#جلسه ششم، پنجشنبه ۱۳۹۵/۸/۲۰

مدلهای داده‌ای

مدل برای ما چه اهمیتی دارد و ما چه نیازی به آن داریم؟ یا بهتر بگوییم مدل داده چیست؟

چگونگی ذخیره داده ها را مدل می‌گویند. چون بعدا با این داده ها کار داریم.

دو نوع مدل داده‌ای که ما در اینجا به بررسی آنها می‌پردازیم عبارتند از:

- مدل رابطه‌ای (Relational Model)
- مدل داده‌ای NOSQL

مدل رابطه‌ای

در مدل رابطه‌ای داده‌ها به شکل جدول ذخیره می‌شود؛ که شامل یک سری سطر و ستون هستند. در این مدل سطرها از ترتیب خاصی برخوردار نیستند و سطر تکراری هم نداریم. کلید و ... جزو مفاهیم اصلی جدول نیست فقط کار را با جدول آسانتر می‌کند. ولی در کار با جدول خود به خود سروکله کلید پیدا می‌شود.

جدول کجاها می تواند برای ذخیره داده‌ها مفید نباشد؟ عکس، فیلم، نقشه، متن و ... .

جدول دارای چه خصوصیتی است که برای متن مناسب نیست؟ طول هر سطر ثابت است، در حالیکه طول متن ثابت نیست. مثلا طول کامنتهای شبکه‌های اجتماعی متفاوت است.

جاهایی که بین سطرهای جداول ارتباطات زیادی داریم جدول پاسخگو نیست.

سوال: آیا نمیشه به هر ترتیبی که شده در جدول ذخیره کرد؟ بله. اما بعدا دردسرهای خاص خودش را دارد.

مدل داده‌ای NOSQL

مدل رابطه ای و جدول برای بسیاری از کاربردها از پایه‌ی اساسی ریاضی خوبی برخوردار است. اما امروزه سیستم‌ها داده‌هایشان را به گونه ای ذخیره می‌کنند که مدل رابطه‌ای برای ذخیره‌ی و کار روی آنها مناسب و یا کافی نیست. حتی اگر استفاده شود دردسر ساز است؛ بنابراین به مدلهای داده‌ای دیگری نیاز داریم.

گاهی به این صورت عنوان می‌شود که NOSQL یعنی رها کردن SQL؛ در حالیکه اصلا به این صورت نیست و NOSQL مخفف Not Only SQL و به معنای داشتن SQL در کنار دیگر مدلهاست. زیرا ۸۰ تا ۹۰درصد کارهای ما حداقل تا الان با جداول حل می‌شود.

پس تا اینجا متقاعد شدید که جداول برای ذخیره‌ی بعضی از مدلهای داده مناسب نیستند. و به مدلهای دیگری نیاز داریم. چهارتا از معروفترین مدلهای NOSQL که با آنها سروکار داریم بصورت زیر هستند:

- Key-value
- Document
- Column-family
- Graph

Key-value

همانطور که از اسم آن مشخص است همیشه اطلاعات در دو بخش(دو فیلد) ذخیره می‌شوند: Key و value. طول این فیلدها ثابت نیستند اما در کاربردها اصولا طول Key را ثابت درنظر می‌گیرند. در اینجا معنای Value "مقدار" است نه ارزش.

مثال: اطلاعات دانشجویان بصورت Key-value ذخیره شده است. (طبیعی است که برای ذخیره نام و فامیلی دانشجویان این روش مناسب نیست، فقط بعنوان مثال بیان شد.)

Key	**Value**
Name 1	Mehdi
Family	Esmaeili

نکته: در اینجا حرفی از نوع متغیر(entity type) نیست. مثلا در مدل رابطه‌ای شماره تلفن از نوع عددی تعیین می‌شود، اما در اینجا ما یک ساختار(structure) داریم که از دو ستون تشکیل شده است: Key و value. Valueها می توانند از انواع مختلفی باشند، مثل text، jpg، xml و ... .

key	value	
631	John Smith, 10.0.30.25, Good customer service	← text
365	10101101010110101011101011010101101010111010110	← image
198	<CustomerId>32195</CustomerId><Total>43.25</Total>	← XML

نکته: مدل رابطه ای خیلی انعطاف‌پذیر نیست، اما این چهار نوع کاملا flexible است(در نوع متغیر محدودیت نداریم). این یکی از محاسن این مدلهاست.

این مدل برای کجاها مناسب است:

- ✓ برای داده‌های غیرساخت‌یافته. {اغلب متون بدون ساختارند. مثل کامنتهای شبکه‌های اجتماعی. حال شما چطوره؟ چطوره حالت؟ حال you چطوره؟}
- ✓ خواندن و نوشتن high performance . {performance معیارهای متفاوتی دارد مثلا سرعت یک معیار است}
- ✓ آیا کلید را می توان تغییر داد؟

A key-value storage device is appropriate when:

- unstructured data storage is required
- high performance read/writes are required
- the value is fully identifiable via the key alone
- value is a standalone entity that is not dependent on other values
- values have a comparatively simple structure or are binary
- query patterns are simple, involving insert, select and delete operations only
- stored values are manipulated at the application layer

این مدل برای کجاها مناسب نیست:

- ✓ کاربردهایی که جستجو و پالایش دیتا روی value است مناسب نیست.
- ✓ جاهایی که رابطه داریم اغلب مناسب نیست.

