

喻识别任务中，我们认为隐喻识别过程也存在有注意力机制。那么，在处理隐喻句的时候，我们是怎么选择句子中的关键信息呢？

隐喻是人类的一种认知手段<sup>[1]</sup>。处理隐喻过程可以认为是认知驱动的。结合第三章的分析，抽象度与认知相关，体现了人类对词语所指称对象的抽象性的评估。因此，词语的抽象度有助于选择注意词，从而构建注意力机制。结合注意力机制的相关认知学研究，我们给出了注意词的定义以及确定依据。

#### 定义 4.1：注意词

注意词体现了在当前任务约束下输入的文本序列中的最为关键的信息。注意词能得到最优先关注，被赋予最高的注意权重。

#### 假设 4.1：

在隐喻识别任务中，文本中具有最高抽象度的词语能传递最为关键的信息，将被优先关注。这个词语即为注意词。

接下来，我们将根据注意力机制的“自下而上”和“自上而下”特点，从两个方面分析假设 4.1 的合理性。

从“自下而上”的形式上看，注意力机制从输入句子入手，根据句子传递的特征，捕捉关键信息。认知学实验指出<sup>[33][34]</sup>，具体词与抽象词具有较显著的认知优势。在一个句子中，相对于抽象词，具体词能被更快地识别。这意味着在处理抽象词时，我们将付出比处理具体词时更多的努力。因此，具有最高抽象度的词语传达着最显著的刺激，系统将被刺激引导，优先关注该词。

从“自上而下”的形式上看，注意力机制根据目标需要选择关键信息。根据 3.2 节的分析可知，隐喻的本质是抽象的。“求同存异”的过程需要对目标域和源域的相似性进行节略和抽象，造成了语义上的冲突。隐喻识别的目的之一便是判断句子中是否具有语义冲突现象。基于这个目的，隐喻识别过程需要关注句子中具有最高抽象度的词语。除此之外，在大部分隐喻句中，目标域的抽象性高于源域的抽象性。我们认为：最高抽象度的词语有极大的可能性能指向句子中的目标

域。因此，选择句子中的具有最高抽象度的词语作为注意词是合理的。

## 4.2 考虑抽象度的基于注意力机制的隐喻识别模型

根据隐喻的认知与语言学特性，本文提出了基于抽象度的具有注意力机制的隐喻识别模型。模型包含四个部分。第一个部分是注意词提取部分：根据抽象度，模型自动抽取句子的注意词，作为输入句子的关键信息。第二个部分是句子特征表征部分：我们引入了双向长短期记忆神经网络对句子进行建模，构建句子特征向量，并基于注意词在句子中的位置信息，对句子的特征向量进行加权运算，构建句子的位置加权的特征向量。第三部分是构建多层注意力机制模块，对位置加权的特征向量进行多次注意力权重计算。最后，第四部分输出句子隐喻性的判断结果。考虑抽象度的基于多层注意力机制的理解模型结构如图 4.1 所示。

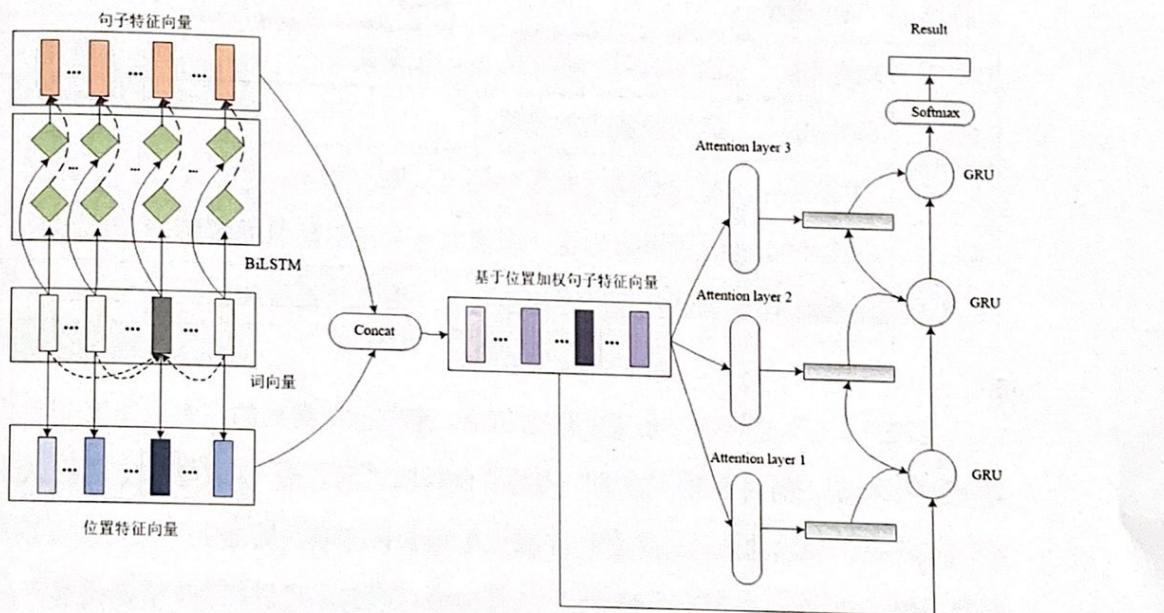


图 4.1：基于抽象度的具有注意力机制的双向长短期记忆神经网络的隐喻识别计算框架。在这里，注意力机制模块的层数为 3。在 BiLSTM 模块中，每一个绿色菱形代表一个 LSTM 单元。在词向量、位置特征向量和基于位置加权句子特征向量中，我们使用不同深浅的颜色区分各维度特征的关键性。

#### 4.2.1 注意词的抽取

系统使用了结巴分词<sup>5</sup>，对汉语句子进行分词处理。由于句子中的一些停用词比如“为了”、“与”、“然后”等，传递较少有用的语义信息。而且这类词的抽象性也难以被衡量。因此，在选择注意词之前，系统预先过滤句子中的停用词。

给定已过滤停用词的句子  $Sentence = \{w_1, w_2, \dots, w_i, \dots, w_n\}$  中，任意一个词语的抽象度数值表示为  $Abs(w_i)$ ，其中  $n$  为句子中不包含停用词的词语总个数。句子中具有最高抽象度的词语将被选为注意词  $w_A$ ：

$$w_A = \arg \max_{1 \leq i \leq n} Abs(w_i) \quad (4-1)$$

表 4.1 展示了部分句子的注意词提取结果。

表 4.1：部分样本的注意词提取结果

样本	句子类型	样本	注意词
1	隐喻句	高原的天，猴儿的脸，说变就变。	高原
2	隐喻句	时间是我垂钓的溪。	时间
3	隐喻句	被绿荫筛过的阳光是玉一般的颜色。	筛过
4	非隐喻句	该国以农为主，国民大多是诚实勤恳的农民。	国民
5	非隐喻句	玛利亚是一名摩登女子，对手绢是喜欢倍至。	喜欢

