中文文本蕴含的推理模型*

徐 幸，王厚峰

北京大学 计算语言学教育部重点实验室，北京 100871

E-mail: xuxing@pku.edu.cn

摘 要：文本蕴含问题是指给定文本与假设对，判断文本和假设之间的关系，是证实、证伪还是未知。本文介绍了一个利用词汇知识库(如北京大学的中文概念词典 CCD)、概率计算模型等判断文本与假设之间蕴含关系的推理模型。主要思想是：将句子间推理问题划归到词汇蕴含概率计算，然后利用知识库、网络信息和依存句法分析等工具计算每个词语的词汇蕴含概率，最后通过概率汇总形成整个句子的蕴含概率。实验结果表明，利用词汇蕴含关系实现文本推理是有效的。

关键词：文本推理；词汇蕴含概率；依存句法分析；网络信息；知识库

Inference Models for Textual Entailment in Chinese

Xu Xing, Wang Houfeng

Key Laboratory of Computational linguistics (Peking University), Ministry of Education, Beijing 100871

E-mail: xuxing@pku.edu.cn

Abstract: Textual entailment is a problem to predict whether an entailment holds for a given test-hypothesis pair. This article presents an inference model to solve this problem by means of using lexical knowledge base (e.g. CCD), web information and probability method. The main idea of the model is to simplify the initial problem of sentence inference to computing lexical entailment probability, which can be easily handled by using knowledge base, web information and dependency syntax analysis. Finally, the entailment probability of the whole hypothesis can be combined by all the lexical entailment probabilities in the hypothesis. The experiment indicates that it is effective to recognize textual entailment via lexical relations.

Keywords: textual entailment; lexical entailment probability; dependency syntax analysis; web information; knowledge base

1 前言

1.1 问题定义

文本推理是指：给定文本 T 和假设 H，根据 T 和 H 所表达的意义推断其间是否具有蕴含关系。文本推理任务一般有三类问题和二类问题两种形式，三类问题需要区分证明/Entailment（通过 T 推断 H 为真），证伪/Contradict（通过 T 推断 H 为假），未知/Unkown（通过 T 无法判断 H 的真假），二类问题则只区分证明/ Yes 和不能证明/ No。

例 1 Entailment

T: 约翰昨天买了一本小说。

H: 约翰买了一本书。

由 T 可以知道 H 是真的，因为小说是书的一种。

例 2 Contradict

T: 四月份原油售价在每桶 37.8 美元，下降了 28 美分。

H: 原油价格上升到每桶 37.8 美元。

由 T 可以知道 H 是假的，因为 T 中提到下降，而下降与上升是反义词。

文本推理的应用领域很多，可以用于自动问答系统、智能搜索和多文本摘要等。但是文本推

* 本文受国家自然科学基金资助，基金号：90920011，91024009

理涉及到了语义分析和篇章结构的问题，难度较高，所以目前更多地停留在研究层面。

1.2 相关工作

（1）概率方法

通过计算 H 的概率 P(H)和条件概率 P(H|T)，以判断 T 和 H 是否存在蕴含关系是一种典型的方法[1]。概率的计算主要基于下面的公式(1)：

$$
\begin{aligned}
&P(H=1)=\prod_{H_u\in H}P(H_u=1)\\
&P(H=1\mid T)=\prod_{H_u\in H}P(H_u=1\mid T)\\
&P(H_u=1\mid T)=\frac{P(H_u=1)\prod_{T_v\in T}P(T_v\mid H_u=1)}{\sum_{c\in\{0,1\}}P(H_u=c)\prod_{T_v\in T}P(T_v\mid H_u=c)}\\
&\begin{cases}P(H=1)<P(H=1\mid T) & \text{yes}\\ else & \text{no}\end{cases}
\end{aligned}
\tag{1}
$$

其中条件概率的计算需要借助于搜索引擎搜索关键字返回结果个数[2]，见公式(2)

$$
\begin{cases}
P(T_v\mid H_u=1)=\dfrac{n(T_v \text{ and } H_u)}{n(H_u)}\\
P(T_v\mid H_u=0)=\dfrac{n(T_v \text{ and not } H_u)}{n(not\ H_u)}
\end{cases}
\tag{2}
$$

其中$n(s)$表示在搜索引擎中搜索s返回结果的数量

这种方法存在的问题是，经过搜索引擎返回结果的条数很难准确反映词语之间的关系，也基本没有考虑任何语义成分和其他相关信息，而且搜索引擎本身也很难正确返回实际得到的结果数量。但是将句子蕴含的问题简化为词语蕴含的思想是很有借鉴意义的。

