ORACLE DATABASE 12c

Oracle Text 12.2 New Features

Adding more value to your textual assets.

ORACLE WHITE PAPER | APRIL 2017

ORACLE

Table of Contents

Introduction

Oracle Text is the full text indexing, retrieval and processing engine in Oracle Database 12c. Fully integrated into the database itself, it allows you to make full use of the textual assets within your organization without the need to install any extra software or learn any new interfaces.

This paper discusses new features available in Oracle Text with Oracle Database 12c Release 2 (version 12.2.0.1).

Sentiment Analysis

Sentiment Analysis - the process of computationally identifying and categorizing opinions expressed in a piece of text, especially in order to determine whether the writer's attitude towards a particular topic, product, etc. is positive, negative, or neutral.

WIKIPEDIA

Sentiment analysis is a way of indentifying positive and negative sentiments in a piece of text, essentially indentifying whether the writer thinks something is good or bad. Typically sentiment analysis identifies the overall tone of a complete item of text – for example to quickly identify how many feedback posts are positive or negative.

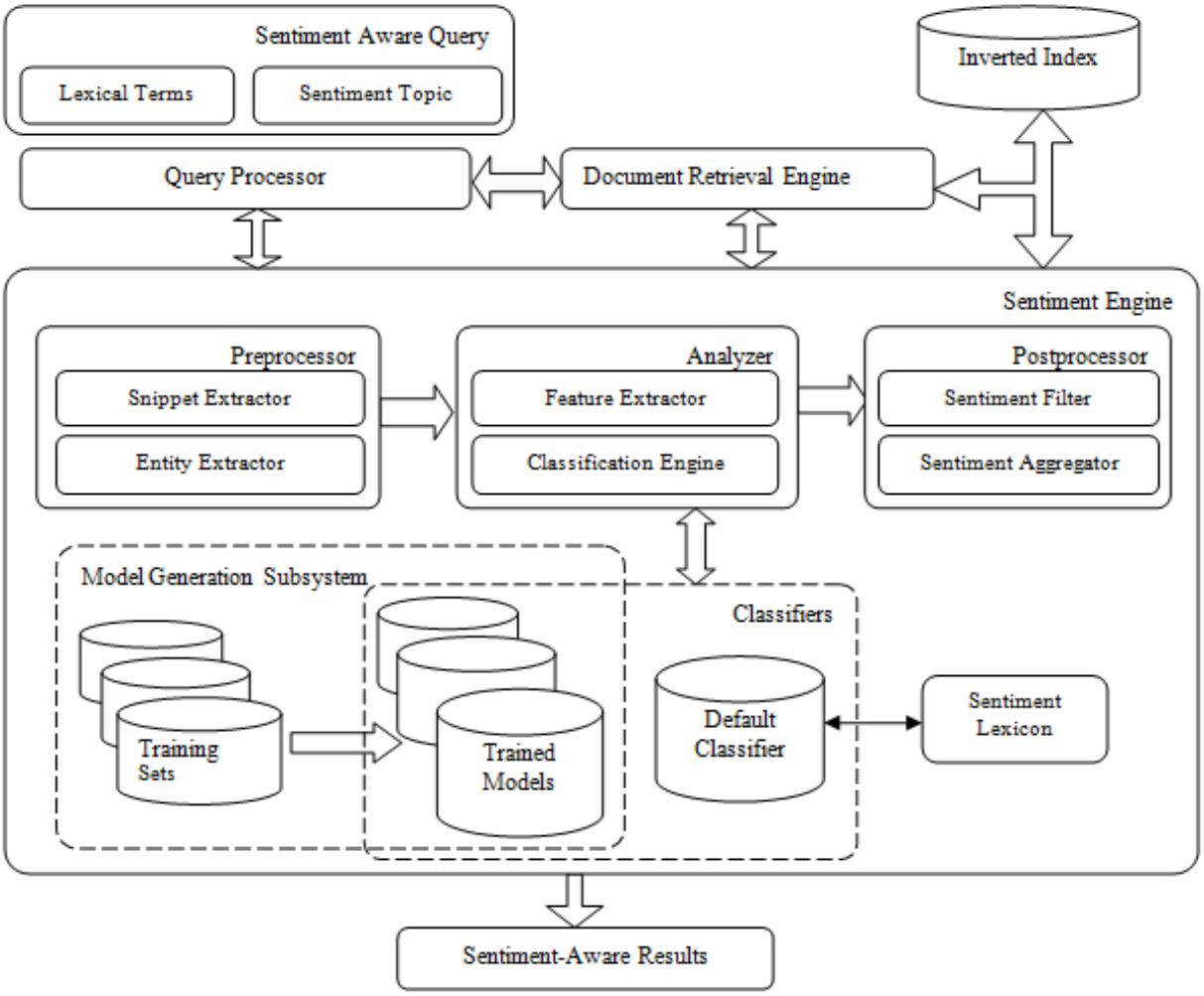
Oracle Text takes that a stage further by identifying positive or negative sentiments about a particular topic. So for example a review of a product might be generally positive, but might be negative about a particular topic of interest to the reader. A reviewer of a camera might be impressed overall, but not happy with one aspect, such as the lens quality.

"Out of all the reviews for this hotel, show me what they reviewers thought about cleanliness, and highlight good and bad comments about cleanliness in individual reviews."