A key-value storage device is inappropriate when:

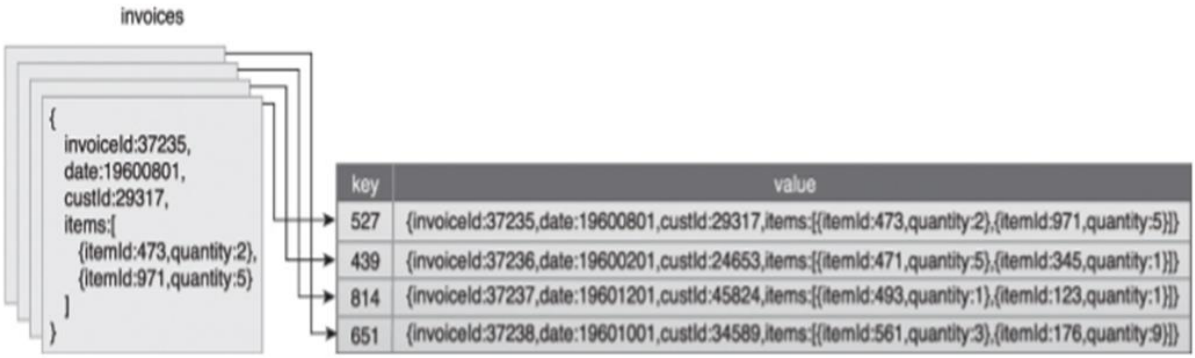
- applications require searching or filtering data using attributes of the stored value
- relationships exist between different key-value entries
- a group of keys' values need to be updated in a single transaction
- multiple keys require manipulation in a single operation
- schema consistency across different values is required
- update to individual attributes of the value is required

مثال: log fileها key-value هستند. Journalها در کامپیوتر یعنی log(ثبت وقایع). {key تاریخ، value عمل انجام شده است}

نمونه ابزارهای key-value: Riak ، Redis ، Amazon Dynamo DB .

Document based

یک نوع key-value است، که value آن document (اغلب هم text است) و ساخت یافته است. یکی از فایلهای که به این صورت ذخیره می‌شود فایلهای نوع JSON است. {یکی از فرمتهای خاص برای ذخیره داده‌هاست که اغلب search engineها برای crawl کردن از این فرمت استفاده می‌کنند.}



A depiction of JSON data stored in a document storage device

بیشتر بدانید:

JSON مخفف کلمه **J**ava **S**cript **O**bject **N**otation (نشانه گذاری شیء جاوا اسکریپت) بوده و یک استاندارد باز است که با ساختاری خوانا برای انسان و هم ماشین، می توان اطلاعات و داده‌های مختلف از جمله داده‌های یک دیتابیس را با استفاده از آن، بین عوامل مختلف مثلاً مرورگر کاربر و یک سایت منتقل کرد یا در فضای ذخیره سازی‌ای، آن را ذخیره نمود.

نوع رسانه‌ی اینترنتی رسمی آن، application/json و پسوند نام پرونده‌های جی‌سان `json` است.

JSON بیشتر برای سریالایز و انتقال ساختمان داده‌ها از طریق ارتباطی شبکه‌ای به‌کار گرفته می‌شود. بیشترین استفاده آن برای انتقال داده‌ها بین یک کارساز و یک برنامه وبی به‌عنوان جایگزینی برای XML است.

نمونه زیر یک شئ در JSON است که یک شخص را شرح می‌دهد. در این شئ نوع داده‌ای متنی برای نام و نام خانوادگی، نوع داده‌ای عددی برای سن، یک شئ برای ذخیره نشانی فرد و یک فهرست (یک آرایه) برای ذخیره شماره‌های تلفن شخص است:

```
{
    "firstName": "John",
    "lastName": "Smith",
    "age": 25,
    "address": {
        "streetAddress": "21 2nd Street",
        "city": "New York",
        "state": "NY",
        "postalCode": "10021"
    },
    "phoneNumber": [
        {
            "type": "home",
            "number": "212 555-1234"
        },
        {
            "type": "fax",
            "number": "646 555-4567"
        }
    ]
}
```

این مدل برای کجاها مناسب است:

A document storage device is appropriate when:

- storing semi-structured document-oriented data comprising flat or nested schema
- schema evolution is a requirement as the structure of the document is either unknown or is likely to change
- applications require a partial update of the aggregate stored as a document
- searches need to be performed on different fields of the documents
- storing domain objects, such as customers, in serialized object form
- query patterns involve insert, select, update and delete operations

این مدل برای کجاها مناسب نیست:

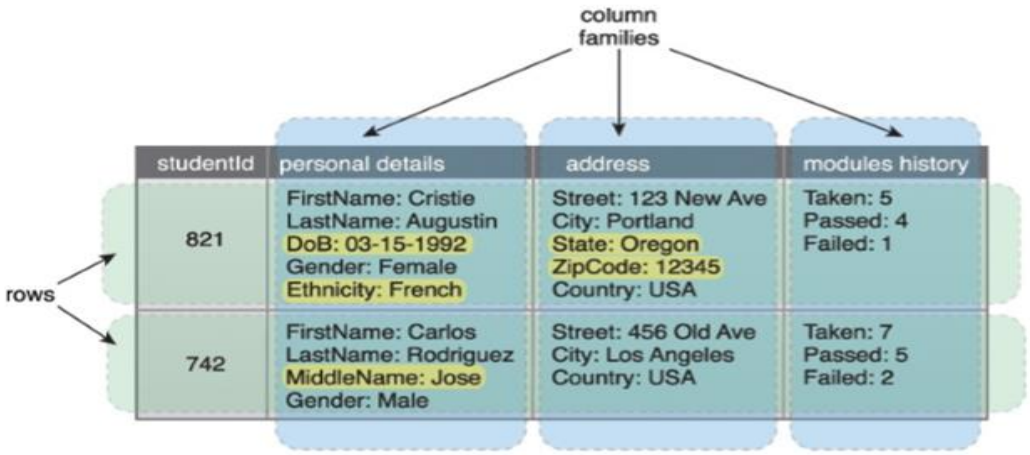
A document storage device is inappropriate when:

- multiple documents need to be updated as part of a single transaction
- performing operations that need joins between multiple documents or storing data that is normalized
- schema enforcement for achieving consistent query design is required as the document structure may change between successive query runs, which will require restructuring the query
- the stored value is not self-describing and does not have a reference to a schema
- binary data needs to be stored

نمونه ابزارهای Document based : Mongo DB ، Catch DB ، Terrastore .

Column-family

این ساختار شبیه جدول است با این تفاوت که در هر ستون آنچه که در یک سطر نگه داشته می‌شود با آنچه که در سطر دیگر نگه داشته می‌شود، متفاوت است. مثلا برای یک شخص نام و نام خانوادگی داریم، ولی برای شخص دیگر علاوه بر نام و نام خانوادگی، شماره تلفن و آدرس هم داریم.



وقتی با جدول کار می‌کنید طول فیلد ثابت است، اما در اینجا ثابت نیست. چون ممکن است ویژگی‌ای برای سطری باشد و برای سطری نباشد. و حتی در موقع لزوم می توان ویژگی‌ای را حذف یا اضافه کرد.

این مدل برای کجاها مناسب است:

A column-family storage device is appropriate when:

- realtime random read/write capability is needed and data being stored has some defined structure
- data represents a tabular structure, each row consists of a large number of columns and nested groups of interrelated data exist
- support for schema evolution is required as column families can be added or removed without any system downtime
- certain fields are mostly accessed together, and searches need to be performed using field values
- efficient use of storage is required when the data consists of sparsely populated rows since column-family databases only allocate storage space if a column exists for a row. If no column is present, no space is allocated.
- query patterns involve insert, select, update and delete operations

این مدل برای کجاها مناسب نیست:

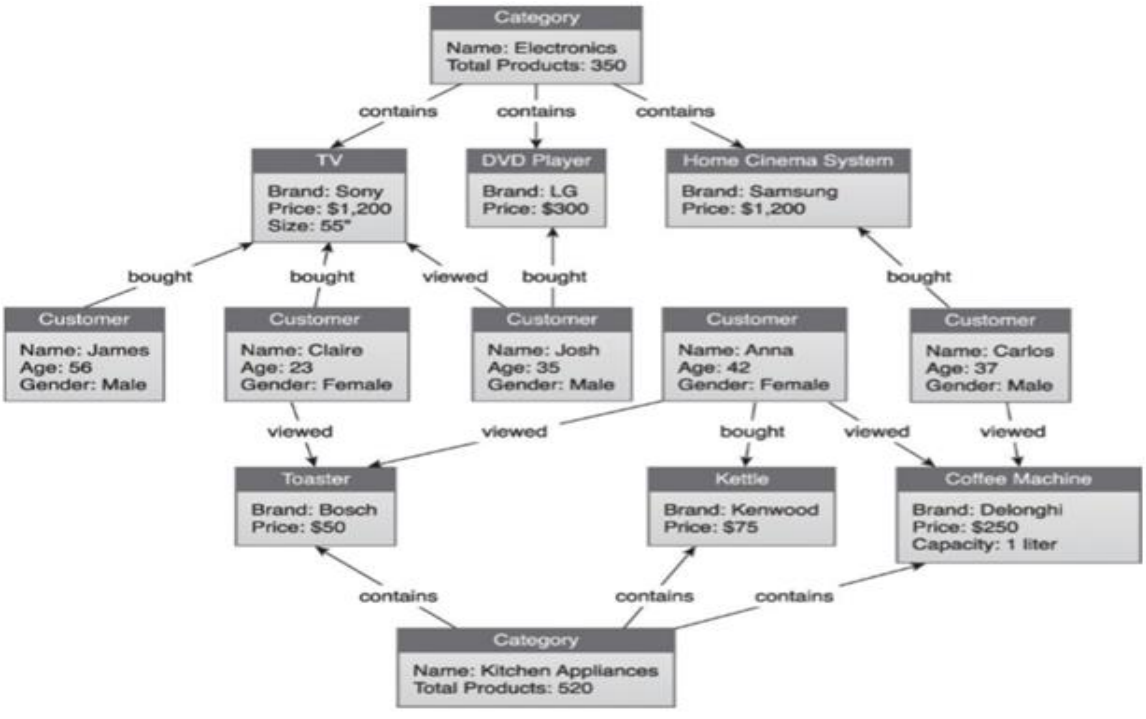
A column-family storage device is inappropriate when:

- relational data access is required; for example, joins
- ACID transactional support is required
- binary data needs to be stored
- SQL-compliant queries need to be executed
- query patterns are likely to change frequently because that could initiate a corresponding restructuring of how column-families are arranged

نمونه ابزارهای Column-family : Cassandra ، Hbase ، Amazon SimpleDB .