由表 4.1 可以看得出，在某些隐喻句中，系统抽取得到的注意词能落脚于目标域，如样本 1 的注意词“高原”从属于目标域“高原的天”，样本 2 的注意词“时间”是该隐喻的本体。注意词亦或是落脚于目标域的周围上下文，如样本 3 的注意词“筛过”是目标域“阳光”的修饰语。这满足我们在 4.1 节提出的“注意词有极大的可能性指向目标域”的猜想。在非隐喻句中，系统抽取得到的注意词，放置于整个句子环境中，也能传递关键信息。比如，样本 4 的注意词“国民”指出了主语内容，样本 5 的注意词“喜欢”传递了句子的情感倾向。我们认为根据抽象度提取注意词的方法能使系统找到隐喻识别任务中的关键信息，为后续构

<sup>5</sup> 中文分词工具: <https://github.com/fxsjy/jieba>.

建基于注意力机制的隐喻识别模型奠定了基础。

#### 4.2.2 基于双向长短期记忆神经网络的句子建模

本文使用 BiLSTM 获取句子的特征向量。标准的长短期记忆神经网络 (LSTM) 解决了循环神经网络 (RNN) 存在的梯度弥散问题<sup>[79]</sup>。一个 LSTM 单元包含三个门输入门  $i$ 、遗忘门  $f$  和输出门  $o$  和一个记忆细胞  $c_t$ ，如图 4.2 所示。每个 LSTM 单元的隐层状态取决于前一时刻的状态及当前时刻的数据流。通过门控的方式，LSTM 可以控制输入流数据的输入、输出和记忆，从而实现较长距离的特征表征。

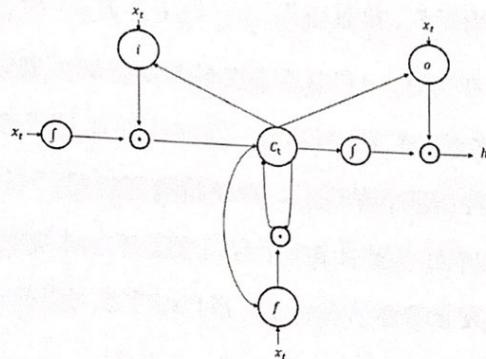


图 4.2：一个 LSTM 单元的结构示意图。 $x_t$  表示当前时间步  $t$  的输入数据， $h_t$  表示隐层状态。

然而，标准的 LSTM 网络只能编码从前到后时序的信息。某一个时段的特征局限于过去时段的隐层特征，而忽视了未来时段的影响。考虑到本文处理的句子具有复杂的语法结构，我们使用 BiLSTM 网络实现前向和后向的句子特征向量建模。

经过注意词抽取步骤，输入的句子可以形式化地表示为： $Sen = \{w_1, w_2, \dots, w_t, \dots, w_A, \dots, w_n\}$ ， $n$  为除去停用词后句子总的词数， $w_A$  为系统自动抽取的注意词。句子序列对应的词向量序列为  $V(Sen) = \{v_1, v_2, \dots, v_t, \dots, v_A, \dots, v_n\}$ 。在前向 LSTM 模型中，任意时间步  $t$  对应的输入词语  $w_t$ ，经由下述计算过程，得

到该时间步下的隐层状态 $h_t$ 和当前特征 $c_t$ :

$$x_t = [h_{t-1}, v_t] \quad (4-2)$$

$$i = \sigma(W_i \cdot x_t + b_i) \quad (4-3)$$

$$f = \sigma(W_f \cdot x_t + b_f) \quad (4-4)$$

$$o = \sigma(W_o \cdot x_t + b_o) \quad (4-5)$$

$$g = \tanh(W_g \cdot x_t + b_g) \quad (4-6)$$

$$c_t = f_t \odot c_{t-1} + i \odot g \quad (4-7)$$

$$h_t = o \odot \tanh(c_t) \quad (4-8)$$

其中， $\sigma$ 为 sigmoid 激活函数， $\tanh$ 为双曲正切函数，也是一个非线性激活函数。算子 $\odot$ 表示向量的内积运算。权重矩阵为 $W_i, W_f, W_o, W_g \in R^{d \times (d+d_w)}$ ，偏置矩阵为 $b_i, b_f, b_o, b_g \in R^{d \times d}$ ，在这里， $d$ 和 $d_w$ 分别为隐层状态和词向量的维度大小。输入门状态、遗忘门状态和输出门状态均为二元控制阀门，用于判断是否对当前输入 $x_t$ 进行处理。基于前向 LSTM 的表征，系统将得到时间步 $t$ 的前向特征 $h_t$ 。

同理，后向 LSTM 可以根据未来信息，表征任意时间步 $t$ 的后向特征 $h'_t$ 。最终，时间步 $t$ 的特征向量表示为 $H_t = [h_t, h'_t] \in R^{d+d}$ 。经过 BiLSTM 的表征，句子的特征向量表示为 $H = \{H_1, H_2, \dots, H_t, \dots, H_A, \dots, H_n\}$ 。

在本文中，我们认为注意词 $w_A$ 会对每一个时间步的特征 $H_t$ 产生影响。 $H_t$ 将根据 $w_t$ 与 $w_A$ 在句子中的相对距离而被调整。与远离注意词 $w_A$ 的词语相比，在 $w_A$ 附近的词语能得到优先的关注。某词语与注意词的距离越近，该词分配的权重越大；与注意词的距离越远，词语的权重越小，这符合人类的阅读和处理文本的习惯<sup>[80]</sup>。基于 $w_A$ 在句子中的绝对位置 $p(w_A)$ ，我们构造了任意一个 $w_t$ 的相对于 $p(w_A)$ 的位置特征 $P_t$ :

$$P_t = 1 - \frac{|p(w_t) - p(w_A)|}{length} \quad (4-9)$$

其中， $p(w_t)$ 是 $w_t$ 在句子中的绝对位置， $length$ 是句子的总长度，表示了句子中所有字符的个数。因此， $length$ 与 $n$ 是不相等的。 $\frac{|p(w_t) - p(w_A)|}{length}$ 表示 $w_t$ 与 $w_A$ 在句中的相对距离。基于位置特征 $P = \{P_1, P_2, \dots, P_t, \dots, P_A, \dots, P_n\}$ ，我们对原始的特征向量 $H$ 进行调整得到基于位置加权的句子向量 $H^* = \{H_1^*, H_2^*, \dots, H_t^*, \dots, H_A^*, \dots, H_n^*\}$ :

$$H_t^* = (P_t \cdot H_t, 1 - P_t) \quad (4-10)$$

基于位置加权的句子向量  $H^*$  强调了注意词的附近语义对句子特征的影响。然而，这并不意味着远离注意词的词语是无用的。尤其是在结构较为复杂的长句子中，若只考虑注意词的近距离文本，容易使得远距离文本的信息缺失，从而使最终的结果不准确。为了解决这个问题。我们构建了多层注意力机制模块。

