brat生成、T5生成、TextRank和Keypred都是用于生成文本摘要的方法，但它们在实现和原理上有一些区别：

1. **brat生成**：brat生成是一种基于规则的文本摘要方法。它使用一系列预定义的规则和模式，通过匹配关键词、短语或语法结构等方式来生成摘要。这种方法通常需要手动编写规则，并且对于不同的文本类型和任务需要进行定制化。brat生成的优点是灵活性高，可以根据需求进行精细的控制，但也容易受限于规则的准确性和覆盖范围。
2. **T5生成**：T5生成是基于预训练的语言模型，如T5（Text-To-Text Transfer Transformer）的生成能力。T5模型通过在大规模文本数据上进行训练，学习到语言的结构和语义信息，然后可以用于生成文本摘要。它可以自动学习文本的特征，并根据给定的输入生成相应的摘要。T5生成的优点是适用于不同类型的文本和任务，并且不需要手动编写规则。然而，T5生成可能需要更多的计算资源和时间来进行模型训练和生成。
3. **TextRank**：TextRank是一种基于图的摘要生成算法。它通过将文本中的句子构建成图的形式，并使用图中的节点和边的权重来评估句子的重要性。TextRank根据句子之间的相似性和连接关系来计算句子的权重，并选择具有最高权重的句子作为摘要的一部分。TextRank的优点是简单有效，并且不需要额外的训练数据。但是，它可能无法捕捉到文本的语义信息，并且对于较长的文本可能存在信息损失。

TextRank算法本身并不直接控制生成的摘要长度，它仅根据句子之间的相似度构建图，并使用PageRank算法对句子进行排序。因此，生成的摘要长度取决于你从排序后的句子中选择的句子数量。

1. **Keypred**：Keypred是一种基于关键词的文本摘要方法。它使用关键词提取算法来识别文本中的重要关键词，并根据关键词的重要性和出现频率来生成摘要。Keypred的优点是简单快速，并且可以根据关键词的覆盖范围和出现次数来控制生成的摘要。然而，Keypred可能会忽略一些语义信息和上下文关系，对于某些任务和文本类型可能不够准确。

总体而言，不同的生成方法在实现和效果上有所区别。选择适当的方法取决于具体的需求、数据和任务要求。一般来说，基于预训练

上面的代码使用了 **CountVectorizer** 将文本向量化，它主要基于词频进行向量表示，而不是使用预训练的词向量模型。

相比于使用预训练的词向量模型，**CountVectorizer** 的优点是简单易用，不需要下载额外的模型文件，并且可以自动处理词频信息。然而，它的缺点是无法捕捉到词语之间的语义关系，因为它只关注了词的出现频率。

使用预训练的词向量模型（如 Word2Vec 或 FastText）可以更好地捕捉词语之间的语义关系，因为这些模型是通过大规模文本数据训练得到的。预训练的词向量模型能够将相似语义的词语映射到向量空间中的相似位置。这使得在计算相似度时能够更准确地捕捉到词语之间的语义相似性。

因此，如果你关注词语之间的语义关系并希望在推荐中考虑到这些关系，使用预训练的词向量模型是更好的选择。而如果只是简单地基于词频进行向量表示，**CountVectorizer** 是一个简单有效的方法。

根据你的需求和数据特点，你可以根据实际情况选择适合的方法。

Genism pre-trained model 并不好

我测试了 fasttext-wiki-news-subwords-300

因为这些模型通常是在非常大的文本语料库上进行训练的，所以它们可以捕获很多复杂的词语关系和语义信息。

但是这些模型的问题是无法在计算机科学这个特定领域处理任务，并且有**词汇不匹配问题**：你的数据集可能包含一些预训练模型的词汇表中不存在的词汇，尤其是一些专有名词、短语或新词。