Graph-based

شکل زیر نمونه ای از مدل گراف است.



در ساختار گراف می توان ارتباطات متنوعی پیدا کرد. مثلا در شکل بالا TV با یک مشتری دارای ارتباط "فروخته شده(bought)"، با دیگری "دیده شده(view)" و با یکی ارتباط "شامل شدن(contains) جزو وسایل الکترونیکی" دارد. پس یک موجودیت(Entity) در مدل گراف می تواند چندین ارتباط متفاوت داشته باشد که مثال بارز استفاده از این مدل داده شبکه‌های اجتماعی است.

این مدل برای کجاها مناسب است:

A graph storage device is appropriate when:

- interconnected entities need to be stored
- querying entities based on the type of relationship with each other rather than the attributes of the entities
- finding groups of interconnected entities
- finding distances between entities in terms of the node traversal distance
- mining data with a view toward finding patterns

این مدل برای کجاها مناسب نیست:

A graph storage device is inappropriate when:

- updates are required to a large number of node attributes or edge attributes, as this involves searching for nodes or edges, which is a costly operation compared to performing node traversals
- entities have a large number of attributes or nested data—it is best to store lightweight entities in a graph storage device while storing the rest of the attribute data in a separate non-graph NoSQL storage device
- binary storage is required
- queries based on the selection of node/edge attributes dominate node traversal queries

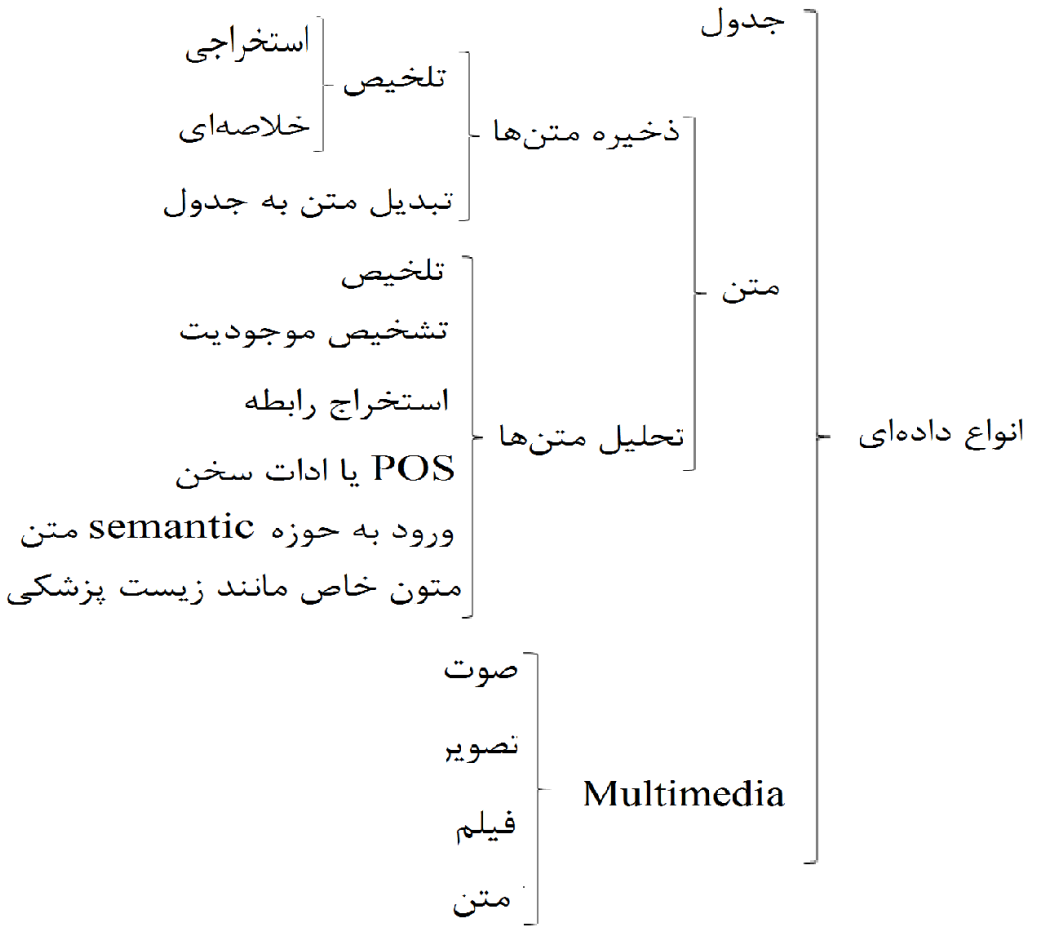
نمونه ابزارهای Graph-based : Neo4J ، Infinite Graph ، Orient DB .

نکته: این ۴ مدل تنها مدلهای NOSQL نیستند، ولی معروفترینشان هستند. مدلهایی هم وجود دارند که زیاد راجع به آنها صحبت نشده و تجاری سازی نشده‌اند.