#### 4.2.3 多层注意力机制

一般情况下，我们使用的句子文本不仅是简单的“X 是 Y”的述谓形式，而是由多个短句子或者从句组合而成。这些子句往往存在着递进复用，转折等特征。根据 LeCun 等人<sup>[81]</sup>的理论，仅仅依靠单层注意力机制，容易使系统表征的上下文特征有所缺失。因此，注意力机制需要多次对句子特征进行抽象化表征和权值计算，从而更好地捕捉远距离文本的关键信息。

多层注意力机制已经被证明在长距离序列中取得了很好的表征能力<sup>[82]</sup>。如何有效组合各层注意力机制的输出是一个关键问题。线性的组合方式容易导致低层次的某些不合理的结果在后续的层次中被放大突出，导致了最终的特征向量包含较多噪音。因此，我们在本文中采取了非线性组合的方式对各层注意力机制的输出做联合处理。

出于门控循环单元 (Gated Recurrent Unit, GRU) 具有较少训练参数和高效的计算性的特点，我们引入 GRU 实现多层注意机制的非线性组合。一个 GRU 单元结构包含两个门：r 为重置门控，z 为更新门控。GRU 示意图如图 4.3 所示：

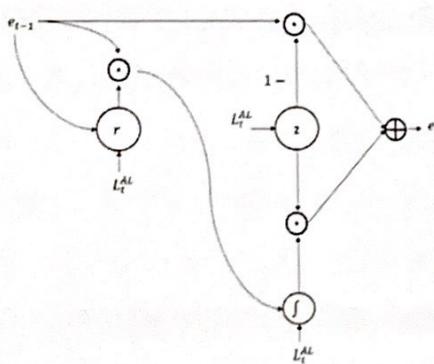


图 4.3：一个 GRU 单元的结构示意图， $L_t^{AL}$  表示输入数据， $e_t$  表示隐层状态。

我们假设在当前时间步t下，任意一层注意力机制AL的输出为 $L_t^{AL}$ ，其计算过程为：

$$L_t^{AL} = \sum_{k=1}^n \alpha_k^t \cdot H_k^* \quad (4-11)$$

$$\alpha_k^t = \frac{\exp(g_k^t)}{\sum_{j=1}^n \exp(g_j^t)} \quad (4-12)$$

$$g_k^t = W_L(H_k^*, e_{t-1} w_A) + b_L \quad (4-13)$$

其中， $W_L$ 和 $b_L$ 分别是当前注意力机制层的权重矩阵和偏置矩阵。

GRU 可以更新任意一层注意力层的时序状态 $e^{AL}$ 。为了后续论述公式方便，我们将 $e^{AL}$ 简化成符号 $e$ 。 $e_{t-1}$ 表示时间步t以前的状态信息。那么，在某AL层 GRU 胞元中， $e_t$ 的更新过程如下所示：

$$M = [e_{t-1}, L_t^{AL}] \quad (4-14)$$

$$r = \sigma(W_r \cdot M) \quad (4-15)$$

$$z = \sigma(W_z \cdot M) \quad (4-16)$$

$$e'_t = \tanh(W_x \cdot L_t^{AL} + W_e(r \odot e_{t-1})) \quad (4-17)$$

$$e_t = (1 - z) \odot e_{t-1} + z \odot e'_t \quad (4-18)$$

其中， $W_r, W_z \in R^{d_g \times (d+d)}$ 分别为重置门控和更新门控的权重。 $W_x \in R^{d_g \times d_g}, W_e \in R^{d_g \times (d+d+1)}$ 是 GRU 胞元的权重。在这里， $d_g$ 是 GRU 隐层状态的维度大小。在任意一层注意力机制AL中，GRU 对当前的输入序列进行了特征的抽象化计算，得到一个新的在 $w_A$ 约束下的特征向量，作为下一个注意力机制层的输入。经过多次注意力机制层的抽象更迭计算，远离 $w_A$ 的关键词语将被赋予合理的注意力权重。最终，模型将得到一个能同时表征短距离-长距离语义的特征向量 $E$ ，作为二元判别模块的输入。

#### 4.2.4 二元分类

识别模型引入了Softmax函数，实现了句子隐喻性的二元分类。Softmax函数可以将特征向量 $E$ 转化成条件概率分布：

$$s = W_s \cdot E + b_s \quad (4-19)$$

$$P(y = \text{label}|E) = \frac{e^{s_i}}{\sum_{i=1}^2 e^{s_i}} \quad (4-20)$$

$$y^* = \arg \max(P(y = 1|E), P(y = 0|E)) \quad (4-21)$$

其中， $W_s$ 和 $b_s$ 分别代表了Softmax函数的权重矩阵和偏置矩阵。在我们的识别模型中，标签 $\text{label}$ 一共有两个类别，标签“1”代表隐喻性，“0”代表非隐喻性。

在训练过程中，我们使用了 $L_2$ 范数降低训练集的交叉熵。同时，模型引入 drop-out 机制<sup>[83]</sup>防止过拟合，引入 Adam 优化器<sup>[84]</sup>实现训练过程中的随机优化。

### 4.3 实验设置

在本小节，我们将介绍实验的训练集和测试集的收集，分析数据集的类型，指明实验使用的词向量模型，最后给出实验的参数设置。

#### 4.3.1 数据集整理与分析

本文从《汉语比喻词典》<sup>6</sup>和文学杂志中人工收集隐喻句子，从文学杂志中人工收集非隐喻句子。我们邀请了三位志愿者参与人工收集过程，并邀请另外一名志愿者剔除数据集中隐喻性不清的句子。这四名志愿者均以汉语为母语，并具有一定的隐喻学相关知识。最终，我们共收集 2591 条汉语隐喻句子，2481 条非隐喻句子。数据集中句子来自不同体裁的文学作品，包括散文、应用文、小说等。隐喻句的类型包括名词性隐喻、动词性隐喻、形容词隐喻等，也包括多类型隐喻夹杂的样本。在本文中，我们认为仅仅由一个“主谓宾”结构构成的句子为简单句子，由多个简单句子组合构成的句子为复杂句子。根据统计分析，大约 55% 的数据样本为简单句子，大约 45% 的数据样本为复杂句子。其中，最长样本包含 138 个字符。

部分隐喻句子包含“像”、“好像”、“如”、“好似”等指向为明喻性用法的词语<sup>[21]</sup>，为了防止这些词语对识别结果产生影响，我们将这一部分词语更换为“是”或者剔除这一类的词语。数据集按照如表 4.2 所示的比例随机分为训练集和测试集。在每一次训练过程中，模型随机抽取训练集的 10% 的样本作为验证集。

<sup>6</sup> 字典资源可以在线获得：<http://gongjushu.Oversea.cnki.net/oversea/R200605063.html>.

