文件编号：Quickman -SWC2020-[TEAMNUMBER]

受控状态：■受控 □非受控

保密级别：□公司级 □部门级 ■项目级 □普通级

采纳标准：CMMI DEV V1.2





“妙记”自动笔记系统

**Auto Notes**

技术研究报告

**Version1.0.0**

2019.11.15

**Written by Quickman**



**All Rights Reserved**

目录

[1 问题聚焦 1](#_Toc24755120)

[1.1 问题描述 1](#_Toc24755121)

[1.2 问题抽象 1](#_Toc24755122)

[1.3 问题定位 1](#_Toc24755123)

[1.3.1 业务领域 1](#_Toc24755124)

[1.3.2 技术领域 1](#_Toc24755125)

[1.4 问题评估 2](#_Toc24755126)

[1.5 问题分解 2](#_Toc24755127)

[2 相关工作 3](#_Toc24755128)

[2.1 语音识别 3](#_Toc24755129)

[2.1.1 模型结构 3](#_Toc24755130)

[2.1.2 模型训练 3](#_Toc24755131)

[2.1.3 模型预测 4](#_Toc24755132)

[2.2 音源择优 4](#_Toc24755133)

[2.2.1 问题分析 4](#_Toc24755134)

[2.2.2 方法介绍 4](#_Toc24755135)

[2.2.3 主要处理过程 6](#_Toc24755136)

[2.3 图文识别 6](#_Toc24755137)

[2.4 自动摘要 8](#_Toc24755138)

[2.4.1 方法概述 8](#_Toc24755139)

[2.4.2 模型介绍 8](#_Toc24755140)

[2.4.3 算法流程 9](#_Toc24755141)

[2.4.4 LSTM 10](#_Toc24755142)

[2.5 关键词搜索 12](#_Toc24755143)

[2.5.1 模型概要 12](#_Toc24755144)

[2.5.2 算法流程 12](#_Toc24755145)

[2.6 短文本关键词提取 13](#_Toc24755146)

[2.7 知识图谱构建 14](#_Toc24755147)

[2.8 文字整合 16](#_Toc24755148)

[3 技术方案 16](#_Toc24755149)

[3.1 技术方向 16](#_Toc24755150)

[3.2 技术选择 17](#_Toc24755151)

[3.3 结果期望 17](#_Toc24755152)

[4 技术实践 19](#_Toc24755153)

[4.1 使用的开发框架及依赖的Library 19](#_Toc24755154)

[4.2 技术实践过程 19](#_Toc24755155)

[5 结果验证 19](#_Toc24755156)

记录更改历史

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **序号** | **更改原因** | **版本** | **作者** | **更改日期** | **备 注** |
| 1 | 创建 | V0.1.0 | 队员3 | 2019.11.01 | 前三个（初赛）部分 |
| 2 | 更新 | V0.1.1 | 队员3 | 2019.11.14 | 问题聚焦和技术方案 |
| 3 | 更新 | V1.0.0 | 队员3 | 2019.11.15 | 部分内容修改，审查全文 |

# 问题聚焦

## 问题描述

1. 如何将用户输入的声音，图片转换为文字
2. 如何根据识别出的文字，进行文字学习与处理，提取关键词，构建知识图谱，然后整合文档生产文本摘要
3. 如何感知文本中关键词的词义，分理处生僻词并为用户提供解释

## 问题抽象

1. 如何通过训练模型将语音信号和图片转为文字
2. 如何通过统计学，概率学加深度学习方法构建模型提取关键词，以关键词为基础构建知识图谱。
3. 如何利用概率学和统计学在不破坏原文上下文关系的基础上提取整合出完整文档，再根据文档生成摘要
4. 如何根据用户需求和用户输入感知关键词义，然后返回用户可能想知道的关键词的解释

## 问题定位

### 业务领域

1. 用户打开之后即可做到即时记录。
2. 根据内容的关键词构建知识图谱，实现关键词自动搜索方便用户快速理解内容。
3. 文档内容自动整理解放用户双手，留给用户更多思考空间。

### 技术领域

1. 构建基于深度学习的一个快速文字识别模型。
2. 利用识别得出的文字，构建一个多功能文字处理平台，实现关键词提取，知识图谱构建，综合文档整理，文档摘要，关键词搜索与解释等功能。

## 问题评估

1. 技术性：采用较新的成熟算法，针对部分对于准确率要求高的功能，使用开源或成熟技术获取结果
2. 实用性：通过自动笔记应用大大提升用户参加各种会议，访谈，讲座的效率，将用户从记录笔记的工作中解放。
3. 便捷性：基于快应用技术，一键面对面开启记录体现其“快速”，“敏捷”，同时无需单独下载节约用户时间。

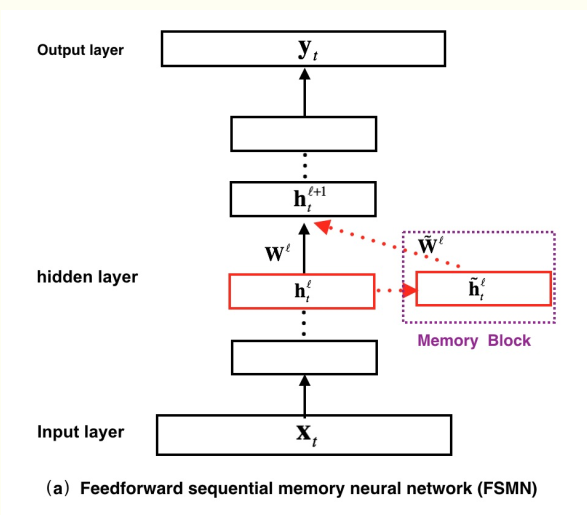
## 问题分解

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 核心功能编号 | 问题描述 | 子问题编号 | 子问题描述 | 子问题难度 | 依赖关系 |
| 1 | 文字识别 | 1-1 | 语音整合与识别 | B | 依赖与输入的语音样本 |
| 1-2 | 图像处理与识别 | B | 依赖于输入的图片样本 |
| 2 | 文本学习 | 2-1 | 关键词提取，获取多段文本的关键词序列 | C | 依赖1输出的文本 |
| 2-2 | 知识图谱构建，获取关键词关联 | A | 依赖1输出的文本以及2-1输出的关键词列表 |
| 2-3 | 文档整理 | B | 依赖1输出的文本以及2-1得到的关键词序列 |
| 2-4 | 文本摘要生成 | A | 依赖2-1得到的关键词序列以及2-3得到的文本 |
| 3 | 词义感知 | 3-1 | 生僻词感知 | A | 以2-1得到的关键词序列和2-2得到的关键词关联为基础 |
| 3-2 | 词义解释 | B | 以3-1得到的关键词序列为基础 |

# 相关工作

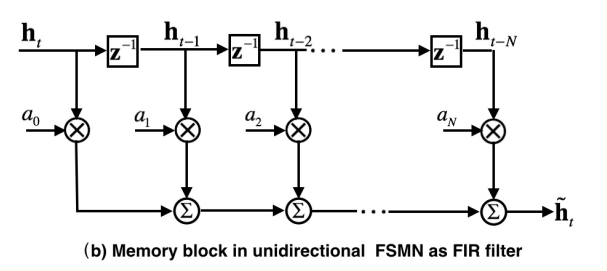
## 语音识别

### 模型结构



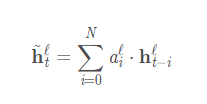
 FSMN 的结构如上图所示，在图中我们可以看到，FSMN 对比普通的前馈神经网络，最大的区别就是引入了 Memory Block..

