智能出题

智能出题是指运用给定的文本段落，自行生成题目题干、正确选项以及错误干扰选项。智能出题主要包含三个过程：

1. 通过给定的文本段落，从中抽取重要信息概括该文本的主要思想，作为题目的题干
2. 运用主体识别分析出文本主要思想中包含的主体内容，例如：时间、地点、事件、人物、组织等，作为题目的正确选项
3. 运用词向量及聚类分析，在语料库中选择与正确选项相近的词语，作为题目的错误选项。

意群

1. 基于机器学习的自动文摘

自动文摘指自动从大量的文本信息中抽取重要的信息来代表文本的主要思想。运用自动文摘可以从给定文本中抽取该文本的主要思想内容，作为题目的题干。自动文摘按照摘要的产生方式可以分为两类，即生成式和抽取式。生成式产生的文摘可以包括原文中没有出现的词和词组，一般基于实体信息、信息融合以及压缩技术等。生成式文摘的缺点在于领域适应能力弱, 因此生成式文摘的普遍适用性受到了约束由于生成式对自然语言处理技术要求非常高，目前还处于起步阶段，产生的文摘离实用化还有相当的距离。

目前, 基于机器学习的方法是抽取式文摘的主流方法, 通过选取原文中的句子形成文摘，其方法分为两类: 基于有监督的学习方法和基于无监督的学习方法。有监督的学习方法视自动文摘为分类问题, 文本中的句子被分成两类——文摘句和非文摘句，并按照归属类别的程度选择文摘句子。无监督的学习方法通过文本的语义信息对句子打分，得分高的输出为文摘句。

* 1. 基于朴素贝叶斯的自动文摘

贝叶斯分类算法是一种基于概率的分类算法，即通过对象的先验概率，利用贝叶斯公式计算其后验概率，也就是该对象属于某一类的概率，通 常选择具有最大后验概率的类作为该对象的所属类。朴素贝叶斯是一种经典的贝叶斯算法，该算法假设对象的特征相互独立。

运用朴素贝叶斯分类器对文本进行自动文摘的基本思想是: 对每个句子s计算其成为文摘 S的概率，文摘包括 个特征 ，基于贝叶斯定理得到公式：



假设特征间相互独立，得到公式：



其中，是常数，和可以从训练语料库中估计，这样每个句子都有自己的概率得分，从高到低排列，高概率的句子输出为题干。

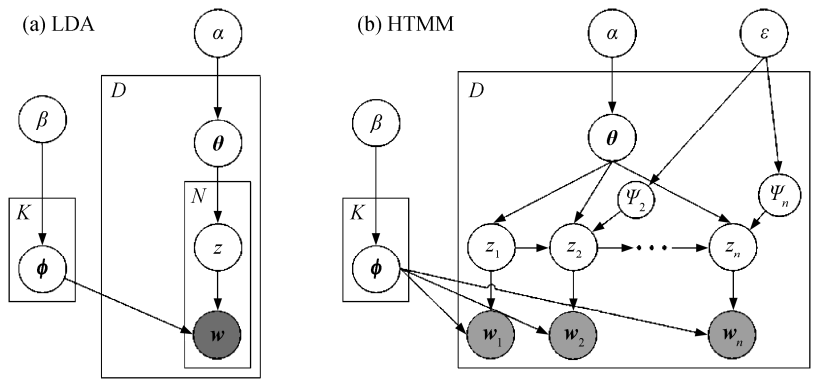
朴素贝叶斯的优点是它发源于古典数学理论，有着坚实的数学基础以及较为稳定的分类效果; 朴素贝叶斯模型所需估计的参数不多，对缺失数据不太敏感，算法也较为简单，与其他模型相比具有较小的误差率。其缺点是分类器假设特征间相互独立，把每个句子单独对待，忽略了句子间的联系，在特征个数比较多或特征间相关性较大时，选择朴素贝叶斯模型生成文摘的效率较低; 其次，朴素贝叶斯分类器是假设预知先验概率，预测出后验概率。不过，在实际应用中先验概率很难知道，通常的做法是取近似值，不过会给计算后验概率带来一定的负面影响。因此，对规模比较小或者特征间关联性较弱的语料库，利用朴素贝叶斯进行自动文摘的效果较好。