**表 4.2: 数据集在训练集和测试集的分布情况**

数据集	隐喻性句子	非隐喻性句子	句子总数目
训练集	2091	1981	4072
测试集	500	500	1000
总数	2591	2481	5072

### 4.3.2 词向量模型

在本次实验中，我们一共引入了三种词向量模型。三种词向量的训练机制均为负采样的 Skip-gram 方法，词向量维度均为 300 维。第一种词向量模型和第三章计算汉语词语抽象度方法使用的词向量模型一致，训练语料来自于百度百科。第二种词向量模型的训练语料来自于文学作品集合，规模小于百度百科模型。前两种词向量模型均可以从网上开源库中获得<sup>7</sup>。第三种词向量模型的训练语料来自于读者语料库<sup>8</sup>，规模是三个之中最小的。我们使用第一种词向量作为实验的基准词向量模型。另外两个词向量模型用于分析词向量模型规模对实验结果的影响，具体的实验分析见 4.4.3 小节。对于未登录词，我们取一个随机 300 维向量作为该词的向量表示。

### 4.3.3 实验的参数设置

本文在 Tensorflow 框架下搭建基于抽象度的具有多层注意力机制的 BiLSTM 深度学习模型。模型的输入词向量维度为 300 维，隐层的维度设置为 300 维。 $L_2$  范数的惩罚系数设置为 0.001，dropout 机制的概率设置为 0.5，学习率设置为 0.005。我们将批处理大小设置为 50。

## 4.4 实验及实验分析

本小节将展示基于抽象度的具有多层注意力机制的隐喻识别计算的实验与

<sup>7</sup> 百度百科及文学作品词向量模型：<https://github.com/Embedding/Chinese-Word-Vectors>.

<sup>8</sup> 读者语料库：<http://www.duzhe.com/#/>.

分析。实验一共分为三个部分：一是与其它神经网络模型进行性能对比，二是分析注意力机制模块的效应，三是分析词向量规模对结果鲁棒性的影响。最后，我们针对测试样例进行分析。

由于本文处理的是二分类问题，我们使用正确率（Accuracy）和 F1 值作为实验的评价指标。识别模型在测试数据集上的预测或正确或不正确，导致了四种不同的分类情况：TP——将隐喻样本预测为隐喻；FN——将隐喻样本预测为非隐喻；FP——将非隐喻样本预测为隐喻；TN——将非隐喻样本预测为非隐喻。

那么，正确率的计算方式为：

$$\text{Accuracy} = \frac{TP + TN}{TP + FN + FP + TN} \quad (4-22)$$

F1 分数是精确率 P 和召回率 R 的调和均值，计算方式为：

$$P = \frac{TP}{TP + FP} \quad (4-23)$$

$$R = \frac{TP}{TP + FN} \quad (4-24)$$

$$F1 = \frac{2TP}{2TP + FP + FN} \quad (4-25)$$

#### 4.4.1 对照实验

目前，大部分以二元分类为目的的基于神经网络的识别系统<sup>[44][45][46]</sup>着眼于短语隐喻识别。这些系统难以迁移到句子级别的隐喻识别。部分能处理句子隐喻的系统<sup>[42][47]</sup>关注英语动词隐喻的识别，需要预先对句子中待识别的动词进行人工标注，从而判断该动词在当前语句中是否具有隐喻性用法。我们考虑到本文不需要预先对句子中的源域或目标域进行人工标注，而且处理对象是汉语句子隐喻，不局限某一特定句法隐喻。为了验证本文识别模型的有效性，我们将模型与其它通用的神经网络分类模型作对比。

(1) 卷积神经网络模型 CNN：本文使用的 CNN 模型是由 Kim<sup>[85]</sup>提出的。该分类模型的卷积层采取了“窄”卷积的方式，pooling 层采用了取特征最大值的方式。该模型并未引入注意力机制。

(2) 基于标准的长短期记忆神经网络模型 LSTM：该模型使用标准的 LSTM 网络实现隐喻识别，使用最后一个隐层的输出作为分类特征的依据。

(3) 基于标准的双向长短期记忆神经网络模型 BiLSTM：该模型的参数与本文中用于表征句子初始特征向量的 BiLSTM 网络是一致的。该模型不考虑抽象度知识，不考虑注意力机制，分类取决于最后一个隐层的表征。

(4) FastText：FastText 是一个开源的词向量构建和文本分类模型<sup>[86]</sup>。该模型包含输入层、投影层及 Softmax 输出层，不包括注意力机制层。它通过叠加词向量的方式获取文本向量，借助多层 Softmax 函数实现分类。特别地，本文训练得到的基于 FastText 的隐喻分类模型使用的是基于 FastText 的预训练好的汉语词向量模型<sup>9</sup>。

对于上述基线方法和本文方法，我们均使用各个方法中具有最小训练误差的分类模型来验证测试集的结果。除了 FastText 基线实验，每个方法使用相同规模的百度百科词向量模型。同时，我们将本文方法的多层次注意力机制的层数设置为 3。对照实验结果如表 4.3 所示。

表 4.3：和其它神经网络的对照实验及实验结果

模型	正确率	F1 值
CNN	0.851	0.852
LSTM	0.886	0.885
BiLSTM	0.905	0.899
FastText	0.906	0.906
本文方法	0.970	0.971

从实验结果可以看出，本文提出的方法表现最优，验证了注意力机制在抽取句子关键信息、提升系统识别性能发挥着较好的作用。尤其是处理“具体隐喻具体”这一类较为棘手的隐喻的时候，我们的方法能根据抽象度从看似均为具体词语的句子中抽取较为关键的信息，从而更好发掘句子中的语义冲突。我们注意到 BiLSTM 方法的效果比 LSTM 方法和 CNN 方法要好。通过双向表征的方式，BiLSTM 能提取句子文本中更丰富的上下文信息。相比之下，CNN 方法在处理

<sup>9</sup> 基于 FastText 的词向量模型获取：<https://fasttext.cc/docs/en/english-vectors.html>.