### 模型训练

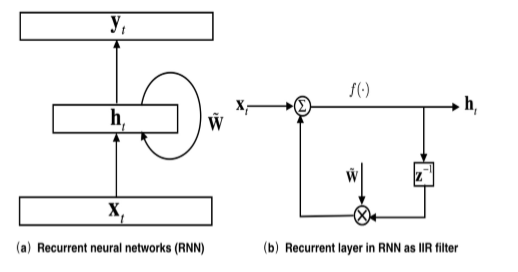


由上图可看出Memory Block中memory的更新过程，同时Memory可以看成一个FIR滤波器。

该Memory Block的表达式如下：



其中为时刻层的特征值，为对应的时不变系数。 这里可以看出，Memory Block中储存了N个的“过去记忆”，该“过去记忆会与当前层的特征值一起前向传入下一层。



如图，RNN中的循环结构可以看成一个IIR滤波器（为上一个时刻导数的Z变换。）由于，IIR滤波器可以通过高阶的FIR滤波器逼近表示。所以FSMN中的Memory Block可以看成是RNN中循环结构的近似。

### 模型预测

音频通过预处理之后再经过FSMN就可以得到输入音频片段对应的字词的概率大小，根据片段将概率大的字词按顺序合成之后，搜索已构建好的词库筛选出正常词汇，即可得出结果。

## 音源择优

### 问题分析

由于本软件用户多样化，应用场景颇为复杂，收集到的语音幸喜可能会受到各种各样的噪声干扰，故拟采用基于多采样的低信噪比语音增强方法来应对噪声污染。

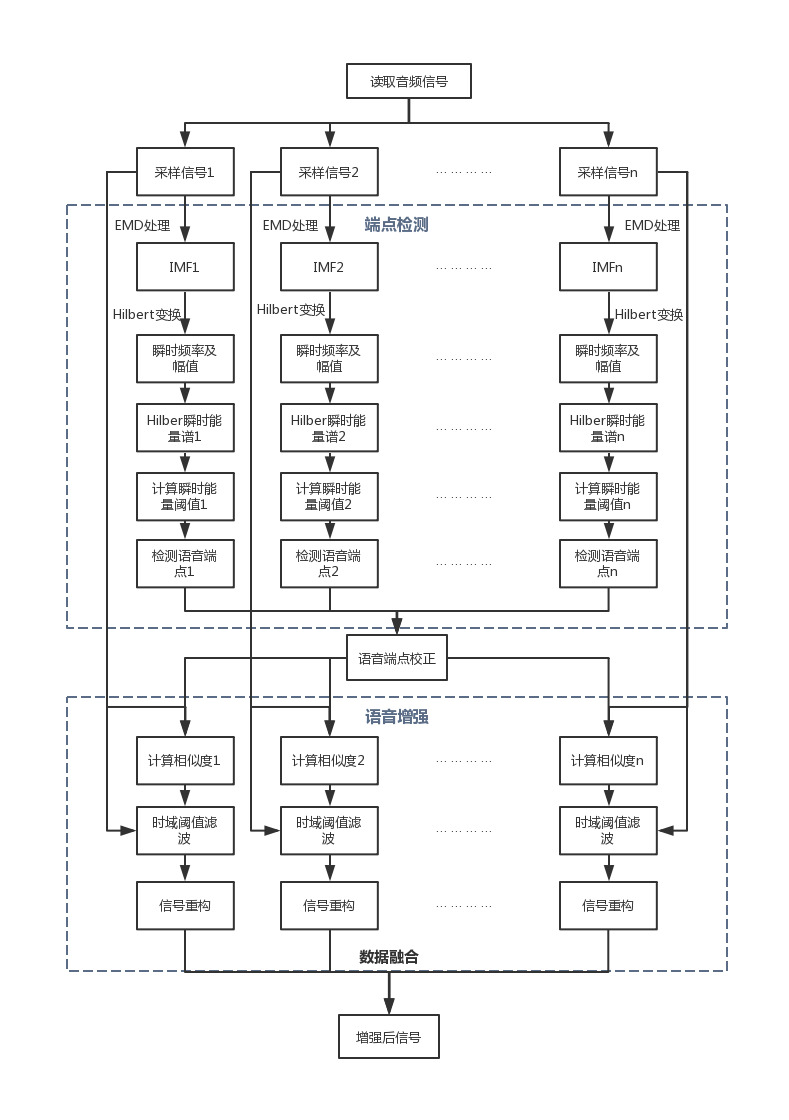
### 方法介绍

1. 多采样方法：

由于快应用的特性，客户端只负责收集和发送声音，不进行数据存储，故只能分段发送给服务器，为保证语音质量，在后续处理过程中通过数据融合来减少非理想因素对语音处理效果的影响。由于语音信号具有短时平稳性，可认为在10ms~30ms内的语音信号分布近似不变，因此在处理信号之前将其分成连续小段再进行处理。

分帧采用交叠分段的方法，此方法使得帧与帧之间平滑过渡，保持了信号的连续性，分帧通过有限长窗口加权滑动来实现。

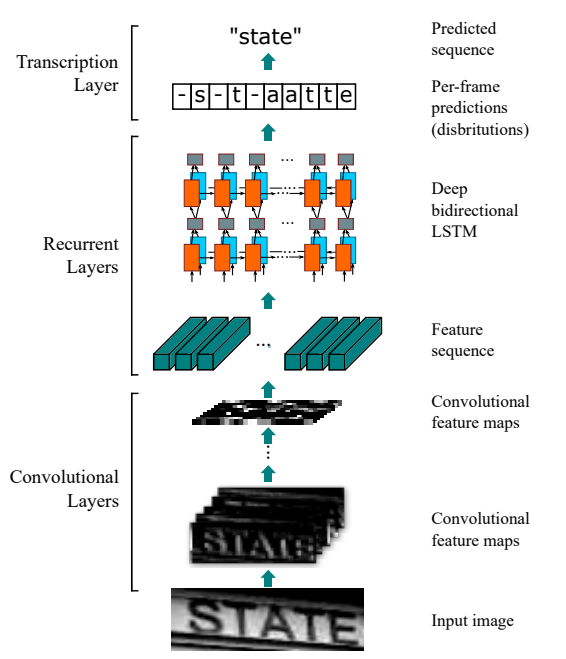
1. 算法主要流程如图：



### 主要处理过程

1. 低信噪比语音端点检测，使用希尔伯特—黄变换的语音信号检测算法
2. 以相同采样频率采样，并分割成相邻没有重叠的信号帧。
3. 对每个不同帧长的带噪语音信号各帧进行EMD处理
4. 对各IMF分量进行Hilbert变换，求解瞬时频率与幅值
5. 对每组数据分别合成Hilbert谱
6. 利用合成的Hilbert谱计算获得信号瞬时能量谱，并做平滑处理
7. 选取没有语音的片段用作背景噪声估计。通过处理无语音信号帧的瞬时能量谱得到每个语音信号检测的阈值
8. 利用预测判断语音段和非语音段，并在带噪信号中标记出语音的起始和截止
9. 选取每个语音片段的起始点和截止点
10. 基于EMD（经验模态分解）——最大相似度的语音增强
11. 获取带噪语音信号片段
12. 提取与检测到语音信号等长的噪声信号
13. 对带噪语音信号进行级值域均值模式分解
14. 计算分解得到的固定模态分量间的最大相似度
15. 确定阈值
16. 对大于阈值的进行滤波处理
17. 将处理后的和未处理过的其他模态分量融合得到增强后的信号
18. 将增强后的信号进行数据融合

## 图文识别



如图所示，CRNN模型 的网络架构由三部分组成，包括卷积层、循环层和转录层（由下到上）。在 CRNN 的底部，卷积层自动从每个输入图像中提取特征序列。在卷积网络层之上，构建一个对卷积层特征序列的每一帧进行预测的循环网络层。在 CRNN 顶部的转录层将循环层输出的帧预测转化为标签序列。

卷积层：卷积层部分由标准 CNN 模型中的卷积层和最大池化层组成。这一部分用来提取表示输入图片的序列特征。在输入到网络之前，所有的的图片需要被缩放到同一高度。然后，一些列的特征向量从特征图谱中被提取出来。这些特征图谱由卷积层产生，作为循环层的输入。

