СЕМАНТИЧЕСКИЙ АНАЛИЗ ДОКУМЕНТОВ И ЕГО РЕАЛИЗАЦИЯ В MSSQLSERVER

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Барановский И.В.

КомличенкоВ. Н. – кандидат технических наук, доцент

Современный тематический поиск хорошо справляется с ситуацией, когда пользователь точно знает, что ищет и составляет правильный поисковый запрос. Тем не менее в поисковой выдаче всегда присутствует много лишних ссылок, в лучшем случае имеющих косвенное отношение к предмету поиска. Альтернативой классическому поиску может стать семантический, алгоритм которого построен так, что учитывается смысл слов в поисковом запросе.

В современных поисковых системах тексты автоматически индексируются по набору составляющих эти тексты слов. Такое представление текстов как простого набора слов имеет ряд очевидных недостатков:

1. Избыточность - в пословном индексе используются слова-синонимы, выражающие одни и те же понятия; слова текста считаются независимыми друг от друга, т. е. смысловаясоставляющаяслова;
2. Многозначность слов - поскольку многозначные слова могут иметь два или более понятия, выражающих различные значения многозначного слова, то маловероятно, что все они интересуют пользователя.

Поэтому предлагается использовать семантическую модель информации, которая лишена этих недостатков, за счет использования концептуального индексирования, т. е. индексирование не по словам, а по понятиям. При такой технологии все синонимы сведены к одному и тому же понятию, многозначные слова отнесены к разным понятиям; связи между понятиями и соответствующими словами описаны и могут быть использованы при анализе текста. [1].

В этом случае пользователь получает не только информацию документов, на которых встречалось упоминание данных слов, но и конкретную информацию, соответствующую сути поискового запроса.Например, если будет введет запрос о наблюдении Луны, то пользователь получит информацию об истории изучения и наблюдения Луны, о технике наблюдения, необходимом оборудовании.

Семантический поиск основан на существующей функции полнотекстового поиска в SQL Server, но дает новые возможности, выходящие за пределы поиска ключевых слов. Полнотекстовый поиск позволяет запрашивать слова в документе, а семантический поиск позволяет запрашивать значение документа. Среди новых возможностей - автоматическое извлечение тегов, обнаружение связанного содержимого и иерархическая навигация по схожему содержимому.

Для выявления аналогичных или похожих документов используется функция semanticsimilaritytable (Transact-SQL).

SEMANTICSIMILARITYTABLE возвращает таблицу документов семантически схожих с указанным документом. С помощью данной функции мы можем определить идентичные по смыслу документы и узнать их процент совпадения. Дляэтогосоздадимхранимуюпроцедуру с использованием semanticsimilaritytable , на входе который исходный документ, а на выходе документы, хранящиеся в базе данных, и оценка их совпадения с исходным документом.

```
PROCEDURE[dbo].[FindRelatedFileScores]
        @Titlenvarchar(255)
AS
BEGIN
        SETNOCOUNTON;
                DECLARE@DocumentIDhierarchyid
                SELECT@DocumentID=path_locator
                FROMMyDataFiles
                WHEREname=@Title
SELECTTOP (5)WP.nameAS[FileName],WP.stream_idasFileId,SST.scoreASScoreSum,TP.TitleASTopic
FROMsemanticsimilaritytable(MyDataFiles,*,@DocumentID)ASSST
JOINMyDataFilesASWP
ONSST.matched_document_key=WP.path_locator
JOINCourseProjectsMyDataFilesasCPFiles
ONWP.stream_id=CPFiles.MyDataFilesId
JOINCourseProjectasCP
ONCP.Id=CPFiles.CourseProjectId
JOINTopicasTP
ONTP.Id=CP.Topic
ORDERBYscoreDESC
END
```

В следующем примере извлекается топ 10 ключевых фраз из документа, указанного в переменной @DocumentID в столбцеDocumenttаблицы Production.Document базы данных AdventureWorks. Переменная

@DocumentID представляет значение изключевого столбца полнотекстового индекса. ФункцияSEMANTICKEYPHRASETABLE эффективноизвлекает результаты, используя поиск по индексу вместо сканирования таблицы. В этом примере предполагается, что столбец настроен на полный текст и семантическое индексирование.

```
SELECT TOP(10) KEYP_TBL.keyphrase
FROM SEMANTICKEYPHRASETABLE
   (
Production.Document,
Document,
@DocumentId
   ) ASKEYP_TBL
ORDER BYKEYP_TBL.scoreDESC;
```

В следующем примере извлекается топ 25 докуменов, содержащие ключевую фразу "Bracket" из колонки Document таблицы Production.Document образа базы данных AdventureWorks. В этом примере предполагается, что столбец настроен на полный текст и семантическое индексирование.

```
SELECT TOP(25)DOC_TBL.DocumentID, DOC_TBL.DocumentSummary
FROMProduction.DocumentASDOC_TBL
INNERJOINSEMANTICKEYPHRASETABLE
   (
Production.Document,
Document
   ) ASKEYP_TBL
ONDOC_TBL.DocumentID=KEYP_TBL.document_key
WHEREKEYP_TBL.keyphrase='Bracket'
ORDERBYKEYP_TBL.ScoreDESC;
```

Таким образом, был рассмотрен семантический поиск , его основные преимущества над тематическим поиском, а также его реализация в MSSQLServer.

Список использованных источников:

1. Семантическийпоиск [Электронныйресурс]. – Электронные данные. Режим доступа: http://masters.donntu.edu.ua/2011/fknt/bazhanova/library/statya.htm.
2. Настройка MS SQL Server 2012 [Электронный ресурс]. – Электронные данные. Режим доступа: http://svenaelterman.wordpress.com/2012/04/14/step-by-step-enabling-semantic-search-on-sql-server-2012/