短距离文本更具有优势，而在处理长文本时容易丢失长距离的信息，在识别复杂隐喻时具有局限性。LSTM 方法只考虑从前向对文本进行表征，忽视了未来信息对当前数据流的影响，导致上下文信息有所缺失。这个实验结果同时从侧面印证了使用 BiLSTM 构建句子的初始特征向量的合理性。FastText 分类模型的性能优于 BiLSTM，原因在于基于 FastText 的词向量模型是面向分类任务而训练得到的。但 FastText 与本文方法在结果上有着约 7% 差距。这体现了注意力机制在识别中的关键作用。

#### 4.4.2 注意力机制的评估实验

在本小节中，我们将从三个方面评估注意力机制对识别结果的影响：一是，对比不同注意力层数下的识别性能；二是，探讨基于相对位置的注意力分配机制对识别结果的影响；三是，分析不同选择方案得到的注意词对结果的影响。

在评估注意力层数的实验中，我们分别设置注意力机制的层数为 1、2、3、4 和 5，保持模型其它参数不变，探讨不同层数下识别结果的变化。表 4.4 展示了在不同注意力机制层数下，系统具有的识别性能差异。

**表 4.4：不同注意力机制层数下系统识别性能的对比**

注意力机制的层数	正确率	F1 值
1	0.934	0.937
2	0.955	0.952
3	0.970	0.971
4	0.964	0.962
5	0.961	0.961

根据表 4.4，我们发现当多层注意力机制的层数取 3 的时候，系统识别性能最佳。若只采用 1 层注意力机制层，部分远离注意词的关键信息可能无法被有效提取。然而，识别性能不与层数呈正比关系。我们注意到当注意力层数取 4 和 5 时，系统的识别性能均低于当注意力层数取 3 的时候。特别地，层数为 4 的模型

的识别性能优于层数为 5 的模型。我们认为：太多次的注意力计算将导致最后一层得到的隐层特征被过度地抽象，原句子的部分特征被高度压缩，最后识别结果出现了偏差。除此之外，过多的注意力层带来了高负担的计算消耗。因此，注意力机制层数的取值不是“多多益善”，而是满足“奥卡姆剃刀”原则。

多层注意力机制层有利于系统捕捉长距离文本的关键信息，而对于注意词本身文本附近的关键信息，我们基于各个词语与注意词的相对距离，构建了一个位置特征向量，用于强调注意词最近上下文的关键性。为了分析位置特征向量对实验结果的影响，我们对注意力的初始赋权方式（公式 4-9、公式 4-10）进行了改动，设置如下两个对比实验：

NoLocal：去掉基于位置的赋权方法。不考虑句子的位置特征向量，将基于 BiLSTM 表征得到的句子特征向量  $H = \{H_1, H_2, \dots, H_t, \dots, H_A, \dots, H_n\}$  直接作为多层次注意力模块的输入。

RanWeight：随机赋权方法。将各个词语对应的初始权重参数赋予一个随机数，生成随机加权向量。模型在原有句子特征向量基础上构建一个随机的加权向量，作为多层次注意力机制模块的输入。

除了位置特征向量模块的差异，上述两个对照实验和本文方法的模型参数是一致的。基于本小节的分析可知，当注意力层数为 3 的时候，识别性能最佳。因此，这两组对照实验的注意力层数均设置为 3。对比实验结果的如表 4.5 所示：

表 4.5：在不同赋值方式下的句子加权特征向量对识别性能的影响

赋值方式	正确率	F1 值
NoLocal	0.934	0.931
RanWeight	0.922	0.918
本文方法	0.970	0.971

表 4.5 的对比实验结果明确了基于位置信息的加权特征向量对隐喻识别的效果有所提升。相对于原始的句子特征向量，加权向量提高了识别模型约 3.6% 的正确率和约 4% 的 F1 值。多层次注意力机制已经被证实能有效抽取句中具有长距

离依赖特性的关键信息，但是由于多层的抽象化计算，导致注意词的附近上下文信息被过度压缩。在进行多层注意力机制的运算之前，我们利用词语的位置信息，对原有句子的特征向量进行了初始的权重分配，从而使得在进入多层注意力机制运算之前的句子特征向量能蕴含更多的信息，注意词的附近关键特征能在多层抽象化计算中被保留。而随机的权重分配方式带来了最差的实验性能。这个现象说明了合理构建句子的加权特征向量的必要性。若初始的加权特征向量无法较准确地体现句子的关键信息，会扰动后续的多次注意力的抽象化计算，带来较大的误差。

在本文中，我们取句子中具有最高抽象度的词语作为识别过程优先关注的关键信息。为了验证最高抽象度词语是注意词能指导系统获取更多关键语义这一假设（假设 4.1），我们提出了如下两个对照实验：

**Random:** 识别模型随机从句子中抽取一个词作为注意词。其中，句子中的停用词已被剔除。

**Low-abs:** 模型将优先关注句子中最具体的词语，选取句子中具有最低抽象度的词语作为注意词。在选取之前，句子中的停用词已被剔除。

这两组对照实验的模型参数和本文的模型一致。所有模型的注意层数均为 3。基于不同的注意词获取机制，系统的性能对比结果如表 4.6 所示。

**表 4.6：基于不同的注意词获取机制的识别性能对比**

注意词的选取机制	正确率	F1 值
Random	0.913	0.914
Low-abs	0.943	0.940
本文方法	0.970	0.971

从表 4.6 可知，选取具有最高抽象度的词语能使识别模型更好地关注句子中的关键信息。考虑最低抽象度的词语作为注意词的方法取得了较差的实验结果。这是因为具有最低抽象度的词语往往与源域相关或集中在源域的附近，传递的信息以非隐喻性意义为主。实验结果验证了我们的猜想：在隐喻识别任务中，句子

中具有最高抽象度的词语能传递更多的冲突语义信息。而随机抽取注意词的方式具有最差的性能。这种方式其实是系统在无抽象度指导下，通过训练的方式学习样本隐喻性的过程。由此可知，抽象度在构建注意力机制具有指导性作用。同时，根据表 4.3，我们注意到该方式取得的实验性能优于基于 BiLSTM 和基于 FastText 两组基线实验的性能。这个结果体现了在隐喻识别中引入注意力机制的必要性，并证明了句子中的各个词语传递的信息对隐喻识别任务的重要性是不一致的。

#### 4.4.3 词向量规模的评估实验

在本文中，词向量模型一共有两个用途：一是，基于词向量模型计算词语与抽象词集和具体词集的语义关联；二是，从词向量模型中取各个词语的向量作为神经网络的输入。由于各个词语的词向量并没有随着训练过程进行调整，我们在本小节只探讨识别系统对词向量规模是否具有鲁棒性。

对照实验使用的三种词向量模型已经在 4.3.2 小节做了介绍。其中，百度百科词向量模型具有 745M 的词条，文学作品词向量模型具有 177M 的词条，而规模最小的读者词向量模型只有 40M 的词条。图 4.4 直观地展示了在三种词向量模型下，系统的性能对比结果。其中，除了使用词向量不同，其它的实验参数一致，且注意力机制层数设置为 3。