* 1. 基于主题模型的自动文摘

主题模型是一种有效的文档浅层语义表示模型。主题可以看成是词项的概率分布, 起源于隐性语义索引主题模型LSI。Hoffman在 LSI 基础上提出概率隐性语义索引 (pLSI), 被 认可为真正意义上的主题模型。LDA模型对 pLSI 进行更加完美的扩展。之后出现许多在 LDA 模型上的扩展工作, 大多数是引入其他相关信息, 例如STM (syntactic topic model)，选择主题时考虑句法信息, AT (author topic)考虑作者信息。

1.2.1 主题模型

LDA 在 pLSI 的基础上提出, 用服从 Dirichlet 分布的 K 维隐含随机变量表示文档的主题概率分布, 并且文档中词项的先验概率服从 Dirichlet 分布。LDA 是一个完整的文档生成模型, 通过主题和词项的概率分布模拟文档的产生过程。在文档的生成过程中, LDA 有两个假设: 主题独立假设和词袋假设。然而, HTMM在模型的构建上打破主题独立假设。HTMM生成文档过程中, 词由不同主题生成, 并且主题与主题之间服从二项式分布。



LDA与HTMM的结构示意图

1.2.2 基于主题模型的自动文摘

主题模型将主题表示为隐性变量, 仅文档内容可观测到, 最主要的参数为各主题下的词项概率分布()和各文档下的主题概率分布()。

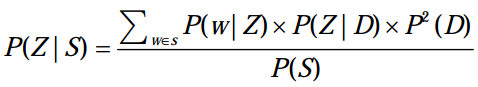
基于主题模型的抽取式文摘(机械文摘)生成, 其文摘句来自文档中的句子。利用主题模型中的与概率分布, 计算句子的主题概率分布和文档的主题概率分布。文摘句具有与文档相似的主题概率分布, 因而本文选取与文档具有高相似主题概率分布的句子作为文摘句。生成文摘可分为 3 个过程:

1) 训练主题模型参数, 得到与概率分布

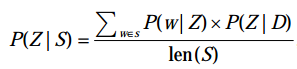
2) 计算句子主题分布及文档主题分布

3) 比较句子主题分布与文档主题分布, 从而抽取候选文摘句.

使用 HTMM 获取与率分布, 这与LDA 参数意义相同, 不同之处在于参数估计的迭代公式中, HTMM 以句子为单位考虑主题转移概率。然后计算文档的主题概率分布和文档中所有句子的主题概率分布。一方面, 主题模型的参数估计可以直接获得文档的主题概率分布 ; 另一方面, 句子的主题概率分布来自两部分: 文档-主题的概率分布和主题-词项的概率分布。利用公式计算句子*S*的主题概率分布:



其中*D*表示句子*S*所在的文档, 为主题*Z*下词项*w*的概率, 为文档*D*下主题 *Z*的概率。式中假设词与词相互独立，同时假设文档出现概率均等, 句子出现概率也均等, 并且消除长句优势, 将上式简化为



其中len(*S*)表示以词为单位的句长。

使用 KL 散度值计算两种概率分布的相似程度:



*P* 与 *Q* 为概率分布, KL 散度值越小, *P* 与 *Q* 概率分布之间的差异就越小。因此本文选取句子主题概率分布与文档主题概率分布 KL 散度值最小的句子集合作为候选题干, 即 *D*KL(*P*(*Z*|*S*)||*P*(*Z*|*D*))最小。