循环层：卷积层之上是一个深度Bi-LSTM层来作为循环层。循环层为特征序列的每一帧预测一个标签分布。循环层的优点有三重。首先，RNN 具有很强的捕获序列上下文信息的能力。其次，RNN 可以将误差值反向传播到其输入层，即卷积层。最后，RNN 能够从头到尾对任意长度的序列进行操作。

转录层：转录是将 RNN 所做的每帧预测转换成标签序列的过程。在实践中，存在两种转录模式，即无词典转录和基于词典的转录。在无词典模式中，使用极大似然估计方法，预测时将概率最大的标签认为是结果。在基于词典的模式中，通过选择具有最高概率的标签序列进行预测。概率序列的产生采用由格雷夫斯（Graves）提出的连接主义时间分类（CTC）层中定义的条件概率。

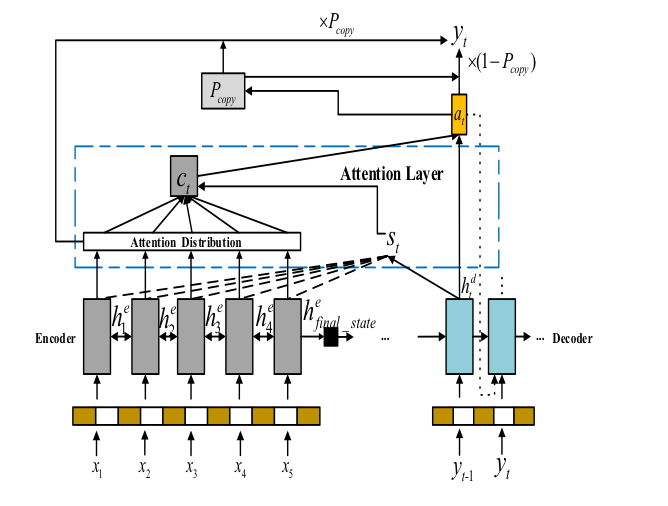
## 自动摘要

### 方法概述

首先使用以及训练好的词嵌入模型获取词向量。然后使用LCSTS数据集训练模型。训练完成之后输入已经生成的完整文档，输出文档摘要。

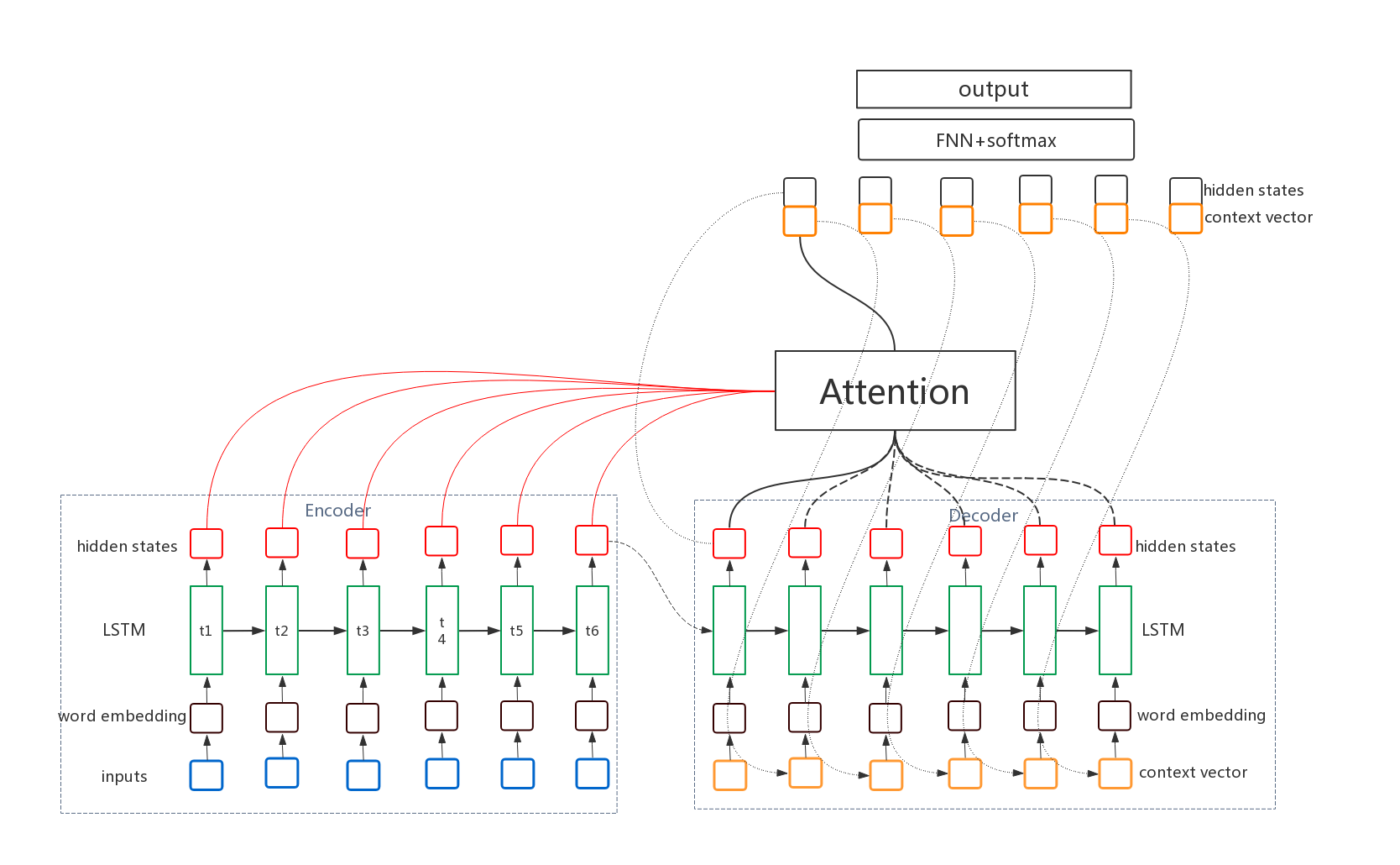
### 模型介绍

模型主体由编码器，解码器，注意力层和复制软转换四部分组成。结构如图所示。



Seq2seq+attention原理：

seq2seq 是一个Encoder–Decoder 结构的网络，它的输入是一个序列，输出也是一个序列， Encoder 中将一个可变长度的信号序列变为固定长度的向量表达，Decoder 将这个固定长度的向量变成可变长度的目标的信号序列。



（左下方的网络为encoder，右下方的网络为decoder）

Encoder负责输入单词序列（也就是句子）通过将单词序列中的单词根据序列的顺序来获得序列中的隐含内容，decoder负责根据encoder输出的隐含内容来将隐含内容转换为单词序列中每个单词的生成概率，在decoder解码的时候，利用decoder获得的隐含内容和encoder的输出来获取每个时刻下单词序列的注意力，根据注意力来调整每个时刻下单词的生成概率。

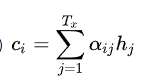
### 算法流程

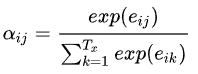
输入：

输出：

(1)  , Encoder方面接受的是每一个单词word embedding，和上一个时间点的hidden state。输出的是这个时间点的hidden state。

(2)  ， Decoder方面接受的是目标句子里单词的word embedding，和上一个时间点的hidden state。

(3)  , context vector是一个对于encoder输出的hidden states的一个加权平均。

(4)  , 每一个encoder的hidden states对应的权重。

(5)  , 通过decoder的hidden states加上encoder的hidden states来计算一个分数，用于计算权重(4)

