**事发原因抽取方案**

1. **数据准备**

**1、数据集：**

从案件文书中获取包含事发原因的固定位置的段落，给段落分词，得到词和词性，再添加其他特征和标注类型，构造crf++ 的训练文件和测试文件的标准数据格式。

**2、特征选择：**

**（1）词：**词语本身

**（2）词性：**POS

**（3）关键词：**

**a.描述原因开始的关键词**：因、由于、发现、于是、遂、在等。

**b.描述原因结束的关键词：**时、处等。

**c.涉及行驶的动词：**如起步、超速、超载、变道、转向、转弯、追尾、超车、停车、停放、倒车、顺行、通行、驶入、刹车、观察、瞭望、操作、坠入、掉入、落入、降低、追、撞、避让、碰撞、撞击、摔倒、翻滚、翻覆、侧翻、侧滑、滑行、刮擦、跌落、倒地、停、压、塌、压塌等。

**d.涉及交通法规的名词：**安全、文明、规范、规定、原则、措施、设施、标准、隐患、交通信号等。

**e.涉及事故发生状态的关键词：**不周、不够、不慎、不当、不力、不足、不及、疏于、大意、忽视、违反、失误、失控、视线、模糊等

**f.涉及案发环境的关键词：**施工、狭窄、冰雪、路面、湿滑、夜间、视线等

**3、****训练文件：**

文件每一行有四列，每列之间用空格分隔：第一列是数据本身（中文词语）；第二列是词性；第三列是；第四列是要预测的标注。

**（1）词性：**（按结巴分词的词性标准）

**（2）关键词：**

描述原因开始的关键词：Start

描述原因结束的关键词：End

涉及行驶的动词：Drive

涉及交通法规的名词：Rule

涉及事故发生状态的关键词：Status

涉及案发环境的关键词：Ement

其他词: N

**（3）标注类型：**

短语开始：B

短语中间：M

短语结尾：E

单独词作为短语：O

无关词语：N

**2、测试文件格式：**

  测试文件与训练文件格式一样。测试后的输出文件会在后面加一列标注结果。

**3. 模板文件**

模板文件中的每一行是一个模板。每个模板都是由%x[row,col]来指定输入数据中的一个token。row指定到当前token的行偏移，col指定列位置。

1. Unigram类型

#取第0列的前后和当前的特征：

U00:%x[-2,0]

U01:%x[-1,0]

U02:%x[0,0]

U03:%x[1,0]

U04:%x[2,0]

#取第0列的交叉组合特征

U05:%x[-2,0]/%x[-1,0]/%x[0,0]

U06:%x[-1,0]/%x[0,0]/%x[1,0]

U07:%x[0,0]/%x[1,0]/%x[2,0]

U08:%x[-1,0]/%x[0,0]

U09:%x[0,0]/%x[1,0]

#取第一列的前后和当前的特征:

U10:%x[-2,1]

U11:%x[-1,1]

U12:%x[0,1]

U13:%x[1,1]

U14:%x[2,1]

#取第一列的交叉组合特征:

U15:%x[-1,0]/%x[1,0]

U16:%x[-1,1]/%x[1,1]

U17:%x[-1,1]/%x[0,1]

U18:%x[0,1]/%x[1,1]

U19:%x[-2,1]/%x[-1,1]/%x[0,1]

U20:%x[-1,1]/%x[0,1]/%x[1,1]

U21:%x[0,1]/%x[1,1]/%x[2,1]

2.Bigram类型

考虑t-1时刻的标签。默认生成f(s',s)。

# Bigram

B

**二、开发准备**

**1、开发环境**

开发工具： CRF++

开发语言： python

**2、分析测试结果**

运行后生成结果文件，最后一列是预测的结果，可以和倒数第二列对比计算准确率和召回率。

精确率和召回率的计算方式：

正确率 = 正确划分词数/分割出来的词数

召回率 = 正确划分词数/文档总词数

F1计算方式：

F值 = 正确率 \* 召回率 \* 2 / (正确率 + 召回率)