هر جا که ما کار می‌کنیم با توجه به نوع داده‌ها و نوع گزارشهایی که مدیر میخواهد ابزار موردنظر را انتخاب می‌کنیم. البته اولویت با مدل رابطه ای است، چون امتحانش را پس داده است. این ابزارها خوبند، اما برای زمانیکه رابطه‌ای جواب نمیدهد. حتی بعضی از ابزارهای رابطه‌ای مانند SQL server و Oracle امکاناتی را اضافه کرده است که بتواند این نوع داده ها را ذخیره کند، که به اینها میگویند NewSQL. حتی اگر یک مقداری از کارتان با رابطه ای انجام میشود آن مقدار را با رابطه‌ای انجام دهید و بقیه را با مدلهای دیگر.

اما مواردی مثل شبکه های اجتماعی را به هیچ وجه نمی توان در جدول گنجاند.

#جلسه هفتم، پنجشنبه ۱۳۹۵/۹/۴



اغلب در بحث بیگ دیتا جداول زیاد کاربرد ندارند. ولی در عوض تولید داده های متنی در حال رشد هستند و ما خیلی راغب نیستیم که متنها را به جدول تبدیل کنیم (البته خیلی اوقات نمی توانیم آنها را به جدول تبدیل کنیم)، پس باید دنبال ابزارها و روشهایی باشیم که بتوان متن را ذخیره و تحلیل کرد. در این جلسه میخواهیم ابتدا در رابطه با متون و سپس گونه های دیگری از داده‌ها صحبت کنیم.

متن (text)

دوتا از نوع داده‌ای که جلسه قبل گفتیم برای ذخیره متن ها مناسب هستند. اولی document base و دومی key-value است(key-value کمتر استفاده می‌شود). ولی دو نوع گراف و cloumnal برای ذخیره متنها کمتر

استفاده می‌شود. مواردی که برای متن کاربرد دارد بحث ذخیره‌سازی و تحلیل است. بحث اول ما ذخیره کردن متنهاست.

ذخیره‌سازی متون

در ذخیره چندتا چالش داریم که باید آنها را حل کرد:

۱- یکی از نکاتی و کارهای خوبی که روی متنها انجام می‌شود تلخیص متن است(summarize). **(نکته:** تلخیص برای تحلیل متون هم استفاده می‌شود.) تلخیص یعنی به جای نگه داشتن کل متن خلاصه‌ای از متن را نگه داریم. تلخیص به دو صورت انجام میشود:

- استخراجی (extractive)

استخراجی یعنی به جای ذخیره ی کل متن جملات و پاراگرافهای مهم آن را تشخیص می‌دهد و آنها را ذخیره می‌کند. به هر حال طبق یک سری پارامترها سیستم با کمک یک سری ضرب و تقسیم این اهمیت را تشخیص می‌دهد. (extractive مثل هایلات کردن در مطالعات ما است.)

- خلاصه ای (abstractive)

در روش قبل عین متن برگردانده می‌شود، اما در روش abstractive خلاصه یا چکیده ای از متن برگردانده می‌شود. کامپیوتر متن را می‌خواند و مثل یک انسان چکیده آن را می‌نویسد. بخاطر همین شاید نتوان عین متن را در متن اصلی پیدا کرد اما مفهوم همان است.

نکته: روش اول ساده تر است.

پارامتر خیلی مهمی که در خلاصه سازی بررسی میشود "درصد خلاصه سازی" است و دقت داشته باشید اگر این درصد خیلی زیاد باشد شاید نتواند مفهوم اصلی متن اصلی را برگرداند. پس پارامترهای دیگر مثل accuracy ، F-mesure و ... نیز در نظر گرفته می‌شوند.

{این مسئله خیلی مرتبط با موضوع سرقت ادبی می‌باشد. یعنی می‌توان یک متن را به سیستم داد و خلاصه آن را بدست آورد و استفاده کرد.}

برای تحلیل هم می توان از این روش استفاده کرد. به این صورت که ابتدا متن را خلاصه کرده سپس روی آن تکنیکهای داده کاوی(این تکنیکها برای تحلیل به کار می‌روند) را انجام دهیم.

۲- برای ذخیره متنها راههای دیگری هم وجود دارد یکی از آنها هم تبدیل متن به جدول است.

جدولها فرمت شناخته شده و رایجی هستند و محصولات بازار هم خیلی برای آنها وجود دارند. پس زمانی که به داده‌ای برمیخوریم که جدولی نیست فورا به نظر ما میرسد که آن را به جدول تبدیل کنیم. حالا سوال اینجاست که چگونه می توان متن را به جدول تبدیل کرد.

مثلا قرآن، غزلیات حافظ یا حتی ایمیلهایمان را به جدول تبدیل کنیم.

نکته ای که وجود دارد اینست که در هر سطر جدول یک متن یا داکیومنت قرار می‌گیرد. این داکیومنت مثلا برای دیوان حافظ هر سطر می تواند اشاره کند به یک غزل، بیت، مصرع یا حتی یک کلمه.

اگر بخواهیم قرآن را به جدول تبدیل کنیم، می تواند هر سطر اشاره کند به یک آیه، سوره، صفحه، حزب یا جزء. پس بستگی به بخش بندی های مورد نیاز شما و کاری که شما می‌خواهید انجام دهید دارد.