(6) , 将context vector 和 decoder的hidden states 串起来。

(7)  ，计算最后的输出概率。

### LSTM

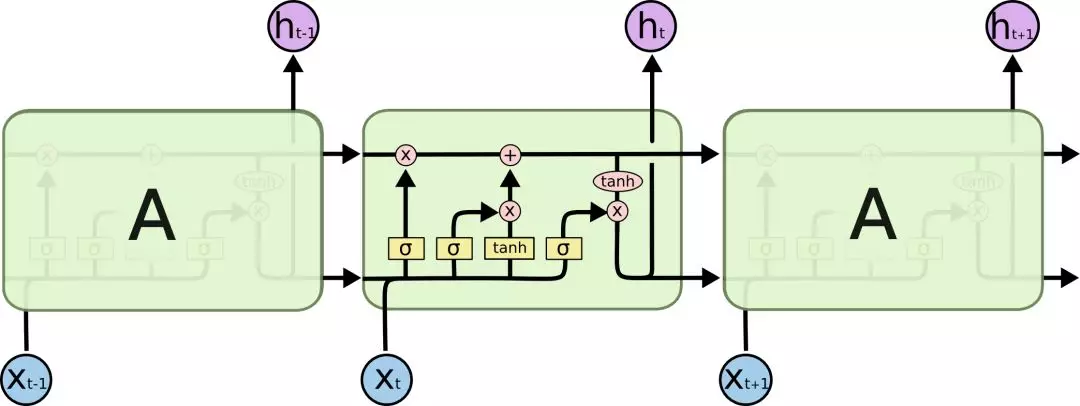
编码器：（Bi-LSTM原理加入）使用单层Bi-LSTM将输入字符进行编码，在编码完成后将最后一个时刻的编码状态作为解码层的初始化。

LSTM的全称是Long Short-Term Memory，它是RNN（Recurrent Neural Network）的一种。LSTM由于其设计的特点，非常适合用于对时序数据的建模。

BiLSTM是Bi-directional Long Short-Term Memory的缩写，是由前向LSTM与后向LSTM组合而成。

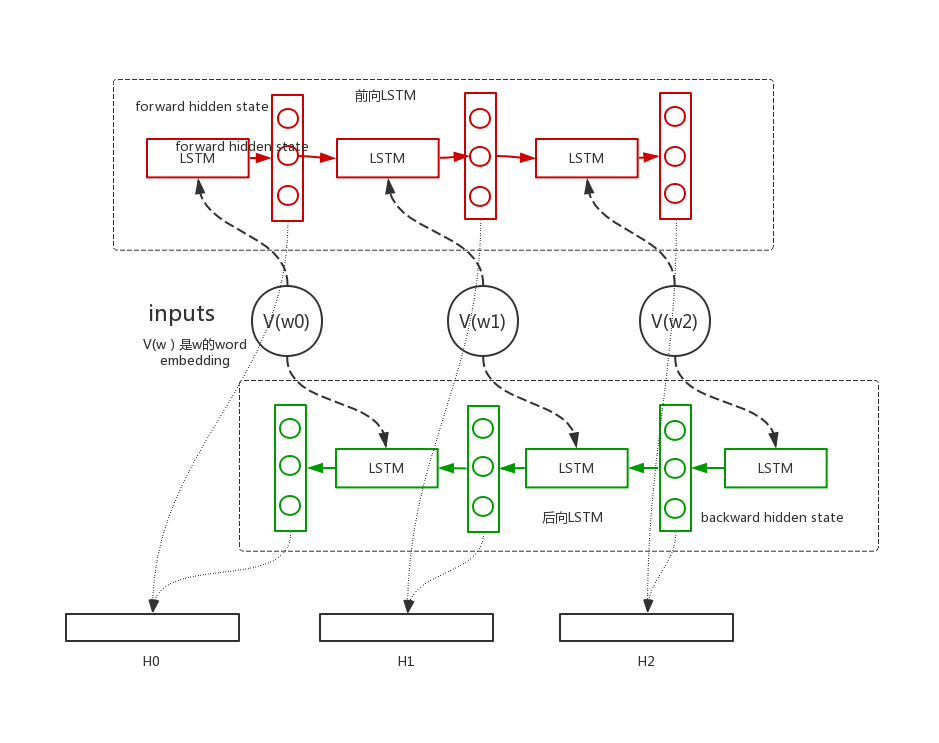
LSTM模型是由时刻的输入词，细胞状态 ，临时细胞状态，隐层状态，遗忘门，记忆门，输出门组成。LSTM的计算过程可以概括为，通过对细胞状态中信息遗忘和记忆新的信息使得对后续时刻计算有用的信息得以传递，而无用的信息被丢弃，并在每个时间步都会输出隐层状态，其中遗忘，记忆与输出由通过上个时刻的隐层状态和当前输入计算出来的遗忘门，记忆门，输出门来控制。

总体框架如图所示。

图1. LSTM总体框架

Bi-LSTM：使用LSTM模型可以更好的捕捉到较长距离的依赖关系。因为LSTM通过训练过程可以学到记忆哪些信息和遗忘哪些信息。但是利用LSTM对句子进行建模也存在一个问题：无法编码从后到前的信息。而通过BiLSTM可以更好的捕捉双向的语义依赖。

前向的LSTM与后向的LSTM结合成BiLSTM。



上图中，前向LSTM获得句子的顺序语义信息，后向LSTM获得句子的逆序语义信息，通过将前向LSTM和后向LSTM得到的结果拼接就可以得到表示单词的前后语义依赖的hidden states。

解码器：（GlobalAttention原理）加入GlobalAttention层，在计算每个解码时刻对应的注意力分布时考虑全局编码器中每个词的隐层状态。将注意力和编码状态加权平均后可以求出某时刻预测输出词在词典中的概率。

假设t 时刻 LSTM 单元输出的 解码层状态为，则的求解为:

表示时刻的解码层状态，为t 时刻 输入的词嵌入向量。接着使用一个 score 函数将t 时刻的解码状态与第i 个输入位置的编码状态实现对齐：

这里选择了作为 score 函数，是对向量，对应了两个向量的相似度大小。对 ,进行归一化处理，得 到解码时刻t 对的注意力大小为

根据式(6)求得解码时刻t 的上下文向量为 :

是所有编码状态与对应注意力大小乘积的加权平均。

将与 拼接并使用 tanh 函数激活，得到t 时刻注意力向量为:

由于每个时刻生成的词是从众多词中进行选择的，属于多分类问题，因此使用了函数求出t 时刻测输出词在词典中的概率：

复制机制：在整个词典中，所有词的输出概率由两部组成（a）源序列中每个词的注意力大小（b）网络自主生成词的概率大小（解码器的输出），因此将注意力和固定概率相加就相当于动态扩展原有的词典。

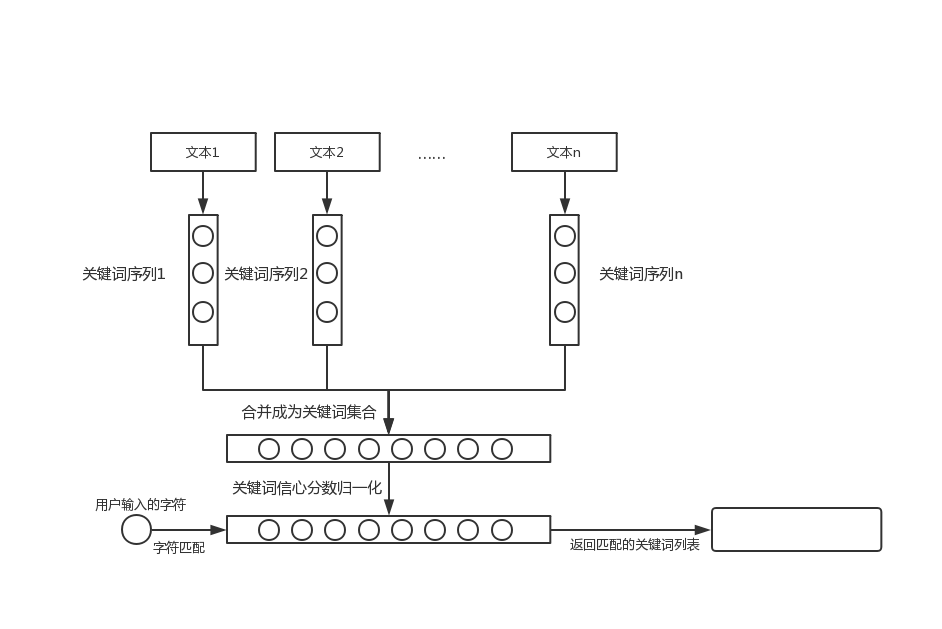
Input-feeding：使用input-feeding对上一个时刻的注意力进行跟踪来确保注意力做出的选择更加准确。