（2）机器学习方法

蕴含关系的判别可以看成是一个分类问题，这样便可以建立机器学习的模型来求解[3]。特征提取主要是通过计算 T 和 H 之间的一些关系得到的。由于目前 RTE 评测给出的训练语料规模有限，所以是不太适合使用机器学习方法。

（3）基于逻辑推理的方法

文本推理既然是一个推理问题，就可以转化为数学上的逻辑推理来实现。逻辑推理方法的基本思路是将 T、H 和可以利用的知识库（比如 wordnet、CCD）中的内容都转化成数学上谓词逻辑的形式，然后将其作为输入给自动推理系统（比如 prolog 语言）进行求解[4]。但是从句子到谓词逻辑的转化是很难保证精确性的，这也会严重影响了最终推理系统的精确性。

（4）使用知识库的方法

这里的知识库主要指具有词汇语义关系的词典，比如 wordnet、中文概念词典(CCD)、hownet 和概念层次网络等。石晶基于知网研究了文本推理问题[5]，贾君枝、邰杨芳的基于汉语框架网络本体的文本推理[6]等等，Iftene[7]则探讨了利用知识库建立完整的文本推理框架。由于知识库主要是人工构建，其词汇语义关系的质量相对较好，通过其得到的蕴含信息也较为准确。

2 主体框架

整个文本推理系统的主题思想是采用划归的理念，将句子间的蕴含问题分解为词语之间的蕴含问题，而词语之间的蕴含问题则可以借助一些知识库（如 CCD、hownet 等等）来判断。

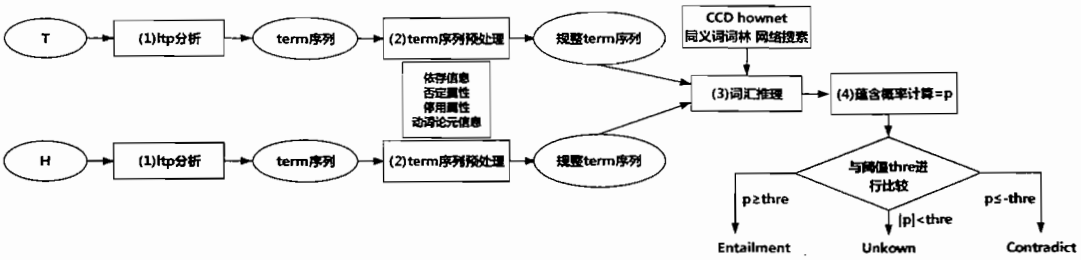

图1　整体框架示意图

2.1　词法与句法分析

本文使用了哈工大开发的语言技术平台 LTP，进行分词、词性标注和句法分析,LTP 分析结果以 Term 序列的形式存储，Term 数据结构存放词语的词法(词语本身、词性、否定属性、停用属性)、句法(依存树上的子节点、父节点、与动词相关的论元)与语义(命名实体、语义角色)信息。

2.2　词语序列预处理

词语序列(Term 序列)预处理是为了得到文本推理是所需的基本特征。预处理包括：

（1）合并词语：合并目的是让分词结果符合粗粒度的原则。命名实体的几个部分需合并为一个整体，如果连续几个词语可以构成一个新词语(以在 CCD 中可以查到为标准)也要合并。

（2）否定词属性：对于一个词语，如果其连接了否定词，那么就认为这个词语的否定词属性 neg=true,否则 neg=false。否定词主要包括“没有”、“不”等词语。

（3）停用词属性：停用词属性有全停用词，半停用词,非停用词三种。全停用词指不参与文本蕴含运算的词，包括标点、否定词和虚词；半停用词只计算正负极性，不计算相似性，主要包括一些特殊动词：“是”、“可以”等等；非停用词需同时计算极性和相似度。

（4）构造论元序列：根据依存句法理论，每一个动词都会与周围的一系列名词发生联系，这些名词称为动词的论元。如果句法分析正确的话，那么动词的论元会被作为动词的子节点。