- حالا در یک غزلیات حافظ که هر سطر اشاره میکند به یک غزل، ما نیازی به نوشتن کل غزل نداریم بلکه بعنوان نامی برای ستونها(فیلدها) از کلماتی که در کل دیوان به کار رفته استفاده میکنیم.

شماره غزل	عشق	دوست	را	...	
غزل ۱	۲	۰	۱۰		
غزل ۲					

برای مثال در جدول بالا می توان از سلولی که با عدد ۲ پر شده فهمید که در غزل۱ واژه عشق ۲بار تکرار شده است. این میشود Term frequency(فراوانی واژه) در سلولی که عدد ۰ نوشته شده یعنی کلمه دوست در این غزل وجود ندارد.

اما این جدول یه سری دردسر دارد. یکی از این چالشها زیاد بودن تعداد ستونهاست(تعداد ستونها به تعداد واژه‌های استفاده شده در دیوان است) و این چالش سبب اسپارس شدن جدول می‌شود(تعداد خانه‌های صفر زیاد می‌شود) چون قطعا همه‌ی واژه‌ها در همه‌ی غزلها بکار برده نشده‌اند.

- مسئله دوم اینست که آیا می‌توان گفت اگر تعداد تکرار واژه‌ای زیاد باشد اهمیت آن زیاد است؟ اتفاقا برعکس کلماتی که زیاد تکرار می‌شوند جزو کلمات بی‌اهمیت هستند(مانند "از" ، "به" ، "که" و ...). که می توان برای حل این مشکل اینگونه کلمات را از جدول حذف کرد. به این کلمات بی اهمیت میگویند stop word(ایست واژه). پس مرحله اول حذف این ایست واژه‌هاست(با این کار تعداد ستونها کم می‌شود).

نکته: کار کردن روی textهای زبانهای مختلف از لحاظ هایی مانند واژه‌های معمولی باهم مشابهند، اما از لحاظ قواعد دستوری و semantic با هم متفاوت هستند. مثل تشخیص همین ایست واژه‌ها؛ چون این واژه‌ها برمی‌گردند به قواعد دستوری آن زبان. مثلا در زبانی مانند فارسی فعل و فاعل با هم مطابقت دارند(از لحاظ شناسه)، یا در بعضی زبانها پسوند و پیشوند داریم، در بعضی زبانهای دیگر جنسیت وجود دارد و در بعضی وجود ندارد.

- می توان به جای انتخاب تک واژه برای نام فیلدها می توان از عبارات استفاده کرد. مثلا به جای اینکه برای عناوین فیلدها بنویسیم "امام" و "رضا" این دو واژه را بعنوان یک عبارت "امام رضا" بنویسیم، زیرا میدانیم در متن ما این دو کلمه همیشه با هم می‌آیند و باهم برای ما یک مؤلفه هستند. حتی امکان دارد بیش از دو کلمه را با هم جمع کنیم مثل "فرودگاه امام خمینی" .

اما ممکن است ما در متن عباراتی مانند "امام خمینی" ، "دبیرستان امام خمینی" ، "فرودگاه امام خمینی" داشته باشیم، بخاطر همین باید ابتدا متن را خوب تحلیل کرد سپس عبارات را مشخص کرد. هر چه بیشتر این تحلیلها انجام شود این تبدیل به جدول کمتر چیزی از دست داده است.

توجه: اگر شما برنامه ای بتوانید بنویسید که با دادن یک متن به آن جملات را جدا کند می توان آن را یک پروژه پایانی کرد و حتی از آن مقاله ISI درآورد.

- کار دیگری که می توان انجام داد اینست که می توان در ستونها به جای تعداد تکرار کلمات صفر و یک قرار داد (صفر=این کلمه در این غزل وجود ندارد، یک=وجود دارد).

نکته: با این کار قطعا حجم جدول ما کم میشود ولی درعوض تعداد تکرار را از دست می‌دهیم.

- به جای نگه داشتن تعداد تکرار می توان از tf/idf استفاده کرد. ایده اصلی این فرمول به این صورت است که برای هر داکیومنت یک ویژگی منحصربفرد را نشان می‌دهد.

ایده‌ی پنهان این فرمول اینست که یک ویژگی منحصربفرد برای هر سطر به ما می‌دهد تا بتوان به کمک آن دقیقا به سطر موردنظر رسید. برای عباراتی که در کل داکیومنت زیاد استفاده شود tf/idf به ما میگوید برای اینگونه کلمات اگر در تعداد اسناد زیادی قرار بگیرد من عدد بالایی(یا پایینی) نشان نمی‌دهم. مثلا می‌گوید در سوره‌ی فلان، فلان عبارت وجود دارد که در هیچ سوره‌ی دیگری نیست؛ پس این واژه را طوری قرار بده که برای این سوره چشمک بزند(اگر در سوره‌ای که می‌خوانی این واژه است، قطعا فلان سوره است).

اگر صدای چند استاد را برای شما ضبط کنند و بگویند تشخیص بدهید صدای کدام استاد است و لحن‌ها را هم یکی کرده باشند، از یک ویژگی خاص(مثل تیکه انداختن) در بیان هر شخص می‌توان به هویت آن فرد پی برد(این می‌شود tf/idf).