模型推断：使用长度规范化的beam-search算法来最大化得分函数，输出预测词。

## 关键词搜索

### 模型概要

已有关键词序列+关键词的概率，根据用户输入的字符做字符匹配，依概率大小进行排序），其基本流程如图所示：

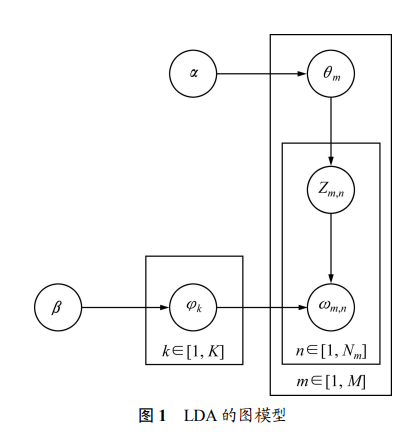


### 算法流程

1. 根据已有的多段文本获取每段文本的关键词序列以及序列中关键词对应的信心分数。
2. 将所有关键词序列整合为关键词字典，关键词对应的信心分数为每个序列中关键词的信心分数和。
3. 对所有关键词的信心分数做归一化处理。
4. 根据用户输入对关键词做字符匹配，返回有关联的关键词列表，依照关键词对应的信心分数排序

## 短文本关键词提取

这里我们使用了基于LDA+TFITF方法来提取文本关键词。

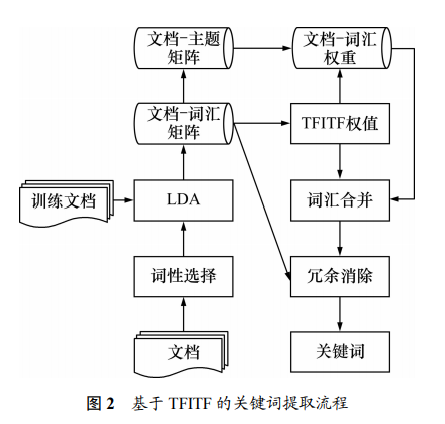


上图中：φk 为主题 k 中的词汇概率分布，θm为第 m篇文档的主题概率分布，φk 和θm服从Dirichlet 分布， φk  和 θm 作为多项式分布的参数分别用于生成主题和 单词；α 和 β 分别为 φk 和 θm的分布参数，α 反映了文 档集中隐含主题间的相对强弱，β  为所有隐含主题自 身的概率分布；K 为主题数目；M 为文档集中文档数 目；Nm 为第 m 篇文档的词总数；ωm,n 和 Zm,n分别为第 m 篇文档中第 n 个单词及其隐含主题。

LDA 模型包含词、主题和文档三层结构，如图 1  所示。

LDA以 pLSI 为基础的一个服从 Dirichlet 分布的 K 维隐含随机变量表示文档的主题概率分布，模拟文档的产生过程；通过对参数 β 施加 Dirichlet 先验分布，使得 LDA 模型成 为一个完整的生成模型。

TFITF 模型在分析文档主题信息的基础上进行文 档关键词抽取，基本流程如下图 所示，：



分为五个基本步骤：

1. 预处理： 为排除无关信息对训练主题模型和关 键词抽取的干扰， 依据关键词多为名词性短语的特点， 对训练主题文档和待抽取关键短语文档分别进行分 词、词性标注和选取名词与形容词等预处理。
2. 主题分析：利用训练文档集产生主题模型，预 测新文档集的词汇对主题的权重以及主题对文档的权 重。
3. TFITF 权值计算：根据词汇对主题的权重以及 词汇在所有主题中出现频率计算词汇对主题的 TFITF 权值，并进一步计算词汇对文档的权重。
4. 词汇合并： 利用词性搭配规则从未预处理的原 始文档筛选出二元候选短语，根据候选短语中各词汇 的共现率以及候选短语权重计算短语构成候选关键短 语的权值，对预处理文档中二元短语进行排序形成候 选关键短语集合{Hi}。
5. 冗余消除：根据短语的主题分布筛选{Hi}权值 较大且反映文档不同子主题的Q个短语构成关键词短语输出。

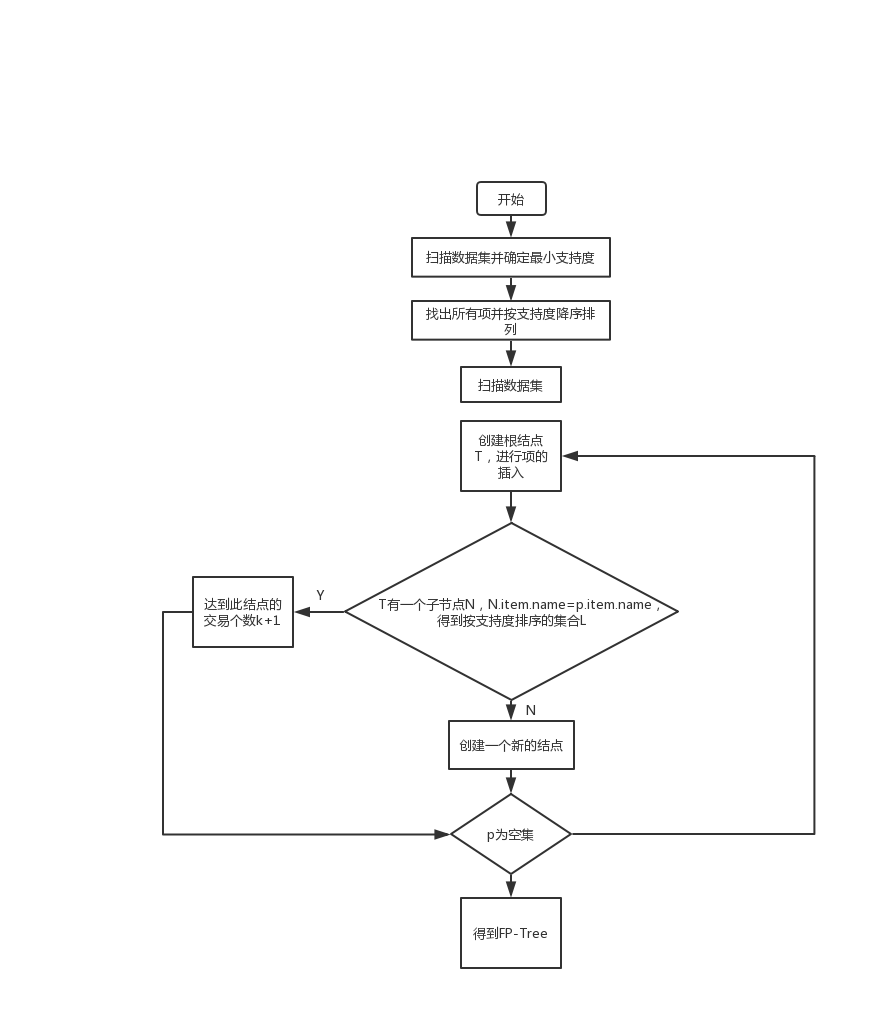
## 知识图谱构建

本方法基于FP-growth算法，根据文本内容以及提取到的关键词构建树图。

FP-growth算法通过将数据集存储在FP（Frequent Pattern)树上发现频繁项集。FP-growth算法只需要对数据库进行两次扫描，而Apriori算法在求每个潜在的频繁项集时都需要扫描一次数据集，所以说Apriori算法是高效的。其中算法发现频繁项集的过程是：