根据图 4.4 可知，三种词向量模型均取得了较好的识别结果，证明了我们的系统对不同规模的词向量模型具有一定的鲁棒性。我们注意到虽然文学作品词向量的规模小于百度百科词向量的规模，但二者取得了不分伯仲的实验结果。隐喻是一种诗性的语言，具有丰富的文学性和文化色彩<sup>[21]</sup>。文学作品词向量模型承载了更多的文学性语义，与隐喻内在的文学性特征一脉相承，因此它能在隐喻识别任务中表现出色。读者词向量模型的实验性能最差。过少词条数目导致了过多的非登录词被赋予了随机向量，带来了噪音，尤其是在计算抽象度的部分，词语关联关系的计算出现了较大误差，使得部分词语的抽象度计算不准确。

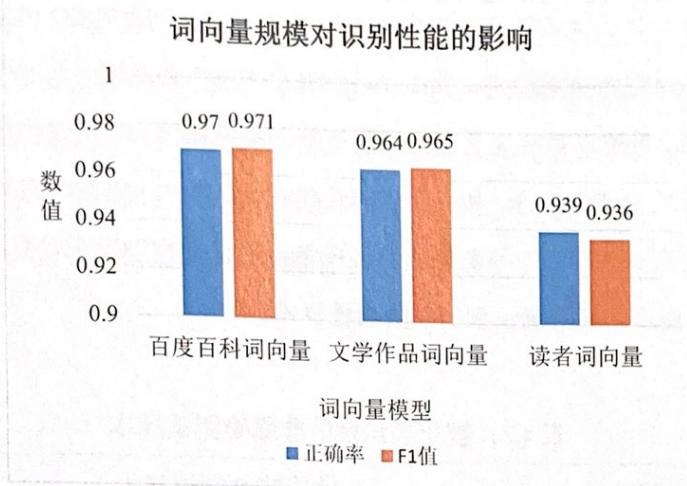


图 4.4: 基于三种词向量模型(百度百科词向量、文学作品词向量和读者词向量),识别系统的结果比较。

#### 4.4.4 实例分析

我们将在本小节对测试样本进行实例分析。部分测试样本的识别结果在附录B给出。表 4.7 展示了被正确识别的隐喻性样本。

表 4.7: 被正确识别的隐喻性测试样本

样本	注意词	隐喻性测试样本
M1	思念	思念是平静的湖水，明澈清透的，荡起阵阵涟漪。
M2	黯淡	没有一片云彩，也没有一丝笑影，带着灰色、空蒙、黯淡的气息，那天空和干涸的荒漠不相上下。
M3	思想	一着急，思想的线要打结了。

表 4.7 证实了具有高抽象度的词语能显著地体现所属句子的语义冲突的这一猜想。隐喻句 M1 和 M3 是典型的“具体隐喻抽象”类型。注意词“思念”和“思想”均落脚于目标域中，它们的抽象度均高于对应的源域“湖水”和“线”，并显著高于其所在句子上下文的其它词，造就了句子整体的抽象度出现了明显的“起伏”变化。这个现象意味着句子存在着语义冲突的可能。由此可知，最抽象

的词语比最具体的词语能指示更多冲突语义的可能。隐喻句 M2 满足“具体隐喻具体”这一个特殊隐喻形式。其中，目标域“天空”和源域“荒漠”的抽象性差异微乎其微。系统选择的注意词虽然没有落脚在目标域，但是该注意词在句子中扮演着目标域的修饰成分，和目标域的距离较近。借助于多层注意力机制，识别系统根据注意词“黯淡”不断修正特征向量，对“天空”赋予较高的注意权值。

表 4.8 展示了非正确识别的非隐喻性样本。

表 4.8：被正确识别的非隐喻测试样本

样本	注意词	非隐喻性测试样本
L1	看法	传统的看法认为人类大脑相对含量之所以发达是出于谋食求生的需要。
L2	轮胎	早期的汽车轮胎要经常修补和调换，比加油还要频繁。
L3	喜欢	每天早晨六点多钟，孩子们会跑进我们的房间，跳到床上，手里晃动着他们最喜欢的书。

根据表 4.8，非隐喻句中的具有最高抽象度的词语往往扮演着句子的主语成分，比如非隐喻样本 L1 和 L2。一般而言，句子的主语部分指的是句子意义的主体，传递着较丰富的信息，对句子的隐喻性与否有着影响。在非隐喻样本 L3 中，注意词“喜欢”作为一个宾语结构，也提供了较丰富的信息，指导系统判断句子的隐喻性。根据上述分析可知，抽象度在隐喻识别过程中扮演着举足轻重的指导性作用：一是，根据句子各个词语的抽象度，勾勒出句子随着时间步变化的抽象性，引导系统找到句子中的异常；二是，抽象度高的词语一般从属于句子的主干部分，传递着更多与句子意义相关的信息。

在一些测试样本中，我们注意到系统得到的注意词“落脚”于源域周围。比如，复杂隐喻长句“快看窗外，树叶铺展着，疏疏落落的，是凤尾一般，别有一种潇洒的风情”中，抽取的注意词是“风情”，和源域“凤尾”的距离较近。这句隐喻样本被错误地识别为非隐喻句。究其原因是基于词语相对距离计算加权的句子特征具有一定的局限性。在上下文环境中距离相近的词只能意味着两个词是

语义相关，但并不能表明二者在语义上具有深层次的相似关系。虽然多层注意机制能对句子的加权特征进行调整，但是由于一开始作为多层注意机制模块输入的特征向量有失偏重。多层注意机制的调整可能难以解决这个“失误”，最后导致了错误的识别结果。因此，若进一步将深层次的语义相似关系引入注意权重的计算调整过程，我们相信会使识别模型达到更好的效果。

## 第五章 基于合作网的汉语句子隐喻理解研究

隐喻意义是目标域和源域互动的结果。所谓的互动，体现在源域构建一个与目标域相似或相关的特征集合，目标域基于此对自身的特征进行选择、强调和压缩。两个概念域通过相互作用，创造了相似关系<sup>[23][24]</sup>。基于互动论，苏畅等人<sup>[35]</sup>提出合作机制，描述目标域和源域的互动关系：源域向目标域展示属性，目标同意选择某一源域属性，以此二者构建了合作关系。

隐喻意义的突显不仅依靠两个概念域的互动，还依赖于特定的语境<sup>[21]</sup>。上下文的概念也参与目标域和源域合作关系的构建中。在本文，我们以合作机制为基础，提出了合作网模型，用以表征隐喻句子中属性与概念间，属性与属性间的多种合作关系，并计算各个合作关系的强度，最后根据合作强度突显隐喻释义。合作关系是一个双向的过程。我们认为，对于合作双方，它们各自与另一方达成合作的意愿是不对等的。为了更好地衡量二者的合作关系，我们提出了基于相关域知识的联想度计算方法，用以突出合作关系的方向性。

