Query纠错

# 错误的类型

拼音类型的错误包含同音别字、模糊音别字, 汉字方面包含错字、缺字错误

多种描述：划痕笔/补漆笔/修补笔/点漆笔，又或者苹果、apple

信息冗余: 冰箱温控器温度控制==冰箱温控器

属性检索：118冰箱、60寸液晶电视机4k高清智能60曲面

宽泛意图：超美吊灯、大容量冰箱

多字、少字、顺序交换、中英文混拼，中文拼音混拼

Non-word Error，指单词本身就是拼错的，比如将“happy”拼成“hbppy”，“hbppy”本身不是一个词。另外一种是Real-word Error，指单词虽拼写正确但是结合上下文语境确是错误的，比如“two eyes”写成“too eyes”，“too”在这里是明显错误的拼写。

**中文纠错以拼音为基础，编辑距离等其他方式为辅的策略。**

# 2、基于规则的纠错方法

线上预料全部的搜索词条和搜索日志统计搜索热词生成原始索引文件。

由原始索引文件生成拼音索引文件和编辑距离索引文件

拼音索引文件是生成关键词的全拼，简拼，尾部不完整拼音，拼音的模糊音变换到关键词的索引

编辑距离索引文件是生成关键词的每个位置的字删掉后余下的字串到关键词的索引

离线统计好高频错词的拼音🡪候选词表映射

# 3、基于统计的纠错方法

## 3.1 语言模型

先对查询词分词，然后将相邻词组合在一起进行拼音纠错活编辑距离纠错，对纠错得到的词进行分割，这样原始查询词的每个分词会得到若干候选词，然后使用维比特算法找到每个分词对应的候选词的最优组合，最优组合的标准是候选词的3-gram条件概率和最大

通常假定后面一个词只与前一个或者两个词有关即2-gram或者3-gram

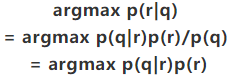
------------------------------

由库中的query生成3-gram检索词表和masked 3-gram

给定query，查询query的ngram是否存在于ngram检索词表，如果存在，则不用纠错，反之，mask掉该ngram最后一个字，查询该ngram是否存在于masked ngram检索词表里，如果不存在，则不做纠错，反之，则找到最多beam size个候选。

## 3.2 贝叶斯

如果把纠错当成一个黑盒，其输入可称为原串：用户query（q），输出可称为纠错串：纠错后的query（r）。纠错算法就是选出r使得q纠成r的概率最大，用贝叶斯公式表示就是：



所以纠错可以分解成：

p(q|r)：计算原串到纠错串的转移概率，常见方法有编辑距离，q和r的共点击概率，以及分别抽取q和r的相应特征，用模型预测p(q|r)等；

p(r)：衡量纠错串作为正常query的概率，比如语言模型，高频query，实体知识库等，即如果r的语言模型得分很高或者是个搜索次数比较多的query，其作为正确纠错串的概率就越大。

# 4、基于模型

## 4.1 序列标注

整个句子中，有错误的标注为1，没错误的标注为0，然后通过CRF之类的方法来进行预测，从而完成抽取。这个能很好的把控准确性，效果还是会比较好的，但问题在于这种标注样本，可能比常规的ner样本更难拿到。

## 4.2 基于上下文的query2vec

和word2vec doc2vec原理一样，利用语言模型的原理，上下文相关的query具有相似性。将共现的query看做句子。比如将用一个session的所有query 或者点击同一个doc的所有query看做句子。把query看做token,训练query向量。详细方法可参照相关论文

## 4.3 deepwalk

deepwal和4.2类似，区别在构造sentence上下文采用随机游走的方法。随机游走的原理：将query之间的关系建立成图。通过从一个点随机游走，建立起多条条路径，每条路径上的query组成一个句子。再使用上下文相关原理(4.2)训练query的embedding. 随机游走的优点就是关系具有传递性，和query共现不同，可以将间接关系的query建立联系。少量的数据经过游走能够产生够多的训练数据。例如session1:q1->q2 session2:q2->q3. 共现的方法无法直接建立q1->q3的关系。而随机游走能够很好的解决。

## 4.4 bert

将数据构造成q1 + SEP + q2 label的方式。训练样本的构造可按参照上面挖掘的方法，把挖到的数据做正样本，随机的query 对为负样本。 将q1， q2作为输入，得到sentence embedding，再根据label来计算loss，这样就可以学习到q1和q2的相似性。