1. 构建FP树；
2. 从FP树中挖掘频繁项

由于构建知识图谱不需要挖掘频繁项，所以只参考FP-tree的生成过程，如下图：

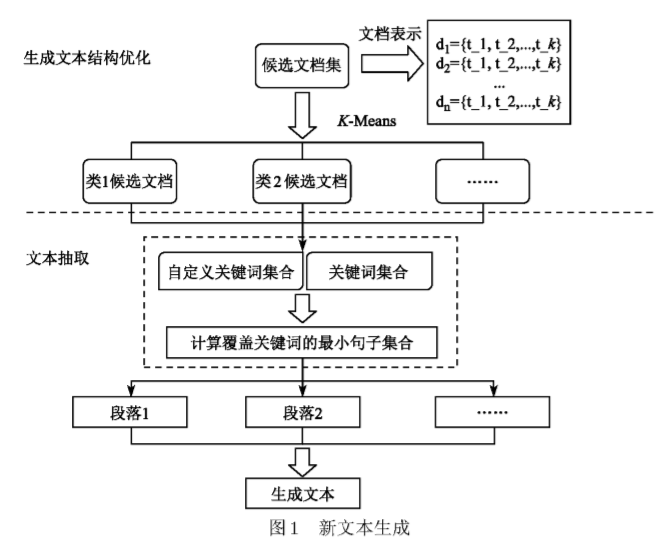


基本流程如下：

1. 扫描整个数据集，得到所有出现过的数据，并得到它们的支持度计数作为候选屏藩1项集。其中，。为该关键字的出现次数，为每个项集中关键字的信心分数。
2. 构造FP-Tree：
   1. 创建树的根节点，用null标记。
   2. 第二次扫描数据集，每个事务中的项都按照递减支持度计数排序，并对每个事务都创建一个分支。
   3. 当为一个事物考虑增加分支时，沿共同的前缀的每个节点的计数增加，为前缀之后的项创建节点和连接。
   4. 构建完成后将根节点删除，根据左右子节点的支持度大小提升子节点。

## 文字整合

按照主题分布对候选文档集（这里是短文本）分类，用每类生成一段主题相同的文本，使得机器文本依照不同主题分段论述，将候选文档集按照主题分类后，首先对每类文档提取关键词集合和自定义关键词集合，抽取与关键词相关的句子，组成文本。流程如图：



在生成文本结构优化阶段，首先通过ＬDＡ主题模型提取候选文档主题分布，然后使用主题分布向量表示文档，最后对文档进行聚类．在文本抽取阶段，先对每类文档采用TF-IDF技术提取关键词集合， 并获得每类文档的自定义关键词集合 ，然后抽取出满足关键词覆盖的最小句子集合组成一个段落，最后输出生成文本。