本文的隐喻理解模型针对汉语名词隐喻句，输出“目标域是某属性”形式的理解结果。比如，输入隐喻句“友情是一种需要小心积蓄和保存的财富”，模型输出的理解结果是“友情是珍贵的”。

### 5.1 考虑相关域知识的联想度计算

Konikowska<sup>[87]</sup>指出了两个对象的相似关系应取决于某一个特定的域。比如，当考虑到职业域的时候，“老师”和“园丁”两个概念是相似的。由此而启发，我们认为：衡量两个词语的相似关系时，我们需要以某一个特定的域作为参照，比如由词语相关词构成的相关知识域。以这些域知识为基础，我们重新探讨两个词语的相互关系。我们定义了联想度，用以衡量某一个词联想到另一个词及其该词相关语义域的能力。联想度具有方向性。对于相关知识域和联想度，我们给出了如下定义：

### 定义 5.1：相关知识域

在本文中，我们将相关知识域简化为某一词语  $Concept$  在语料中的相关词集合  $Domain(Concept) = \{k_1, k_2, \dots, k_i, \dots, k_D\}$ ，其中， $D$  为词语  $Concept$  的相关词的个数。任意一个相关词  $k_i$  与  $Concept$  是相关的，体现在两个词的上下文是相似的。二者对应的词向量之间具有较高的余弦相似性。

### 定义 5.2：联想度

给定两个不同的词语  $A$  和  $B$ ，及其对应的相关知识域  $Domain(A)$  和  $Domain(B)$ 。对于  $A$ ， $A$  联想至  $B$  的过程需要参照  $Domain(B)$ 。若  $A$  和  $Domain(B)$  具有较高的余弦相似性，且考虑到  $B$  与  $Domain(B)$  本身具有高的余弦相似性，则认为  $A$  联想至  $B$  的可能性是高的。我们将  $A$  联想至  $B$  的联想度表述为  $Rel(A \rightarrow B)$ 。同理， $B$  联想至  $A$  的联想度表述为  $Rel(B \rightarrow A)$

基于定义可知，当计算  $Rel(A \rightarrow B)$  时，我们需要对  $A$  和  $Domain(B)$  的余弦相似性  $Sim[A, Domain(B)]$  以及  $B$  和  $Domain(B)$  的余弦相似性  $Sim[B, Domain(B)]$  作比较。本文采取的计算方式如下所示：

$$Rel(A \rightarrow B) = \frac{Sim[A, Domain(B)]}{Sim[B, Domain(B)]} \quad (5-1)$$

其中， $Sim[X, Domain(Y)]$  衡量的是词语  $X$  对应的词向量  $Vec(X)$  与相关知识域  $Domain(Y)$  对应的域向量  $Vec(Domain(Y))$  之间的余弦相似度。我们认为相关知识域是一个词袋模型。无需考虑各个相关词之间的顺序。根据 Kamp 和 Partee<sup>[88]</sup> 的理论，可知整体意义是各部分意义的函数，我们取所有相关词  $\{k_1, k_2, \dots, k_i, \dots, k_D\}$  的平均词向量作为域向量：

$$Vec(Domain(A)) = \frac{1}{D} \sum_{i=1}^D Vec(k_i) \quad (5-2)$$

此时， $Vec(Domain(A))$  和各个相关词对应的向量  $Vec(k_i)$  具有一致的维度。

由公式(5-2)可知， $Rel(A \rightarrow B)$  的值越大， $A$  联想至  $B$  的可能性越高，反之， $A$  难以联想至  $B$ 。而且， $Rel(A \rightarrow B)$  不等于  $Rel(B \rightarrow A)$ 。在计算余弦相似度的基础上，本文的方法强调了两个概念的关联关系是基于某一个语义域的，而不仅仅只从概念自身的词向量维度信息而探讨二者的关系。并且，本文方法得到的联想度

具有方向性，满足后续合作网中需要模拟两个不同对象互动关系的需求。考虑到  $Rel(A \rightarrow B)$  的值区间在  $[-1,1]$  之间，为后续计算的方便，我们将  $Rel(A \rightarrow B)$  的原始值转化至  $[0,1]$  区间。若  $Rel(A \rightarrow B) > 0.5$ ，则认为 A 到 B 的联想度是较强的；若  $Rel(A \rightarrow B) < 0.5$ ，则认为 A 到 B 的联想度是较弱的。

## 5.2 合作网模型的构建

根据苏畅等人<sup>[35]</sup>的理论，隐喻中目标域和源域“求同存异”的过程可以表述为二者的属性之间具有合作机制，即源域主动向目标域展示源域的属性，目标域判断是否接受该属性。合作机制的表现形式可以为：目标域具有源域的某一个显著属性，目标域和源域直接达成合作；或是，目标域的属性与源域的某一属性是相似的，目标域同意将这个源域的属性作为合作的基础。

在互动论和合作机制理论的基础上，本文构建了一个合作网模型，用以表征隐喻句中目标域属性和源域属性、上下文和源域属性的合作关系。首先，我们给出了本章节的基本符号说明。

表 5.1：本章节的基本符号及说明

符号	说明
$S$	源域
$T$	目标域
$C$	上下文的词语集合，包括 $S$ 和 $T$
$Pro(x)$	某对象 $x$ 的属性集合

其中，上下文的词语集合表述为  $C = \{C_1, C_2, \dots, C_j, \dots, C_N\}$ ， $N$  为句中包括目标域和源域的词语总数。在本文中，我们关注目标域和源域的属性集合，即为  $Pro(T) = \{t_1, t_2, \dots, t_M\}$ ， $Pro(S) = \{v_1, v_2, \dots, v_K\}$ ， $M$  和  $K$  分别为目标域和源域的属性总个数。

### 5.2.1 合作强度的计算

合作强度体现了各个合作关系的耦合强弱。在本小节，我们定义了合作强度，并介绍合作强度的计算方法。

#### 定义 5.3：合作强度

给定两个不同的词语 $A$ 和 $B$ ，他们的合作强度 $Coop(A, B)$ 取决于两个词语各自与对方的联想关系，联想度 $Rel(A \rightarrow B)$ 和 $Rel(B \rightarrow A)$ 。合作强度的计算过程可以表述为 $Coop(A, B) = f(Rel(A \rightarrow B), Rel(B \rightarrow A))$ ，其中，函数 $f(\dots)$ 是综合衡量联想度的算子。合作强度 $Coop(A, B)$ 应该满足如下四个条件：