删除主语和动词会影响word2vec的哪方面判断

Word2Vec模型通过词语的上下文来理解和表示词语的含义。如果你从文本中删除主语和动词，那么将影响Word2Vec模型的以下几个方面：

1. **上下文理解**：动词和主语常常是句子的关键组成部分，它们帮助确定句子的行动和行动的主体。移除它们可能会改变或混淆模型对上下文的理解。
2. **语义关联**：Word2Vec模型的一个优点是它能够理解词语之间的关联和相似性。例如，模型可能会理解"cat"和"dog"有相似的上下文，因为它们都可能与"pet"、"animal"等词语一起出现。如果你删除主语和动词，这可能会改变这种关联和相似性的表示。
3. **句法信息**：Word2Vec模型能够捕捉一些句法信息，比如单词的词性、语态等。如果你删除主语和动词，模型可能无法有效地捕获这些信息。

请注意，这些影响在一定程度上取决于你的具体任务和应用。有些任务可能并不需要强大的上下文理解或语义关联。然而，如果你的任务依赖于理解句子的具体含义或者词语之间的关系，那么删除主语和动词可能会对模型的性能产生负面影响。

word2vec-google-news-300

推荐的效果可以接受，但是由于是通过上下文理解词语的含义，这个模型对于计算机领域的单词有些不理解，可能会误将某单词理解成其他单词，推荐出不符合用户输入的导师。

总体来说，Word2Vec因为通过上下文理解词义，他理解的词义可能与真实情况相反。

例如text analysis是一个东西，但是他可能接受到text analysis后，给你推荐介绍里包含其他单词+analysis的导师，导致出现歧义。

**Word2Vec**：这种方法将文本中的每个词转化为向量，而且这个向量可以捕捉到一些语义特性，比如相似的词会有相似的向量。Word2Vec实际上包含两种模型：CBOW（连续词袋模型）和Skip-gram模型。在CBOW模型中，我们根据上下文预测当前词，而在Skip-gram模型中，我们用当前词预测上下文。Word2Vec的主要限制是它只考虑单个词，不考虑文本的整体语义。

**Doc2Vec**：Doc2Vec（也称为段落向量或句子向量）可以被看作是Word2Vec的扩展，不仅可以将词转换为向量，也可以将较大的文本单元（如句子、段落或整个文档）转换为向量。这使得我们能够考虑到更多的上下文和词序，从而更好地捕捉文本的整体含义。Doc2Vec同样有两种模型：DBOW（分布式词袋）和DM（分布式内存）。

在许多任务中，例如文本分类或情感分析，Doc2Vec通常会比Word2Vec表现得更好，因为它可以考虑更大的上下文和词序信息。然而，哪种方法最好取决于具体的应用场景和数据。

使用Doc2Vec将导师信息当做一个整体向量化后在将用户输入使用**Doc2Vec计算余弦相似度。**

**设计了两种文本清理方法，第二种和第一种的区别是使用了spacy过滤了动词，连接词，主语，只保留了名词、形容词、副词和专有名词。**

1. **对CountVectorizer很有帮助，因为它只关心词语是否出现，以及出现的频率，而不考虑词语在其他文档中出现的频率。因此，它可能过度强调在所有文档中频繁出现的词语（例如“the”、“is”等常见词）。所以删除掉这些单词后，推荐算法可以更客观地进行推荐。**

**并且经过分析后发现导师信息和用户输入全部使用CountVectorizer能达到最好的效果。因为CountVectorizer更适合主题非常特定的文档。**

1. **TF-IDF（Term Frequency-Inverse Document Frequency）：这种方法不仅考虑了词频（TF），还考虑了逆文档频率（IDF）。这意味着如果一个词在一个特定的文档中频繁出现，但在其他文档中很少出现，那么这个词可能就很重要，因此会被赋予较高的权重。这种方法试图减轻一些常见词语（例如停用词）可能带来的影响，它们在所有文档中都很常见，因此可能不会提供太多有关特定文档的信息。**

**综上，TF-IDF在许多情况下可能会提供更好的结果，因为它能够减少常见词语的影响，并突出显示可能更具区分性的词语。但是，哪种方法更好可能取决于特定的应用场景和数据集。**

**所以TF-IDF提供了一种更纯净的方法，不用对原文档删除词语，得到了和CountVectorizer一样优秀的效果，综上所述TF-IDF向量化是本推荐算法最好的向量化方法。**