# 技术方案

## 技术方向

本项目使用了如下技术：

1. 深度学习
2. 自然语言处理
3. 数据挖掘

## 技术选择

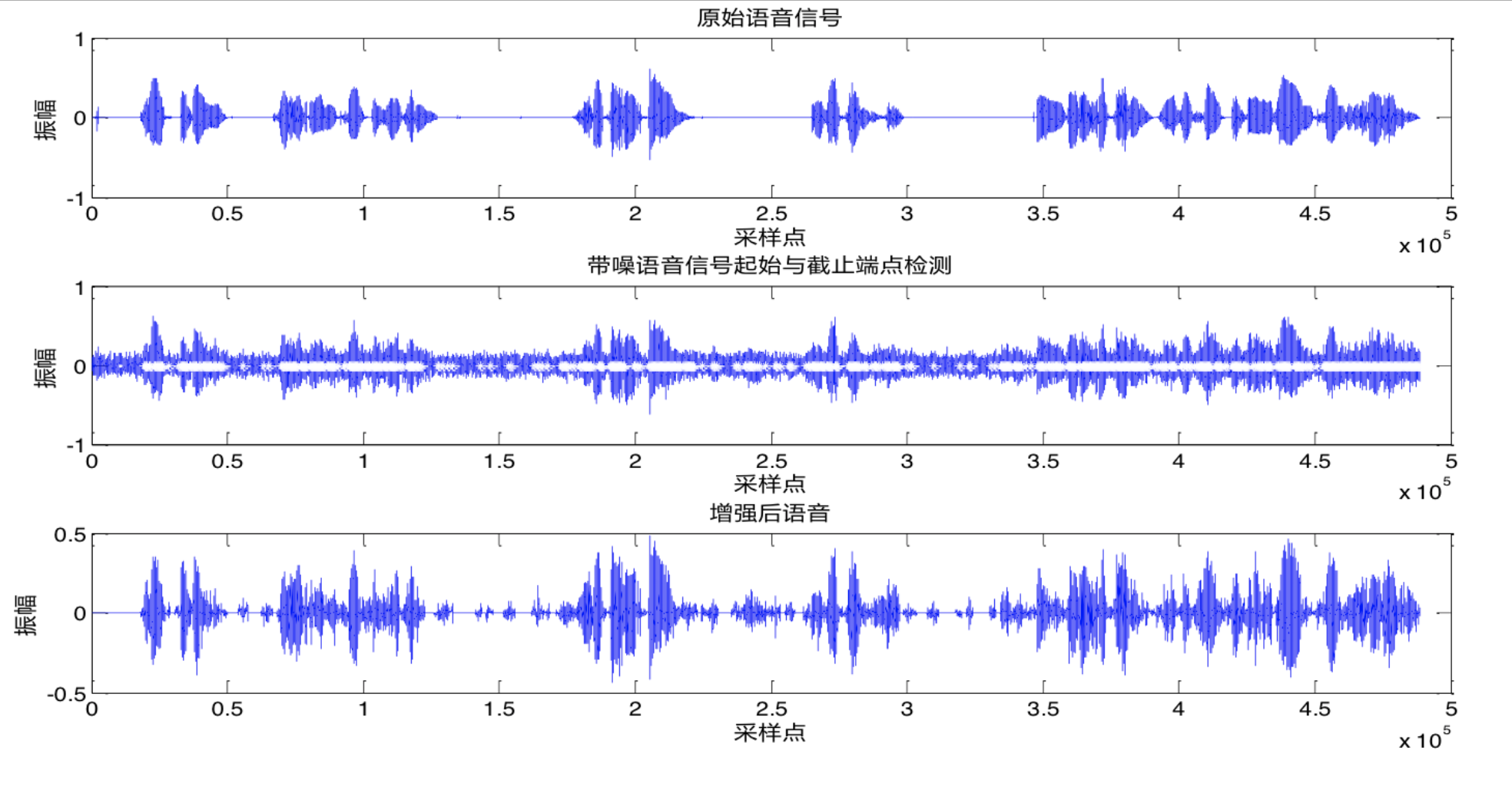
结合系统的功能，选择一下备选技术：

1. FSMN模型
2. CRNN模型
3. seq2seq模型
4. LSTM模型
5. LDA和TFITF方法
6. FP-growth算法

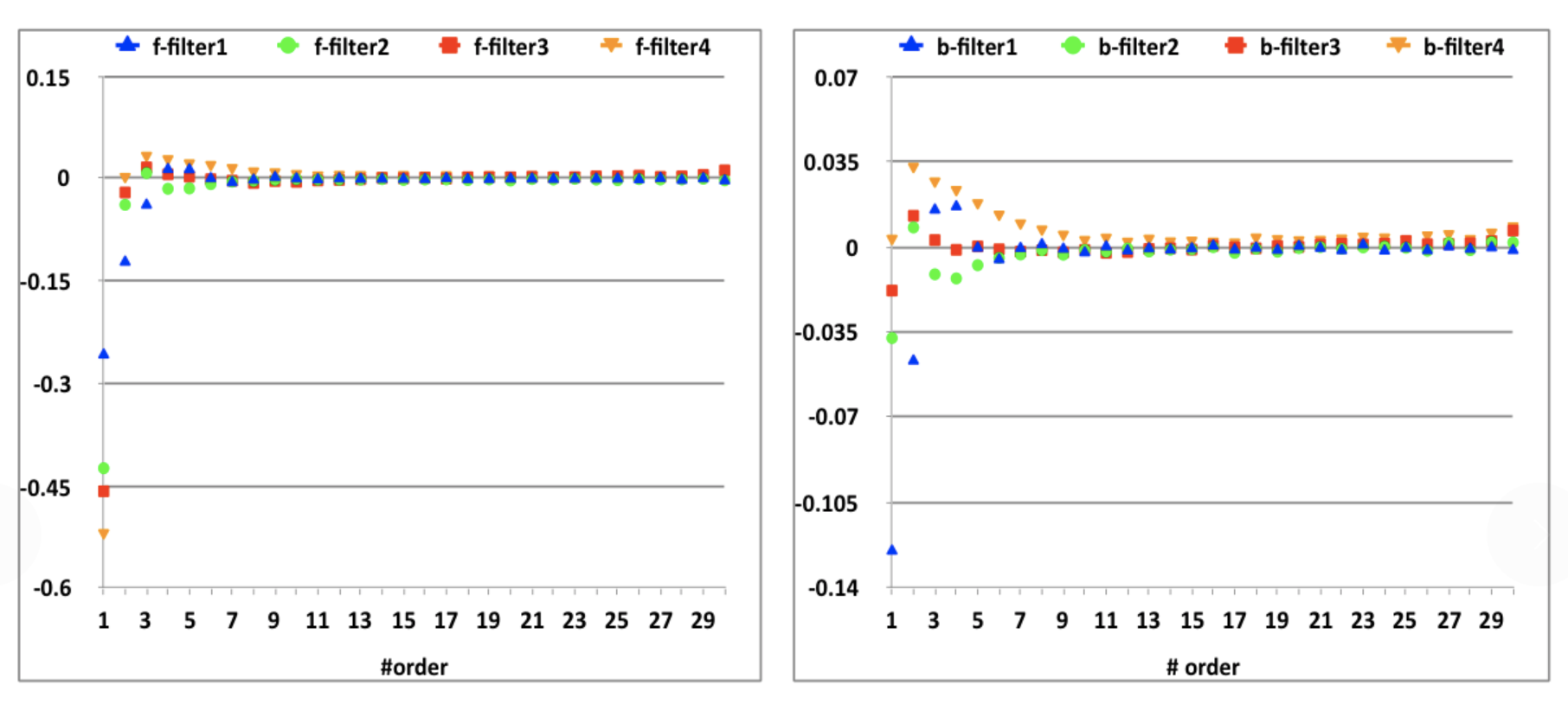
## 结果期望

以下是使用相关技术后，对期望结果的可视化：