## 4.5 基于翻译的思想

query作为源语言，rewrite作为目标语言，训练翻译模型，将query通过推理翻译达到rewrite目的

# 5、基于日志挖掘

基于用户的行为数据，挖掘query和query之间的关系。query点击日志和query session日志是比较容易想到的数据。对于点击同一文档的query建立query共现关系，以及基于同一个session下的query建立共现关系。 点击日志和session日志的数据各有优缺点，session的数据不过度依赖搜索结果。点击的数据普适性更强。可以结合使用

由query🡪点击的共现关系，可以建立query的相似关系

基于session的优点是不依赖现有的搜索结果。如果用户搜q1搜不到结果，可以继续搜q2，发现了想要的。如此，q1和q2就建立了联系。以后用户搜q1，就可以把q2的结果返回回来。和session数据相比，点击的数据缺点就是依赖现有的搜索返回的结果。如果搜索q1在现有搜索引擎下召不回任何结果，q1就没法和其他任何的query建立联系

# 6、基于知识体系的挖掘

就是近义词替换

# 7、其他

基于文本相似度，基于编辑距离（字或词），基于拼音

基于同义词

基于用户点击行为：query-doc协同过滤

基于用户同一个时间段的连续操作：基于session embedding，同一session中用户搜索词一般会相似，可以挖掘搜索词共现矩阵。可以对多个用户在同一session的搜索词序列组成user->query矩阵变化成query->user矩阵,计算不同的query的相似度，做相似query挖掘，或者做query聚类。

错误query识别：query正确则点击量高，query错误则搜不到内容或者内容很少，或者点击量低，通过这种方式可以对高频错误的query统计，线上用户输入的query如果存在这些词中，属于错误的query，需要纠正

共现query挖掘、混淆集挖掘

**新词发现**

# 8、系统设计

## 8.1 离线数据生成

搜索日志统计高频搜索词、拼音字典，同音词典、编辑距离词典、常见错误词候选词典、同音纠错库

## 8.2 字典存储与归一化

Trie树

## 8.3 线上检索

### 8.3.1 召回

基于词典和基于NLG

### 8.3.2 候选排序

最简单的方式就是使用语言模型的perplexity，即混淆度，这是用来一个句子他真的是句子的打分，一般而言这个正常无错的句子ppl就会比较小，有错的句子是ppl的比较大，可以用这个指标来衡量最佳的纠错结果是什么。



光一个ppl的评判是不够的，不仅仅是ppl的相对量，还有绝对量，还有就是pl虽然下降但是还是很高，还有和ppl无关的因素，如拼音的相似度、和原句的相似度等，因此可以先升级为机器学习，把前面提到的指标抽取为特征，通过简单的机器学习进行计算。

进一步地，同样可以使用平行样本，通过深度学习的方式来衡量是否需要纠正。

======================

先粗排：人工抽取的特征主要包括：频率比值（候选词频次越高分数越高）、编辑距离（编辑距离越小分数越高）、拼音jaccard距离（拼音越小分数越高）、Ngram统计语言模型分数差值（候选替换后语句越通顺分数越高）等。然后采用逻辑回归模型打分。

再精排：采用Xgboost模型对候选进行打分，同时在特征工程方面我们也进行了更精细化地处理。候选特征主要分为局部特征、全局特征两个方面。局部特征主要包括：分词变化、频次变化、形音变化、PMI互信息变化、Ngram语言模型分数变化以及一些其他的基础特征。

### 8.3.3 规则

规则。纠错系统中需要大量的规则，错误检测阶段衡量错误的阈值我们要用规则卡，排序阶段我们也需要一些提权降权保证最终我们需要的内容能排在前面，例如一些专有名词的保护，”电池“不能被改为”滇池“，”嬴政“不能被改为”行政“。这些规则看着简单，但是要想提出这些规则，必须对数据有足够的了解。

## 8.4 纠错效果评估

微观：查看搜索日志中无结果或结果少的纠错query以及点击模型中点击较少的纠错query等方式发现bad case

宏观：关注搜索效果评估系统中的MAP和MRR分数，使用AB test，查看使用纠错模块后或纠错算法升级后的带来的效果提升