比如例句 1：***Oracle 泄露了一个机密文件。***

句法分析的结果：

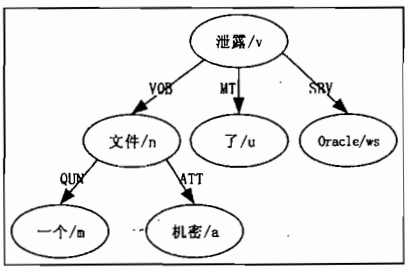

图2　例句 1 依存树

用 arg(*w*)表示词语 *w*对应的论元，那么arg(泄露）= {Oracle,文件}

3　词汇推理

文本推理的最重要思想就是划归，即，将句子间的推理问题转化为词汇间的推理问题，得到假设句子中每一个词语的蕴含概率以后就可以通过一定的方式计算出整个假设蕴含的概率。

首先定义下面的计算经常要出现的两个辅助函数：

符号函数：$\mathrm{sgn}(x)=\begin{cases}1 & x\geqslant 0\\ -1 & x<0\end{cases}$

求绝对值最大的项：$\mathrm{maxabs}(f_1,f_2,...,f_n)=f_i$ 满足对$\forall j\,|f_j|\leqslant|f_i|$

3.1 两个词语间的蕴含关系 $p(T_w\rightarrow H_w)$

对于词汇蕴含概率的计算主要有以下几种方法：

（1）直接对应法：

$$p_{identify}(T_w\rightarrow H_w)=1 \text{ if } T_w=H_w \text{ or } H_w\text{是}T_w\text{的前缀或后缀} \tag{3}$$

（2）基于知识库 CCD 的推理：

我们为每个同义词集定义 Syn 结构，包含同义词、反义词、上位词、下位词、整体词、部件词、属性词以及词语的解释和注释等信息，这些信息都可以从 CCD 中获取。计算 $p_{\mathrm{CCD}}(T_w\rightarrow H_w)$ 需要考虑 T_w 的所有同义词集。

$p_{CCD}(T_w\rightarrow H_w)=\underset{syn\in syn(T_w)}{\mathrm{maxabs}}\,p(syn\rightarrow H_w)$ 这里$syn(T_w)$表示T_w对应的所有同义词集(*Syn*结构)

$$p(syn\rightarrow H_w)=\begin{cases}H_w\in syn\text{的反义词} & -1\\ H_w\in syn\text{的解释}def\text{或注释}note & 0.5\\ H_w\in syn\text{的其他词集} & 1\end{cases} \tag{4}$$

通过 CCD 可以知道反义关系，比如 $p_{\mathrm{CCD}}(\text{下降}\rightarrow\text{上升})=-1$。

（3）使用网络搜索辅助推理 $p_{web}(T_w\rightarrow H_w)$：

知识库可能涵盖的语言知识是有限的，尤其是新词和专有名词。本文利用网络搜索的结果去挖掘更加隐蔽的词语间蕴含关系。本文假定，如果 T_w 和 H_w 存在蕴含关系，那么两个词语就极有可能在网络中共同出现。但是，网络数据量庞大，任何两个词共现的可能性都会比较高，为此本文还引入了一些辅助搜索词 help_word，比如“是”、“意味着”等可以反映蕴含关系的词语，最终生成的搜索关键词 $s=T_w \text{ and } H_w \text{ and } help\_word$

$$p_{web}(T_w\rightarrow H_w)=1 \text{ if搜索}s=\text{“}T_w \text{ and } H_w \text{ and } help\_word\text{”}\text{ 返回结果中确实含有}s \tag{5}$$

比如我们要计算 $p_{web}(\text{家乡}\rightarrow\text{出生})$，在百度中搜索“家乡 出生 是”，发现返回的前几条结果都同时出现这三个词，可以认定 $p_{web}(\text{家乡}\rightarrow\text{出生})=1$。

（4）最终蕴含概率的确定：

将上述各种蕴含关系按如下公式(6)综合：

$$p(T_w\rightarrow H_w)=\mathrm{maxabs}(p_{identify}(T_w\rightarrow H_w),p_{CCD}(T_w\rightarrow H_w),p_{web}(T_w\rightarrow H_w)) \tag{6}$$

3.2 利用句法信息改进 $p(T_w\rightarrow H_w)$ 计算

前面考虑了 T_w 和 H_w 词语之间的关系，完全没有考虑它们在句子的地位以及与其他词语之间的联系。文本推理本身要涉及到对句子意义的理解，有必要将每个词语的句法与相对位置信息引入进来。在 H_w 是动词的情况下，可以考虑 H_w 的论元（在预处理阶段求出）和蕴含词 T_w 之间的关系，以改进 $p(T_w\rightarrow H_w)$ 的计算：

$$p_{\mathrm{arg}}(T_w\rightarrow H_w)=p(T_w\rightarrow H_w)*\frac{|\arg(T_w)\cap\arg(H_w)|+1}{|\arg(H_w)|+1} \tag{7}$$