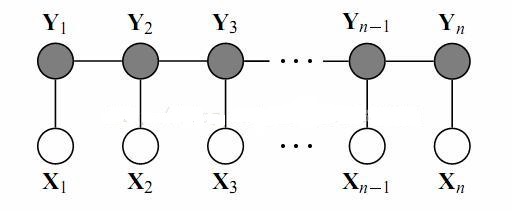
1. 基于条件随机场的命名实体识别

命名实体识别的主要功能是从文本中抽取出特定的事实信息。当自动文摘生成了文本段落的摘要内容后，利用信息抽取技术，抽取特定信息，作为试题的选项。

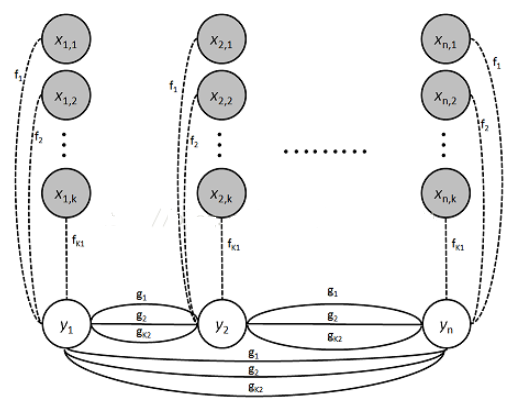
在智能出题中，运用命名实体识别技术识别文本中的特定实体，例如时间地点、组织名、人名等，作为题目选项。

2.1 命名实体识别

命名实体识别（NER）是在自然语言处理中的一个经典问题，其应用也极为广泛。其任务是从文本中识别出诸如人名、组织名、日期、时间、地点、特定的数字形式等内容。传统的公认比较好的处理算法是条件随机场（CRF），它是一种判别式概率模型，是随机场的一种，常用于标注或分析序列资料，如自然语言文字或是生物序列。条件随机场在命名实体识别中的应用是，给定一系列的特征去预测每个词的标签。如下图：



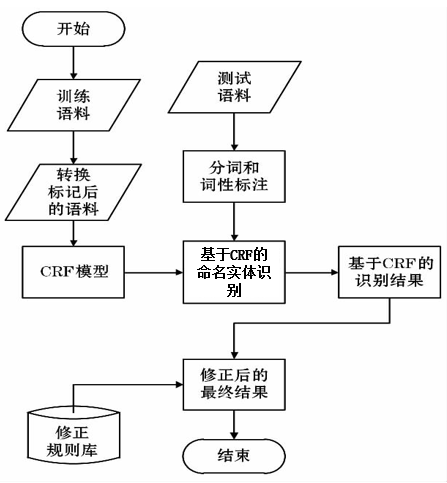
在上图中，X可以看做成一句话的每个单词对应的特征，Y可以看做成单词对应的标签。这里的标签就是对应场景下的人名、地名、事件名等等。特征X可以表示词性，如名词、动词等，然而单一的特征无法完成标签的学习，所以需要更多的特征。往往这些特征需要我们根据不同的场景去人工的抽取，比如抽取人名的特征我们往往可能看看单词的第一个字是不是百家姓等。所以，标签Y与特征X是一对多的关系。如下图：



2.2 基于CRF的命名实体识别流程

基于CRF的命名实体识别分三步进行：

1. 将试题库文本分为训练语料与不重叠测试语料，将训练语料进行标注转换后，利用CRF模型对转换后的语料进行训练，最终生成模型参数。
2. 利用分词技术对测试语料进行分词和词性标注，并利用上一步得到的CRF模型进行命名实体的识别。最终将词形和词性标注序列转换为本文定义的标注集序列。
3. 通过挖掘未识别实体样本的内部特点和上下文特征，设计了大量的人工规则，在上一步识别的基础上，进行二次识别。



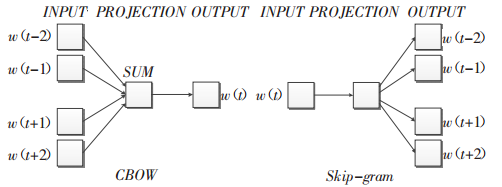
1. 基于神经网络算法和聚类的干扰选项生成

利用命名实体识别选择出了题目的选项后，需要智能生成与正确选项相近的错误干扰选项。由一些属性相近的词组成的词可以看成是单个词语到语意一般概念的映射。中文词语看成一系列独立词的“词袋模型”，这种模型将语言中词语之间的关系做了简化，仅仅考虑词语的统计特性；之后使用深度神经网络算法将词转化为n 维向量，在传统三层神经网络算法的基础上做了延伸，将网络从三层扩展到多层；最后用k-mean 算法计算对这些向量进行聚类，同一类中的词语具有类似的特征，选择与正确选项词语具有类似特征的词即可作为干扰选项。

