向所有词性进行判断的方法，有些具有时间含义的词

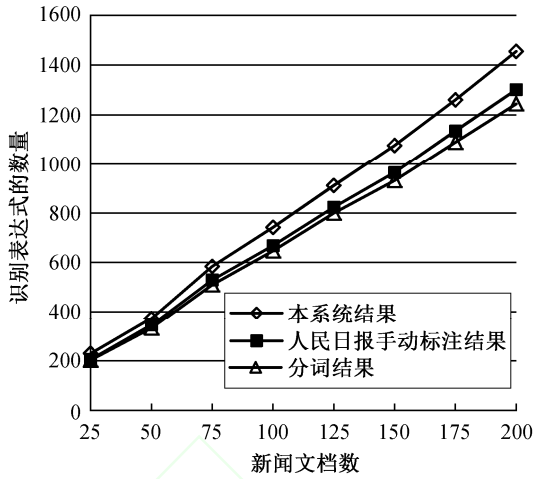

图 2 ReminDaily 时间表达式识别比较

Fig. 2 Comparison of Timxes recognition on RenminDaily

语并没有标注为时间词性。而人民日报手工标注的结果并没有考虑时间段和不确定性的时间，因此对时间做特殊处理的本算法识别出的表达式个数多于其他两种方法识别出的时间表达式个数。

算法识别出的时间表达式个数只能部分反映识别结果的好坏。根据 3.1 节所列测评指标，我们把 200 篇新闻文档分为 4 个部分，分别计算准确率和召回率，具体结果如表 2 所示。

根据结果可以发现，测评指标在不同数据范围相对稳定，平均标注和规范化 *F*-measure 分别为 0.947 和 0.952。文献[10]中的中文 TIMEX2 自动标注系统(CTAT)在《人民日报》1998 年 1 月的语料集上得到的 *F*-measure 分别为 90.2 和 83.3，我们的算法性能有所提高，主要原因是由于规则相对完备一些。

我们进一步利用数据集 TempVal 和 CAPFnews 进行测试，测试结果如表 3 所示。C-TERN 算法在 TempVal 数据集上的表现并不理想，对照人工标注结果，发现 TempVal 将“现”、“那时”、“与此同时”等以及一些专业术语，如“商”、“战国”、“七五”、“八五”、“虎年”、“新世纪”等都标注为时间词，由于 C-TERN 规则集中并没有相应的规则，所以没能识

表 2 RenminDaily 时间信息处理结果

Table 2 RenminDaily temporal information processing result

新闻文档	RP	RR	NP	NR
1~50	0.969	0.896	0.924	0.968
50~100	0.971	0.926	0.942	1
100~150	0.977	0.923	0.933	0.987
150~200	0.984	0.938	0.903	0.963
1~200	0.975	0.920	0.926	0.979

别和规范化。解决这一问题的方法是将相应时间词和规则增补到算法中。

在CAPFnews数据集上, C-TERN算法的识别与规范化 *F*-measure 分别达到 0.948 和 0.939, 与人民日报的结果相当。这说明军事新闻和普通文本中时间信息的表述方式基本相同, 而且军事新闻中的时间信息更规整一些, 因此在通用语言规则基础上增加诸如“战备期间”等少许特殊时间词之后, 就能够满足军事新闻文本时间信息处理的需要。

从上述实验结果来看, 造成 C-TERN 识别出错的主要原因是中文本身存在的歧义。比如算法会从“二十一点九亿美元”中识别出“二十一点”, 从“在北纬二十三点三十的地方”中识别出“二十三点三十”。避免这种错误出现的方法是在标注时利用上下文语义信息与语境信息进行判断。另外, C-TERN 中的规则还需要能嵌套使用, 否则无法完整识别出“星期三(七月九日)上午八时”这个时间表达式。

4 结论

本文的研究目标是对军事新闻文本进行时间表达式的识别与规范化。针对这个目标, 我们对通用和军事新闻文本中的时间表达式进行了分析, 整理了一套规则及相应的词表, 提出了基于层叠有限状态机的中文时间表达式标注、推理与规范化算法C-TERN。