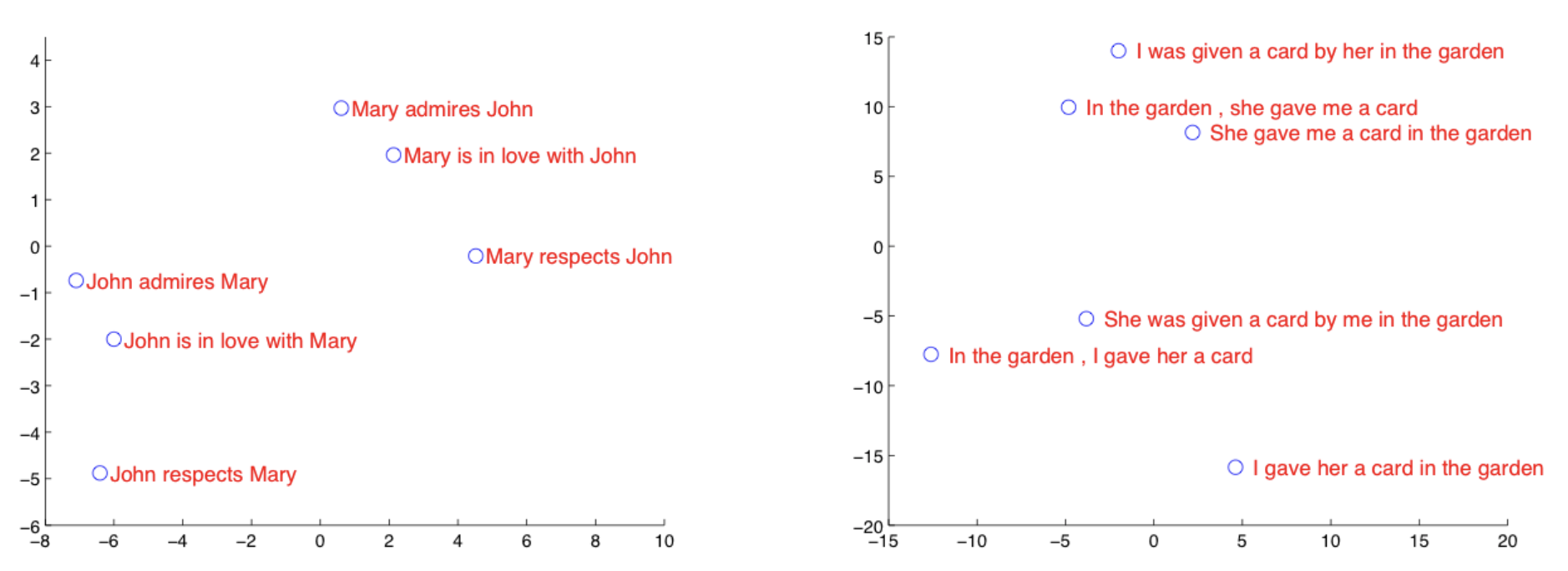
1. 语音端点检测与增强效果



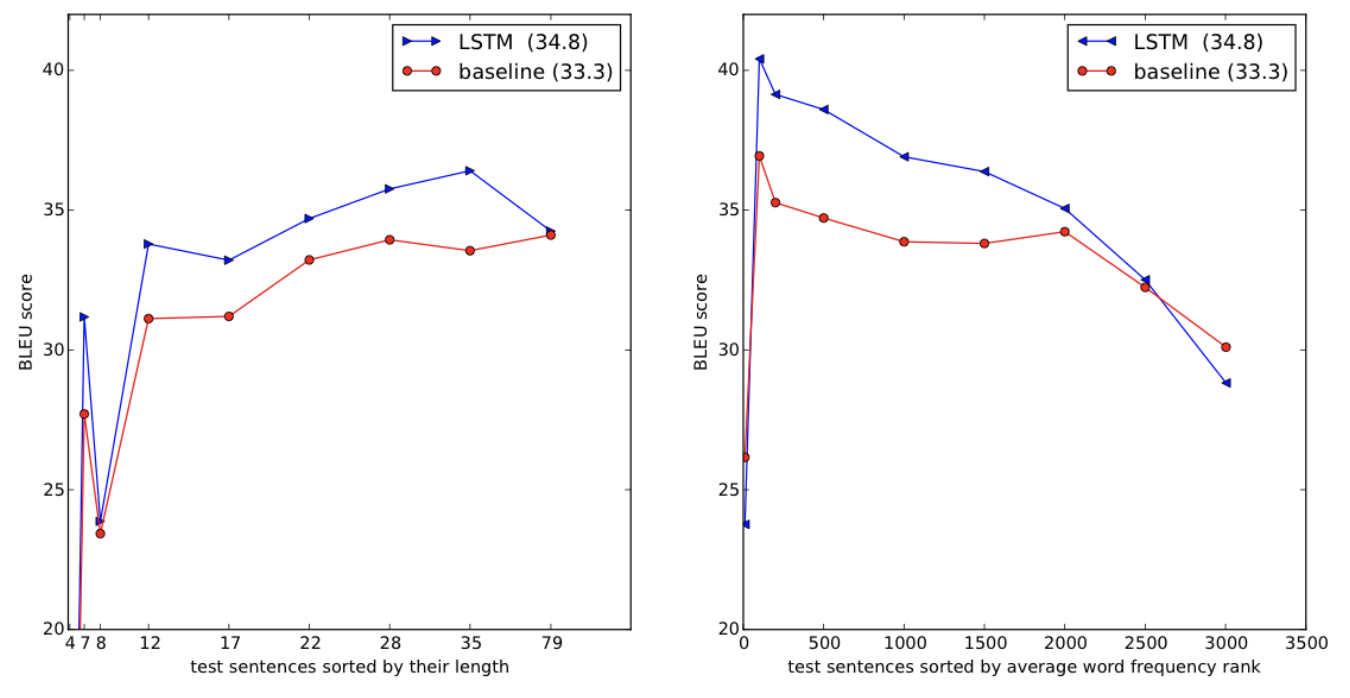
1. FSMN模型中的学得的过滤器



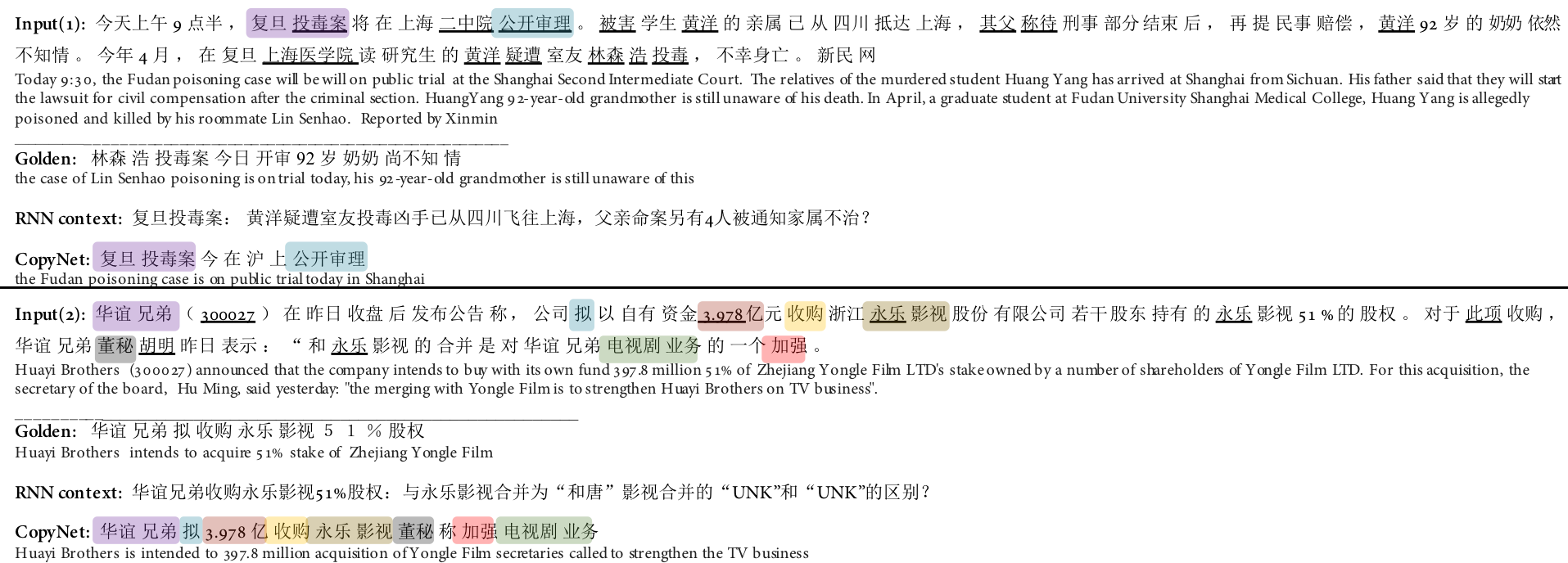
1. LSTM的2维PCA投影



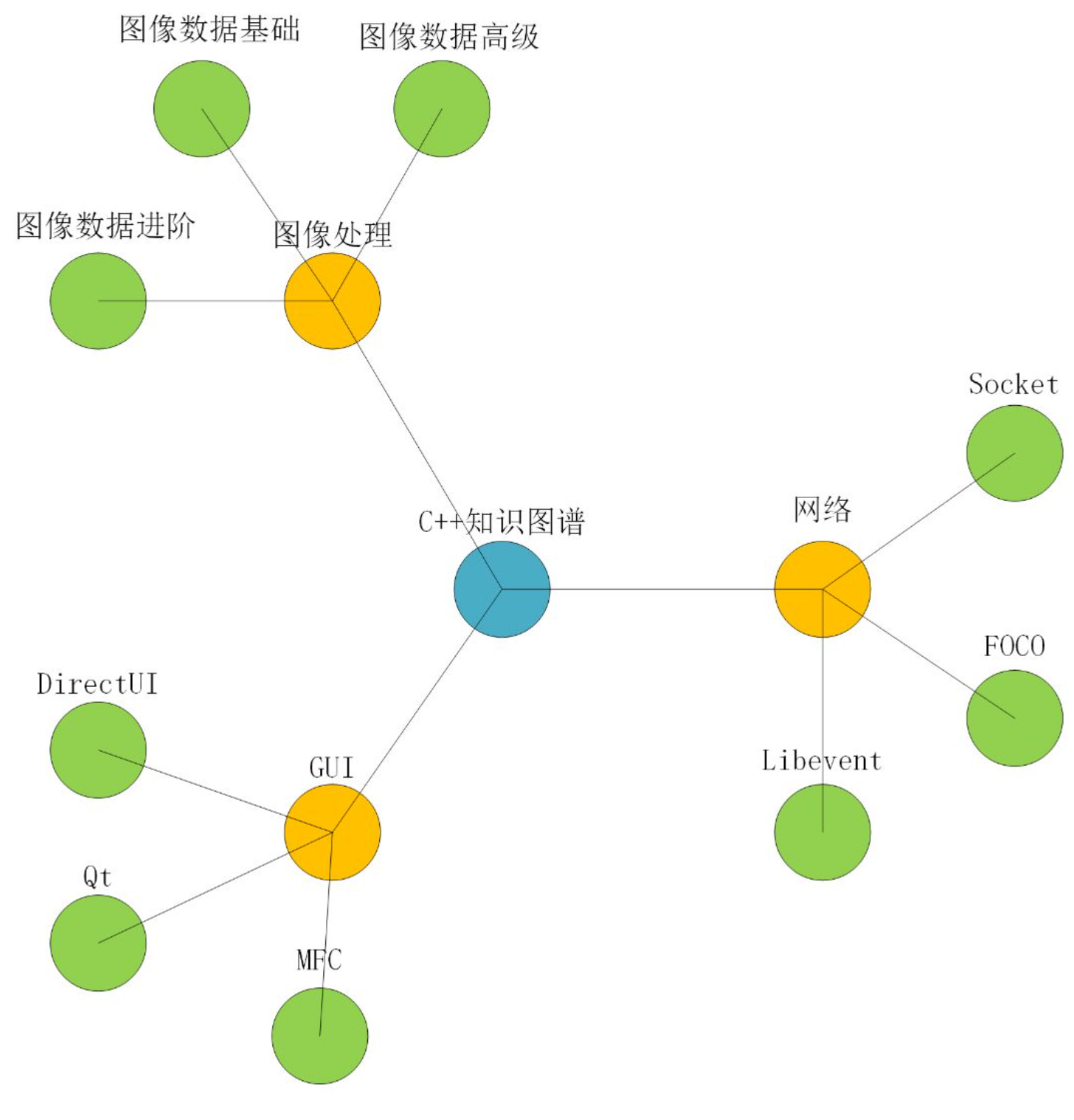
1. LSTM性能



1. 自动摘要



1. 知识图谱



# 技术实践

## 使用的开发框架及依赖的Library

## 技术实践过程

# 结果验证