Sentiment analysis based on individual aspects, not just "whole document"

Topic-specific sentiment analysis allows a reader to quickly identify the aggregate sentiment for such an aspect (are the reviews, overall, positive or negative about lens quality?), but also to zoom in on the actual sections of text which mention such aspects.

Sentiment Architecture



The core part of sentiment analysis is a classifier. A classifier can be either the *default classifier* which relies on a supplied *sentiment lexicon*, or it can be a custom classifier built using trained models. Using the default classifier is of course much easier, but provides lower accuracy than a trained model.

A trained model is based on a training set – a collection of documents, ideally similar to those which you will be analyzing, which are already marked with positive or negative sentiments. Such sentiments might be set by humans reviewing the documents, or it might be based on metadata such a “star” ratings which indicate whether the document is likely to be positive or negative in tone.

Reverse Index

A particular challenge for all text search engines is the retrieval of partial marches, commonly known as “wild card” searches. A wild card character matches an arbitrary string within a word, so for example in Oracle Text a search for “ant%” would match “ant”, “antelope” and “antidisestablishmentarianism”. Wild card usage can be divided into four main categories:

1. Trailing wild cards, such as “ant%” (as above)
2. Middle wild cards, such as “an%t” which would match “angst”, “anoint”
3. Leading wild cards, such as “%ant” which would match “rant”, “pendant”, etc.
4. Leading and trailing wild cards, such as “%ant%” which would match “ranting”, “plants” and many others.

Leading and middle wild cards are easily handled by having an index on the “word list” – the list of indexed tokens. Leading and trailing wild cards together (category 4) are dealt with in Oracle Text by means of the *SUBSTRING_INDEX* wordlist attribute.

That leaves us with the third category – leading wild cards. While these can be handled by the SUBSTRING_INDEX attribute, that attribute is expensive in terms of storage, index time and query processing. So a new attribute REVERSE_INDEX is included to handle this type of search in a more efficient manner. A functional index is used which reverses the order of tokens in the word list table before indexing them. This allows for fast lookups of words with leading wildcards, without requiring any extra storage in the word list table itself.

NEAR2 Operator

In pattern recognition and information retrieval binary classification, precision (also called positive predictive value) is the fraction of retrieved instances that are relevant, while recall (also known as sensitivity) is the fraction of relevant instances that are retrieved.

WIKIPEDIA

In text retrieval systems, there is always a trade-off between ***precision*** and ***recall***. Precision is a measure of the relevance (or accuracy) of your search results as a whole, whereas recall is measure of how many relevant documents are found. A very precise search will generally have high precision, whereas a much looser search will have high recall.

Let’s say I want to search for all documents containing “Oracle Cloud Database”. I can search for that exact string, which will give me very high precision. All the documents returned will contain exactly what I looked for. However, the recall will be low as I would miss a document that mentioned “the cloud database from Oracle” or “Oracle is a leader in database systems on the cloud”. So instead I could search for “oracle AND cloud AND database”. That gives me higher recall, but much lower precision – a document might mention “Oracle Database” at the start, and have an incidental reference to “cloud” many pages later.

The existing NEAR operator helps balance precision and recall. It looks for all of the terms, but insists they must be close to each other in the text. And the closer the terms appear to each other, the higher the document scores.

This works well, but has some important limitations. Firstly, all of the terms must be present and near to each other for the document to be considered a hit. If you search for seven terms, but only six are found, that is not considered a hit. Secondly, clusters of words near the end of the document are considered just as valid as clusters of words near the beginning.

NEAR2 avoids these limitations. It will search for a group of words close to each other, and score the result higher according to the number of matching words. It also has the option to score groups of words higher when they appear near the start of the document.

Nested NEAR

In addition to NEAR2, the normal NEAR operator has been enhanced. Previously only simple words or phrases were allowed inside a NEAR operator. Now, you can use OR conjunctions, so I can say: *find "*oracle*" near to either "*database*" or "*rdbms*".* And for power users, they can now nest one NEAR operator inside another. So I could say: *find "*oracle*" within three words of "*database*", and find "*cloud*" within twenty words of that combination.*

Collocates

In corpus linguistics, a collocation is a sequence of words or terms that co-occur more often than would be expected by chance.

WIKIPEDIA.

In a text retrieval application, it is often useful to know about terms which appear close to our search term in many of the documents of our result set. This might give us more information trends within the result set, it might give us ideas about alternate search terms to try, or it might allow us to narrow down the result set to just documents which mention a specific term. Such terms are known as *collocates*. The XML-based Result Set Interface has been enhanced to allow the extraction of such collocates. You can choose the radius of search (how close the terms should be to the actual search term) and the maximum number of terms to find. Collocate terms are ordered by relvency to the search, using a similar algorithm to that used to rank documents by relevance.

Conclusion

Oracle Text with Oracle Database 12c Release 2 continues to expand the capabilities for building leading-edge text retrieval applications using the Oracle Database and the SQL language.

ORACLE

Oracle Corporation, World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065, USA

Worldwide Inquiries
Phone: +1.650.506.7000
Fax: +1.650.506.7200

CONNECT WITH US


blogs.oracle.com/oracle


facebook.com/oracle


twitter.com/oracle


oracle.com

Integrated Cloud Applications & Platform Services

White Paper Title
April 2017
Author: Roger Ford
Contributing Author: Asha Makur


Oracle is committed to developing practices and products that help protect the environment