- در مورد تحلیل، مثلا برای قرآن یک ستون به نام مکی/مدنی اضافه کرده و نوع آن را از این جهت مشخص کنید. سپس برای این جدول یک درخت تصمیم(decision tree) رسم کنید و طبق آن تحلیل کنید که اگر مثلا این کلمه و آن کلمه و آن کلمه به این صورتها در سوره‌ای تکرار شد، آن سوره مکی است.

برای غزلیات حافظ می‌توان گفت اگر به این صورت و آن صورت بود از این نوع است. اگر از این قانون تبعیت نکرد می‌فهمیم این غزل از حافظ نیست.

مثلا برای association ruleها داریم، اگر در یک مصرع واژه‌ی x به کار برده شده بود، و در دو مصرع بعد واژه‌ی y استفاده شده بود، حتما در مصرع آخر واژه‌ی z استفاده شده است. به این صورت الگوها استخراج می‌شود.

برای ذخیره متنها می توان از تکنیک فشرده‌سازی نیز استفاده کرد(compress)

اما نکته دیگری که وجود دارد اهمیت نوع متن است. متن شما چیست؟ قرآن، متن کتاب، مجموعه‌ای از مقاله ها، غزلیات حافظ، ایمیلهایتان، کامنتهای شبکه‌های اجتماعی و ... ؟ با توجه به نوع هر متن نحوه‌ی ذخیره‌سازی آن نیز مشخص می‌شود. مثلا برای ذخیره مقالات و تبدیل به جدول می توان از column base استفاده کنید. و عنوان ستونها را به این صورت قرار دهید: عنوان، چکیده، کلمات کلیدی، مقدمه، متن، نتیجه‌گیری، منابع و ... . column base مناسب است زیرا می توان اندازه‌ی فیلدها را متغیر در نظر گرفت.{این می‌شود یک ایده}

تحلیل متون

۱- تلخیص

۲- **تشخیص موجودیت (Entity Recognition)**. اسامی مکان، اشخاص، زمانها و ... موجودیتهای ما هستند. در یک متن اسامی مکانها و اشخاص و ... را پیدا میکند که یکی از کاربردهای مهم و محاسن این مورد اینست که می‌توان فهمید متن در رابطه با چه موضوعی است.

۳- **استخراج رابطه(relation extraction)**. یکی دیگر از کارهایی که در تحلیل متن انجام می‌شود استخراج رابطه است. یعنی همان موجودیتهایی که استخراج کردید چه رابطه‌ای با هم دارند. از این طریق می توان فهمید که پدر شخص x چه کسی است. یا اینکه در سال ۱۹۹۰ مدیرعامل گوگل چه کسی بوده است. چون یک متن را crawl کرده و از آن فهمیده که مدیرعامل گوگل این شخص است.

توجه: به یاد داشته باشید که نباید خودتان را جای ماشین بگذارید آدم وقتی بخواند این روابط را بخوبی متوجه می‌شود، نحوه‌ی فهمیدن ماشین مهم است. برای سیستم‌های question/answering خیلی کاربرد دارد؛ همه ی اینها برای هک کردن و دزدی و پولشویی هم استفاده می‌شود.

مثلا می توان برای مهمانان(مهمانان کاری) یک شرکت متنهای اینترنت را relation extraction کرد و موارد موردعلاقه‌ی هر طرف را استنباط نموده تا رئیس شرکت بتواند در آن موارد با آن مهمان قرارداد ببندد.

۴- **ادات سخن یا POS (part of speech)**. تشخیص نقش واژه(فعل، فاعل، صفت و ...). و اما کاربرد این مورد کجاست؟ یکی از این کاربردها تصحیح متون است. مثلا وقتی صفت آوردی سیستم هم سریعا دنبال موصوف بگردد. یا وقتی چند حرف تایپ شد کلماتی که با این حروف شروع می‌شوند و در اینجا مناسبند را نشان دهد.

اگر تشخیص بدهیم که این واژه صفت است، پس می توان سریعا تشخیص داد که واژه بعد موصوف است. پس می تواند این دو کلمه را با هم در یک ستون قرار دهد. یا مضاف و مضاف الیه هم همینطور. و ...

اگر کسی با نقش واژه‌ها در هر زبانی آشنا باشد می تواند به درستی به آن زبان صحبت کند. پس نقش مهم است.

۵- **ورود به حوزه semantic(معنایی) متن**. آیا دوتا متن وقتی باهم دقیقا یکی هستند مشابه اند؟ این جمله الزاما درست نیست. زیرا مثلا دو کلمه‌ی "فرش" و "قالی" می توانند به جای هم به کار روند (شاید کلماتی مانند "دوست" و "یار" در کاربرد متفاوت باشند، اما در معنا مشابه‌اند). جالب اینجاست بعضی از افرادی که سرقت ادبی دارند از این حوزه استفاده می‌کنند. در یک مقاله عبارات مترادف را جایگزین و استفاده می‌کنند. یکی از استفاده‌های عمده دیگر نیز از این حوزه در موتورهای جستجو است، که بتواند این مترادفها را تشخیص دهد.