C-TERN 首先利用成熟的分词与词标标注工具ICTCLAS初步识别出文本中的时间词, 然后将时间信息处理规则分成多层, 逐层进行时间表达式的精细识别与规范化。算法注意了规则集提取的正确性、规则之间冲突的消解、以及匹配方式的合理性。在“人民日报语料库”、TempEval 数据集以及CAPFnews 军事新闻数据集上的各项评估效果较好, 能够满足军事新闻文本时间信息处理的需要。

根据对测试中错误结果的分析, 发现算法性能瓶颈主要来源于中文的歧义。因此, 我们的下一步工作是在标注规则中加入上下文语义与语法信息, 提高算法的准确率。

表 3　TempEval 和 CAPFnews 时间信息处理结果

Table 3　Result of temporal information processing on TempEval and CAPFnews

数据集	RP	RR	NP	NR
TempEval	0.795	0.834	0.893	0.916
CAPFnews	0.962	0.935	0.922	0.957

参考文献

[1] TERN-2004. http://timex2.mitre.org/tern.html

[2] Ferro L, Gerber L, Mani I, et al. TIDES 2003 standard for the annotation of temporal expressions. 2004

[3] Gerber L, Huang S, Wang X. TIDES 2003 standard for the annotation of temporal expressions. Chinese supplement draft. 2004

[4] Mani I, Wilson G. Robust Temporal Processing of News // Proceedings of the 38th Annual Meeting on Association for Computational Linguistics. New Brunswick, 2000: 69–76

[5] Ahn D, Adafre S F, de Rijke M. Towards task-based temporal extraction and recognition // Proceedings Dagstuhl Workshop on Annotating, Extracting, and Reasoning about Time and Events. 2005

[6] 贺瑞芳, 秦兵, 刘挺, 等. 基于依存分析和错误驱动的中文时间表达式识别. 中文信息学报, 2007, 21(5):36–40

[7] 刘莉, 何中市, 邢欣来, 等. 基于语义角色的中文时间表达式识别. 计算机应用研究, 2011, 28(7): 2543–2545

[8] 朱莎莎, 刘宗田, 付剑锋, 等. 基于条件随机场的中文时间短语识别. 计算机工程, 2011, 37(15): 164–167

[9] 邬桐, 周雅倩, 黄萱菁, 等. 自动构建时间基元规则库的中文时间表达式识别. 中文信息学报, 2010, 24(4): 3–10

[10] 林静, 曹德芳, 苑春法. 中文信息的 TIMEX2 自动标注. 清华大学学报: 自然科学版, 2008, 48(1): 117–120

[11] Jang S B, Baldwin J, Mani. Automatic TIMEX2 tagging of Korean news. ACM Transactions on Asian Language Information processing, 2004, 3(1): 51–65

[12] Estela S, Martinez-Barco, Patricio, Munoz R. Recognizing and tagging temporal expressions in Spanish // Workshop on Annotation Standards for Temporal Information in Natural Language. LREC. 2002

[13] Li W, Wong K F, Yuan C. A design of temporal event extraction from Chinese financial news. International Journal of Computer Processing of Oriental Languages, 2003, 16(1): 21–39

[14] Wu M, Li W, Lu Q. CTEMP: a Chinese temporal parser for extracting and normalizing temporal information. IJCNLP, 2005: 694–706

[15] ICTCLAS 中文分词系统. http://www.ictclas.org/index.html

[16] 刘群, 张华平, 俞鸿魁, 等. 基于层叠隐马模型的汉语词法分析. 计算机研究与发展, 2004, 41(8): 1422–1429

[17] 徐永东, 徐志明, 王晓龙, 等. 中文文本时间信息获取及语义计算. 哈尔滨工业大学学报, 2007, 39(3): 438–442

[18] 林静, 苑春法. 汉语时间关系抽取与计算. 中文信息学报, 2009, 23(5): 62–67