1.  $Coop(A, B) = Coop(B, A)$ ;
2. 若两个词 $A$ 和 $B$ 均具有较强的联想至另一个词的可能性，合并之后，二者的联想度可相互强化， $A$ 和 $B$ 有较强的意愿实现合作， $Coop(A, B)$ 则较为显著。
3. 若两个词均有较弱的联想至另一个词的可能性，合并之后，二者的联想度相互弱化，二者的合作强度不显著。
4. 若一方具有较强的联想度，而另一方具有较弱的联想度，即为二者的联想度是互为对立可相互抵消的趋向，那么 $Coop(A, B)$ 介于二者联想度之间。

Klement 等人<sup>[89]</sup>提出的联想补偿算子 (Relciative Compensatory Operator) 及苏畅等人提出的弱逻辑推理系统<sup>[90]</sup>实现了二元对象的组合运算。由此启发，我们设计了算子 $g(x, y)$ ，组合 $x$ 和 $y$ 各自的联想强度，得到二者的合作强度 $Coop(x, y)$ ：

$$g(x, y) = \begin{cases} \max\{\max(x, y) - 0.5, 0\} & \text{if } \min(x, y) = 0 \\ \min\{\min(x, y) + 0.5, 1\} & \text{if } \max(x, y) = 1 \\ \frac{x * y}{x * y + (1 - x)(1 - y)} & \text{else} \end{cases} \quad (5-3)$$

基于上述公式，我们发现算子 $g(x, y)$ 满足如下几个特征：

1. 若 $0 < x < 0.5$ 且 $0 < y < 0.5$ ，那么 $g(x, y) < \min\{x, y\}$ ；
2. 若 $0.5 < x < 1$ 且 $0.5 < y < 1$ ，那么 $g(x, y) > \max\{x, y\}$ ；
3. 若 $0 < x < 0.5$ 且 $0.5 < y < 1$ ，那么 $x < g(x, y) < y$ ；

$$4. \quad g(x, y) = g(y, x)$$

这些特征与合作强度的计算初衷相符。本文使用算子  $g(x, y)$  衡量合作强度是合理的。经由  $g(x, y)$ , 系统可以强化较显著的互动关系, 并削弱较微弱的关系, 从而实现后续隐喻意义的突显。于是, 我们得到合作强度的计算方式是:

$$\text{Coop}(A, B) = g(\text{Rel}(A \rightarrow B), \text{Rel}(B \rightarrow A)) \quad (5-4)$$

### 5.2.2 模型的基本结构

给定的两个词语  $A$  和  $B$ , 它们之间的合作机制如图 5.1 所示。

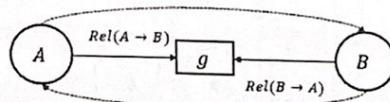


图 5.1: 两个词语  $A$  和  $B$  的合作机制图示。 $\text{Rel}(A \rightarrow B)$  指的是  $A$  联想至  $B$  的联想度;  $\text{Rel}(B \rightarrow A)$  指的是  $B$  联想至  $A$  的联想度。

合作网模型是一个包含诸多词语间合作机制的网络。根据概念隐喻理论<sup>[1]</sup>, 隐喻意义是源域向目标域映射的结果。在本文中, 我们认为源域向目标域展示属性。目标域根据自身的属性知识, 达成与源域的合作。同时, 由于隐喻意义取决于特定的语境<sup>[2]</sup>。我们还需考虑源域属性能否与隐喻句中的各个词语达成合作关系。对于任意一个源域属性  $v_k$ , 它与隐喻句中的上下文  $\{C_1, \dots, C_j, \dots, C_N\}$  及目标域属性集合  $\{t_1, \dots, t_m, \dots, t_M\}$  构成的合作网如图 5.2 所示。

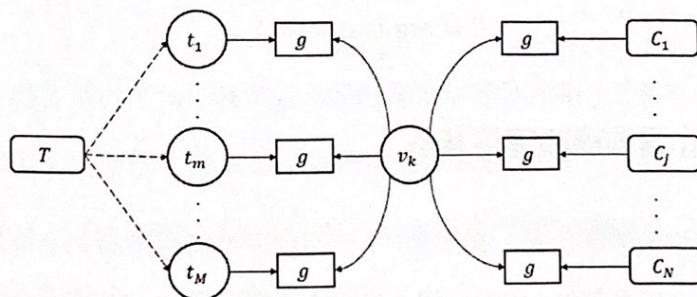


图 5.2: 某源域属性  $v_k$  和上下文及目标域属性的合作网图示。

在某一个隐喻句中，存在着由一系列源域属性  $Pro(S) = \{v_1, \dots, v_k, \dots, v_K\}$  构成的多个合作网。隐喻句中的多个合作网示意图如图 5.3 所示。

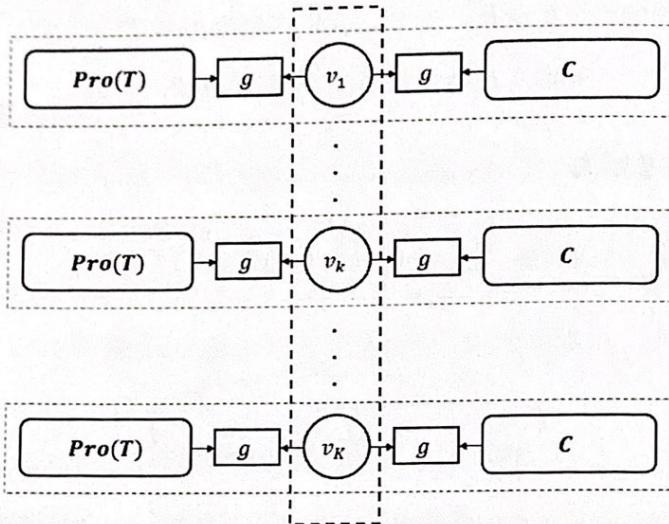


图 5.3：隐喻句中多个合作网的图示。每一个蓝色框表示一个以源域属性  $v_k$  为重心的合作网。

每个合作网均能输出在当前源域属性  $v_k$  的联结下，句子中各个合作关系的整体合作强度  $Y(v_k)$ ，其计算方式如下所示：

$$Y(v_k) = \frac{1}{M} \sum_{m=1}^M \text{Coop}(v_k, t_m) + \frac{1}{N} \sum_{j=1}^N \text{Coop}(v_k, C_j) \quad (5-5)$$

结合 Searle 的理论<sup>[28]</sup>，最终理解模型将根据各个源域属性的合作强度  $Y(v_k)$ ，选择一个具有最强合作强度的属性作为描述目标域的最佳释义  $R$ 。隐喻理解结果将以“ $T$ 是 $R$ ”的形式呈现：

$$R = \arg \max_{1 \leq i \leq K} Y(v_k) \quad (5-6)$$

### 5.3 基于合作网的隐喻理解算法

本文在合作网的基础上，提出了一个针对汉语名词性隐喻句的理解算法。为了提高理解算法的高效性，我们介绍了属性剪枝方法，解决属性生成过程中可能存在的冗杂性问题。