arg(w)表示词语 w 的论元集合，采用加 1 平滑防止概率为 0，见下面的例子：

例 3 Unkown

T: 新加坡基因组研究院（GIS）的科学家们发现了从一位SARS 患者体内提取出来的冠状病毒的完整基因序列。

H: 新加坡科学家发现，SARS 病毒的基因发生了变化。

利用 CCD 计算 $p_{CCD}(\text{提取} \rightarrow \text{变化}) = 1$，但这两个词语并没有实际蕴含关系。通过考察论元 arg(提取)={病毒}，arg(变化)={基因}，最终概率 $p_{\arg}(\text{提取} \rightarrow \text{变化}) = 1 * \frac{1}{1+1} = 0.5$，可以发现两个词语不存在很强的蕴含关系。

3.3 蕴含概率 $p(H_w)$ 的计算

（1）求词语 $H_w \in H$ 对应的蕴含词 $T_{w-entail}$：

$$T_{w-entail} = \max_{T_w \in T} \text{abs } \arg p(T_w \rightarrow H_w) \tag{8}$$

（2）否定性，文本推理任务和句子相似度计算任务最大的区别就是在于在知道两个词语是相关时，还需要区分它们是正相关的还是负相关的。这就需要考虑两个词语在各自的句子中是否定出现的还是肯定出现的。T_w 和 H_w 两个词语的否定性在预处理阶段可以计算出来，其最终的蕴含概率计算公式如下:

$$p(H_w) = P(T_{w-entail} \rightarrow H_w) * neg(T_{w-entail}) * neg(H_w)$$

其中$neg(t)$表示词语t的否定性，由词语t对应的$term$结构$term(t)$里面的neg属性得到：

$$neg(t) = \begin{cases} 1 & term(t).neg = false \\ -1 & term(t).neg = true \end{cases} \tag{9}$$

例 4 Contradict

T: Oracle 已经尽力避免文件被泄露。

H: Oracle 泄露了一个机密文件。

$p(\text{泄露}_H) = p(\text{泄露}_T \rightarrow \text{泄露}_H) * neg(\text{泄露}_T) * neg(\text{泄露}_H) = 1 * (-1) * 1 = -1$

为了避免混淆，泄露_T表示T中相应词语，泄露_H表示H中相应词语。

4 句子蕴含概率的计算

如果已经求出了所有词语的蕴含概率，就可以计算整个句子被蕴含的概率。本文采用的方法是将正负极性和相关性分开来计算。

对于之前求出来的单个词语的蕴含概率都可以分解成极性和相关性两项：

由$p(H_w) = polarity(H_w) * simlarity(H_w)$可得：

$$\begin{cases} \text{极性} & polarity(H_w) = \text{sgn}(p(H_w)) \\ \text{相关性} & simlarity(H_w) = |p(H_w)| \end{cases} \tag{10}$$

4.1 极性计算

$$polarity(H) = \prod_{H_w \in H} polarity(H_w) \tag{11}$$

4.2 相关性计算

$$simlarity(H) = \sqrt[n]{\prod_{H_w \in H} simlarity(H_w)} \quad n = H\text{中非停用词的个数} \tag{12}$$

4.3 最终蕴含概率的计算

$$entailment(H) = polarity(H) * similarity(H) \tag{13}$$

4.4 最终蕴含类型的判断

在算出最终的蕴含概率，就可以根据给定的阈值 thre(thre>0)决定最终的蕴含类型：

三类问题的形式：

$$result = \begin{cases} entailment(H) \geqslant thre & entailment \text{ 证明} \\ entailment(H) \leqslant -thre & contradict \text{ 证伪} \\ |entailment(H)| < thre & unkown \quad \text{未知} \end{cases} \tag{14}$$

二类问题的形式：

$$result = \begin{cases} entailment(H) \geqslant thre & yes \text{ 证明} \\ entailment(H) < thre & no \text{ 不能证明} \end{cases} \tag{15}$$

针对之前的例 2 得到如下的推理结果：

表 1 词汇蕴含表

词语$\in H$	蕴含词$\in T$	蕴含概率
原油/n	原油/n	I=1.0
价格/n	售价/n	CCD: syn=1.0
上升/v	下降/v	CCD: ant=-1.0

$$polarity(H) = \prod_{H_w \in H} polarity(H_w) = -1$$

$$simlarity(H) = \sqrt[n]{\prod_{H_w \in H} simlarity(H_w)} = 1$$

$$entailment(H) = polarity(H) * similarity(H) = -1 < -thre$$

所以$result = Contradict$

5 实验及结果分析

5.1 实验数据的来源

系统采用的无指导的方法，所以只需测试语料。测试语料是翻译自 RTE1（第一届 RTE 评测）的训练测试语料，共 154 条。二类测试和三类测试都采用同一个语料。

5.2 实验结果之间的比较

定义精度=分类正确的 item 数/总 item 数，进行对比实验：

表 2 不同模块的精度比较

	二类测试	三类测试
基本模块	73.38%	67.53%
基本模块+句法信息	77.27%	70.78%
基本模块+句法信息+网络搜索	75.32%	68.83%