3.1 词向量及word2vec

词向量具有良好的语义特性，是表示词语特征的常用方式。词向量每一维的值代表一个具有一定的语义和语法上解释的特征。所以，可以将词向量的每一维称为一个词语特征。词向量具有多种形式，一个词向量是一个稠密、低维的实值向量。词向量的每一维表示词语的一个潜在特征，该特征捕获了有用的句法和语义特性。可见，词向量将词语的不同句法和语义特征分布到它的每一个维度去表示。

Word2vec 层次化的 Log＿Bilinear 模型的一个实现，可以用来快速有效地训练词向量。Word2vec 包含了两种训练模型，分别是CBOW 和Skip＿gram，如下图所示：



可以看出，CBOW 和 Skip＿gram 模型均包含输入层、投影层和输出层。其中，CBOW 模型通过上下文来预测当前词，Skip＿gram 模型则通过当前词来预测其上下文。

在模型的训练过程中，梯度是训练参数更新的依据。为了获得梯度公式，需要构造出训练模型的目标函数。在 CBOW＋HS 框架的训练中，给定一个训练样本（*context*（*w*），*w*），则在输出层从根节点到词 *w* 的路径上，对于每一个非叶子节点均对应一个辅助向量 和一次二分类且左右两边各对应一个哈夫曼编码 。其中，j 表示从根节点到叶子节点 w 路径上非叶子节点的编号。对于，规定向左的分支对应 1，向右的分支对应 0。这样，通过若干二分类后，最终可到达叶子节点 w。可将非叶子节点向左的分支定义为负类，将向右的分支定义为正类。通过逻辑回归知识，可以得到一个二分类被分成正类的概率为

被分成负类的概率为

其中是context（w）中包含的词所对应词向量的累加和，而为非叶子节点对应的辅助向量。

对于 Skip＿gram＋HS，给定一个训练样本（*w*，*context*（*w*）），其中 *context*（*w*）包含 2c 个词。可以将通过 *w* 预测 *context*(w）的问题，即

*p*（*context*（*w*）｜*w*）转换成 *2c* 个通过 *w* 预测下一个词为 *u* 的问题 *p*（*u*｜*w*），其中 *u*∈*context*（*w*）。*p*（*u*｜*w*）可以利用 CBOW＋HS 框架的思路来解决，即将 *u* 视为叶子节点。不同的是，在 CBOW＋HS框架中，指的是 *context*（*w*）中所有词对应词向量的累加，而在Skip＿gram＋HS 中指的是 *w* 对应的词向量。Skip＿gram＋HS 框架的目标函数为

获取到词向量表之后， 我们就可以计算不同词向量直接的余弦距离， 作为不同词之间的“距离”。 在这之后，使用 K-means 聚类算法将近似的词聚集到一起。

3.2 K-means聚类算法

K-means 算法接受输入量k，然后将n 种数据对象划分为 k 组聚类以便使得所获得的聚类满足：同一聚类中的对象相似度较高；而不同聚类中的对象相似度较小。聚类相似度是利用各聚类中对象的均值所获得一个“中心对象”来进行计算的。算法基本步骤：

1) 从n 个数据对象任意选择k 个对象作为初始聚类中心;

2) 根据每个聚类对象的均值，计算每个对象与这些中心对象的距离；并根据最小距离重新对相应对象进行划分;

3) 重新计算每个（有变化）聚类的均值;

4) 计算标准测度函数，当满足一定条件，如函数收敛时， 则算法终止；如果条件不满足则回到步骤 2。

通过聚类算法，分析出与命名实体识别中识别出的题目正确选项属于同一类的词汇，作为题目正确选项的“同义词”，即题目干扰错误选项。