۶- **متون خاص مانند زیست پزشکی(biomedical)**. ما یک سری متون خاص داریم که خاص‌ترین آنها biomedical است. مثل DNA، RNA و ... . اینها متن هستند درست است صفت و مضاف و مضاف الیه نیستند اما متن هستند که می توان کارهای جالبی روی آنها انجام داد. مثلا می توان روی DNA یک انسان مطالعه کرد و به دنبال موارد مهم آن گشت. مثلا متوجه می‌شویم که این شخص در آینده به آلزایمر مبتلا خواهد شد و جلوگیری کنیم. این پردازش بسیار سخت و طولانی اما بسیار مفید است. مثلا از بزاق دهان انسان می توان دریافت که آیا در آینده به کدامیک از بیماریهای سرطان، آلزایمر و ... مبتلا خواهد شد.

درست است که حجم این داده‌ها زیاد نیست(مثلا یک رشته DNA یک سطر و نهایتا یک میلیون ستون است) اما به علت تنوع زیاد داده‌ای آن در دسته‌ی بیگ دیتا قرار می‌گیرد.

متن می تواند همراه با تصویر باشد، متن داخل تصویر باشد، web mining باشد و ... که می تواند موارد مختلف تحلیل را داشته باشد.

یکی از کارهایی که در متن انجام می‌شود سیستم‌های Question/Answering است. مثلا در قم مجتهدان زیادی هستند که مردم نیز برای پاسخ به سوالات شرعیشان به آنها مراجعه می‌کنند که بسیاری از این سوالات هم تکراری هستند. پس می توان با داشتن یک مخزن از سوالات و جوابها، میخواهیم یک سامانه طراحی کنیم که هر کس سوال پرسید با توجه به سوالات قبلی جوابگو باشد. البته امکان دارد آن سوال دقیقا در آنها نباشد. البته در اینجا ما خط قرمزهایی داریم(با اندکی تغییر در سوال جواب نیز متفاوت باشد) اما جاهایی هم هستند که این خط قرمزها را نداریم؛ مثل پاسخگویی به ایمیلهای شرکتهای بزرگ؛ مثلا مایکروسافت. حتی سیستم‌های question/answering تلفنی داریم. یعنی ماشین جواب سوالتان را از طریق تلفن میدهد. حتی sentiment analyze انجام می‌دهد، یعنی از لحن افراد متوجه می‌شود که مثلا این شخص عصبانی است پس آن را به یک اپراتور انسان وصل می‌کند. برای شرکتها و موسساتی که بسیار در بخش پشتیبانی کار دارند از این سیستم‌ها استفاده می‌شود.

Multimedia(چند رسانه‌ای)

که شامل همه چیز می‌شود؛ از جمله صوت، تصویر، فیلم، متن و ... . که البته روی هر کدام به تنهایی هم می توان کار کرد. اغلب تکی استفاده می‌شوند اما در کتابها نوشته شده اگر باهم استفاده شوند چیزهای به دردبخور بیشتری می توان از آنها دریافت کرد، اما تحلیل آن قطعا سخت تر می‌شود.

اولین راه‌حل تبدیل اینها به جدول است. که البته باید ابتدا با اصطلاحات مربوط به هر زمینه آشنا شد و سپس وارد بحث تحلیل آن شد. تعدادی از این اصطلاحات مفاهیم به شرح زیر است:

صوت: نت، سیاه، سفید، چنگ، چندلاچنگ، ریتم و ...

فیلم: فریم، سکانس و ...

تصویر: فریم، مولفه، PCA، کامپوننت، خلاصه و ...

متن: متن را چون خیلی کار کردیم برای ما ملموس است.

در مالتی مدیا یکی از کارهایی که بسیار خوب انجام می‌شود و گام اول برای تحلیل بیگ دیتاست زمانی که مالتی مدیا داریم، استخراج feature(ویژگی) است. یعنی یکی از کارهایی که برای کار کردن روی اینها انجام می‌شود

استخراج feature است. مثلا یک feature برای یک تصویر می تواند گل بودن، فرمت، حجم و ... ویژگیهای آن باشد. برای بدست آوردن این ویژگیها می توان سوالهایی را که در هنگام جستجوی عکس پرسیده می‌شود درنظر بگیریم. مثلا عکسهایی را پیدا کن که در آن حداقل یک گل رز هست. به این میگویند Multimedia Database.

ابتدا باید یک برنامه نوشت که featureها را بیرون کشیده سپس آنها را در جدول ذخیره کند؛ حال هر سوالی که پرسیده می‌شود با توجه به این دیتابیس جواب دهد. پس قسمت اعظم و مهم آن استخراج feature است. و اما اینکه چطور این برنامه این featureها را پیدا می‌کند جزو درس ما نیست و در حیطه‌ی درس Image processing و Multimedia database است.

یا مثلا وقتی در رابطه با فیلم صحبت می‌کنید سوال ما هم فیلمی است. مثلا در کدام سکانس فلان هنرپیشه بیشتر از سه ثانیه روبه تصویر است.

یا ما می توانیم یک سایت طراحی کنیم که از کاربری که دنبال یک موسیقی خاص می‌گردد بخواهد که یک قسمت از آن را بخواند یا با دهان بنوازد تا آن را در دیتابیس جستجو و پیدا کند. (سرچ صوتی)

اغلب در تحلیل مالتی مدیا توصیه می‌شود باهم انجام شود. مثلا صوت همراه با تصویر. اما در عمل به تنهایی تحلیل می‌شوند.