

# 里德堡原子基态拓扑量子信息传输

答辩人：杨微羽    指导教师：吴金雷

哈尔滨工业大学 (威海) 理学院

2022 年 11 月 3 日

# 目录

- 1 引言
  - 研究背景
  - 主要工作
- 2 词表示模型
- 3 直接表示模型
- 4 研究方法 with 数据集特征
- 5 算法和代码
  - 算法
  - 代码
- 6 Future Work

# 目录

## 1 引言

- 研究背景
- 主要工作

## 2 词表示模型

### 3 直接表示模型

## 4 研究方法与数据集特征

## 5 算法和代码

- 算法
- 代码

## 6 Future Work

# 研究背景



图: SUSTech Campus

- 短信息 (SMS) 成为现代通讯的重要组成部分
  - 很多组织或网站使用短信息作为身份验证的辅助通道

# 研究背景



图: SUSTech Campus

- 短信息 (SMS) 成为现代通讯的重要组成部分
  - 很多组织或网站使用短信息作为身份验证的辅助通道
- 现代短消息的发送，在抵达终端之前不接触蜂窝网络
  - 短信息 (SMS) 成为现代通讯的重要组成部分



# OTT 服务



图: OTT 服务

OTT 服务支持在数据网络上提供短信和语音等第三方服务。  
OTT 可以使用云服务来存储和同步 SMS 到用户的其他设备。

# 目录

## 1 引言

- 研究背景
- 主要工作

## 2 词表示模型

### 3 直接表示模型

## 4 研究方法与数据集特征

## 5 算法和代码

- 算法
- 代码

## 6 Future Work



# 词表示

在 NLP 任务中，可以利用各种词表示模型，将“词”这种符号信息表示成数学上的向量形式。。将语义信息表示成稠密、低维的实值向量，这样就可以用计算向量之间相似度的方法（如余弦相似度），来计算语义的相似度。词的向量表示可以作为各种深度学习模型的输入来使用

## 词表示模型分类

### 直接表示模型

- One-Hot Representation

### 分布式表示模型

- 计数模型（基于共现矩阵）
- 预测模型（基于神经网络）



# One-Hot Representation

最简单直接的词表示是 One-Hot Representation。考虑一个词表  $\mathbb{V}$ ，里面的每一个词  $w_i$  都有一个编号  $i \in \{1, \dots, n\}$ ，那么词  $w_i$  的 one-hot 表示就是一个维度为  $n$  的向量，其中第  $i$  个元素值非零，其余元素全为 0。例如：

$$w_2 = [0, 1, 0, \dots, 0]^T$$

$$w_3 = [0, 0, 1, \dots, 0]^T$$

## 缺点

- 彼此正交，不能反应词间的语义关系
- 稀疏表示，维度很高，和词典大小成正比

仅仅是为了区分词，不包含语义信息，语义信息应该从上下文中挖掘



# 目录

## 1 引言

- 研究背景
- 主要工作

## 2 词表示模型

### 3 直接表示模型

## 4 研究方法与数据集特征

## 5 算法和代码

- 算法
- 代码

## 6 Future Work

# 研究方法 with 数据集特征

表: 公共网关抓取的信息数

| Site                    | Messages |
|-------------------------|----------|
| receivesmsonline.net    | 81313    |
| receive-sms-online.info | 69389    |
| receive-sms-now.com     | 63797    |
| hs3x.com                | 55499    |
| receivesmsonline.com    | 44640    |
| receivefreesms.com      | 37485    |
| receive-sms-online.com  | 27094    |
| e-receivesms.com        | 7107     |

# 研究方法 with 数据集特征

## ■ 使用 Scrapy 框架爬取公共网关

表: 公共网关抓取的信息数

| Site                    | Messages |
|-------------------------|----------|
| receivesmsonline.net    | 81313    |
| receive-sms-online.info | 69389    |
| receive-sms-now.com     | 63797    |
| hs3x.com                | 55499    |
| receivesmsonline.com    | 44640    |
| receivefreesms.com      | 37485    |
| receive-sms-online.com  | 27094    |
| e-receivesms.com        | 7107     |

# 研究方法 with 数据集特征

- 使用 Scrapy 框架爬取公共网关
- 收集 8 个公共短信网关在 14 个月的数据

表: 公共网关抓取的信息数

| Site                    | Messages |
|-------------------------|----------|
| receivesmsonline.net    | 81313    |
| receive-sms-online.info | 69389    |
| receive-sms-now.com     | 63797    |
| hs3x.com                | 55499    |
| receivesmsonline.com    | 44640    |
| receivefreesms.com      | 37485    |
| receive-sms-online.com  | 27094    |
| e-receivesms.com        | 7107     |



# 研究方法 with 数据集特征

- 使用 Scrapy 框架爬取公共网关
- 收集 8 个公共短信网关在 14 个月的数据
- 共抓取 386,327 条数据

表: 公共网关抓取的信息数

| Site                    | Messages |
|-------------------------|----------|
| receivesmsonline.net    | 81313    |
| receive-sms-online.info | 69389    |
| receive-sms-now.com     | 63797    |
| hs3x.com                | 55499    |
| receivesmsonline.com    | 44640    |
| receivefreesms.com      | 37485    |
| receive-sms-online.com  | 27094    |
| e-receivesms.com        | 7107     |

# 消息聚类分析

## 基本思路

- 使用编辑距离矩阵将类似的消息归于一张连通图中。
- 使用固定值替换感兴趣的消息，如代码、email 地址。
- 查找归一化距离小于阈值的消息，并确定聚类边界。

# 消息聚类分析

## 基本思路

- 使用编辑距离矩阵将类似的消息归于一张连通图中。
- 使用固定值替换感兴趣的消息，如代码、email 地址。
- 查找归一化距离小于阈值的消息，并确定聚类边界。

## 实现步骤

- 1 加载所有消息。
- 2 用固定的字符串替换数字、电子邮件和 URL 以预处理消息。
- 3 将预处理后的信息按字母排序。
- 4 通过使用编辑距离阈值 (0.9) 来确定聚类边界。
- 5 手动标记各个聚类，以确定服务提供者、消息类别等。

# 目录

- 1 引言
  - 研究背景
  - 主要工作
- 2 词表示模型
- 3 直接表示模型
- 4 研究方法 with 数据集特征
- 5 算法和代码**
  - 算法
  - 代码
- 6 Future Work



# 代码

## HOSVD 在 Python 的代码实现和分析:

```
1 def hosvd(X):
2     U = [None for _ in range(X.ndims())]
3     dims = X.ndims()
4     S = X
5     for d in range(dims):
6         C = base.unfold(X,d) #mode n分解
7         U1,S1,V1 = np.linalg.svd(C) #SVD分解
8         S = base.tensor_times_mat(S, U1.T,d) #迭代求解核心张量
9         U[d] = U1
10    core = S
11    return U,core #返回伴随矩阵和核心张量
```







# Thank you

# Thank you for listening!

# Q&A

Questions?