从上面的比较可以看出，引入句法信息（论元信息）对实验效果有一定程度的提升，但是目前考虑的句法信息成分还太少，可以考虑更多的更深层次的句法信息。

加入网络搜索实验效果反而有所下降，一种可能的原因是当前在进行网络搜索时的算法太过简略，而且使用的搜索帮助词也较少，会遗漏一些蕴含关系；另一方面，也可能将无关的词语误认为具有蕴含关系。需要构建更合理的方法法，如，借助像维基百科、百度百科之类的网络资源去挖掘更加准确的词语蕴含关系。

5.3 最终的实验结果

（1）测试标准的定义

$$精度=\frac{正确分类的item数量}{总的item个数}$$

$$就类别A\in\{Entailment,\ Contradict,Unkown\}而言:\begin{cases}准确率P=\dfrac{系统分类结果=标准结果=A的item数量}{系统分类=A的item数量}\\ 召回率R=\dfrac{系统分类结果=标准结果=A的item数量}{标准结果=A的item数量}\\ F值=\dfrac{2PR}{P+R}\end{cases}$$

（2）实验结果

表3 二类测试结果

类别	实际数量	系统数量	正确数量	准确率	召回率	F值
Yes	85	98	74	75.51%	87.06%	80.87%
No	69	56	45	80.36%	65.22%	72%
总数	154	154	119	77.27%		

表4 三类测试结果

类别	实际数量	系统数量	正确数量	准确率	召回率	F值
Entailment	85	98	74	75.51%	87.06%	80.87%
Contradict	19	18	10	55.56%	52.63%	54.05%
Unkown	50	38	25	65.79%	50%	56.82%
总数	154	154	109	70.78%		

最终实验结果还是比较满意的，尤其是二类测试的精度达到了77.27%。

6 总结

本文介绍了解决文本蕴含问题的推理模型，即通过将文本间的蕴含问题化简为词汇间的蕴含问题，然后利用CCD等知识库工具和网络搜索去解决词汇的蕴含问题。

但仍然存在不足，主要包括：（1）极性的计算：只考虑简单的相乘，可能导致很多极性判断错误；（2）没有专门为命名实体、数字等因素建立对应的有效的推理机制。

下一步我们将从如下方面进行改进：（1）句法信息方面：目前考虑的句法因素仅仅是动词的论元，可以考虑引入更多的句法信息；（2）网络搜索方面：需要设计更加完备的搜索算法改进词汇间相似性判定的问题，考虑使用维基百科、百度百科等数据资源去挖掘和发现更多潜在的词汇蕴含关系。此外，测试语料的规模相对较少，我们将构建更大规模的语料。

参 考 文 献

[1] Glickman, Oren, Ido Dagan and Moshe Koppel. A Probabilistic Lexical Approach to Textual Entailment[C]. In Proceedings of the Nineteenth Inter national Joint Conference on Artificial Intelligence. 2005: 1682-1683.

[2] Oren Glickman, Ido Dagan, and Moshe Koppel. Web based probabilistic textual entailment[C]. In Proceedings of the 1st Pascal Challenge Workshop, Southampton, UK. 2005.

[3] Prodromos Malakasiotis.AUEB at TAC 2009[C]. In Preproceedings of the Text Analysis Conference (TAC). National Institute of Standards and Technology Gaithersburg, Maryland USA, 2009.

[4] Akhmatova, Elena. Textual Entailment Resolution via Atomic Proposition[C]. In Proceedings of the PASCAL Challenges Workshop on Recognising Textual Entailment. 2005.

[5] 石晶, 戴国忠. 基于知网的文本推理[J]. 中文信息学报,2006 年 20 卷(1): 76-84.

[6] 贾君枝, 邰杨芳. 基于汉语框架网络本体的文本推理案例研究[J]. 图书情报工作, 2008, 52(7): 75-75.

[7] Adrian Iftene. TEXTUAL ENTAILMENT[D]. Iasi, Romania: "Al. I. Cuza" University, 2009.