THE TRUTH ABOUT TRIPLESTORES

The Top 8 Things You Need to Know When Considering a Triplestore

TABLE OF CONTENTS

INTRODUCTION

Modern approaches to database management are becoming extremely popular, with no shortage of vendors discussing the latest alternatives to relational database engines. One type of graph database, triplestores, has grown rapidly of late. Triplestores have the ability to ingest diverse data, providing flexibility with respect to schema changes and mappings. They also allow for greater freedom, efficient handing of powerful queries and serving unforeseen information needs. Triplestores employ intelligent data management solutions which combine full text search with graph analytics and logical reasoning to produce deep, rich results. The cost for data integration, management and query definition is much lower than other approaches. These databases (also known as RDF, OWL or Graph Databases) are now widely used to manage unstructured and structured data in media & publishing, life sciences and financial services. Interestingly enough, the growth for this type of database is only just beginning.

In an Information Week article published in 2011, the question was posed “Will triplestores replace relational databases in three or five years?” The author wrote that there are usually two responses. The first response was “yes” because they are 100 times more flexible than relational databases and make it very easy to add new information.

The second response was “no” and the author indicated they would be used in conjunction with relational databases enabling smart integration of data using intelligent Metadata to describe the core information. Triplestores are the “smart brain” on top of legacy systems leveraging knowledge, rules and inferences to bring meaning to all of your data.

Whether you believe answer 1 or answer 2, the triplestore is becoming an essential part of database architecture today. One challenge buyers face is the lack of information available to make smarter buying decisions. In the same vein, business managers and decision makers do not have enough information about this important technology, preventing them from releasing applications that could transform any business.

Not all triplestores are created equally. In support of de-mystifying triplestores, this paper describes eight capabilities that buyers should investigate before licensing a triplestore. It concludes with sample use cases in the areas of search & discovery, personalized content recommendations and data visualization.

In many cases triplestores are used for content management (e.g. search or classification) in conjunction with text mining technology allowing for automated enrichment of content with metadata. Oftentimes, solutions are created that involve both technologies. To the extent that buyers need to consider related technology beyond the pure triplestore, this technology has been referenced below. Before we start, let’s answer the question, “Why are triplestores important?”

THE IMPORTANCE OF TRIPLESTORES

Triplestores store semantic facts in the form of subject - predicate - object using the Resource Description Framework. RDF is a standard model for data publishing and interchange on the Web. RDF has features that facilitate data merging even if the underlying schemas differ. RDFS and OWL are its schema languages; SPARQL is the query language, similar to SQL. This ecosystem of standards is defined by the W3C consortium within community processes that involve all major data management vendors (IBM, ORACLE, HP, Microsoft, to name just few). Unlike other proprietary NoSQL specifications, these standards are solid and have as much industry support as the basic specifications that make WWW work: HTTP and HTML.

RDF specifically supports the evolution of schemas over time without requiring all of the data transformations or reloading of data. A central concept is the Universal Resources Identifier (URI). These are globally unique identifiers, the most popular variant of which are the widely used URLs. All data elements (objects, entities, concepts, relationships, attributes) are identified with URIs, allowing data from different sources to merge without collisions. All data is represented in triples (also referred to as “statements”), which are simple enough to allow for easy, correct transformation of data from any other representation without the loss of data. At the same time, triples form interconnected data networks (graphs), which are sufficiently expressive and efficient to represent even the most complex data structures. To give a simple example, “This white paper is about triplestores” can be stored as an RDF statement inside a triplestore and contains a relationship between the subject of the sentence (white paper) and the object (triplestores). “Is about” is the predicate which indicates the type of relationship existing between the subject and the object.

In a very basic way, triples are an atomic form of intelligence that can be used to describe just about everything. Because this model is so simple, it allows structured, semi-structured and unstructured data to be mixed, exposed, and shared across different applications.

Triples contain relationships and are also related to one another. As a result, they are linked and represent a database version of a visual graph which is why they are often referred to as “graph databases”.

Triples can also represent connections between databases (structured data) and documents that contain free flowing, unstructured text. These connections are also extremely valuable. They can link entities from the database to the documents that mention them denoting relationships from which they were extracted. Some triplestores have specific support for such connections which allow data and text to form a single interconnected information space. That information space can be accessed through hybrid queries, combining the richness of the full-text search with the selectivity and precision of the databases. As we will see, when this important connection exists, organizations can keep all of their data synchronized. When a new triple is added or deleted, documents can be checked to see if new relationships can be mined and whether the old ones are still valid. When changes to a document are made, new triples can be created. Triplestores result in lowering the costs of content management systems. Unlike relational databases, triplestores provide schema flexibility and ease of data integration and querying.

Unlike document-oriented NoSQL databases and full-text search engines, triplestores allow for comprehensive structured queries. They are unique in their capability to interpret data through reasoning and to discover new facts and relationships. Triplestores do this based on semantic schemas (named ontologies) that provide formal definitions of the meaning of object and relationship types. Their capability to interpret data allows triplestores to provide rich, deep answers even when queries are unaware of the syntax in which data is ingested. Getting diverse data from multiple sources requires less work. Storing and maintaining tens of billions of facts on one commodity server means hardware costs will be lower. Inexpensive cloud options that are fully hosted and on demand provide a low cost alternative to enterprise solutions.

WHY TRIPLESTORES?

To net this out, here's why Triplestores are well liked and used:

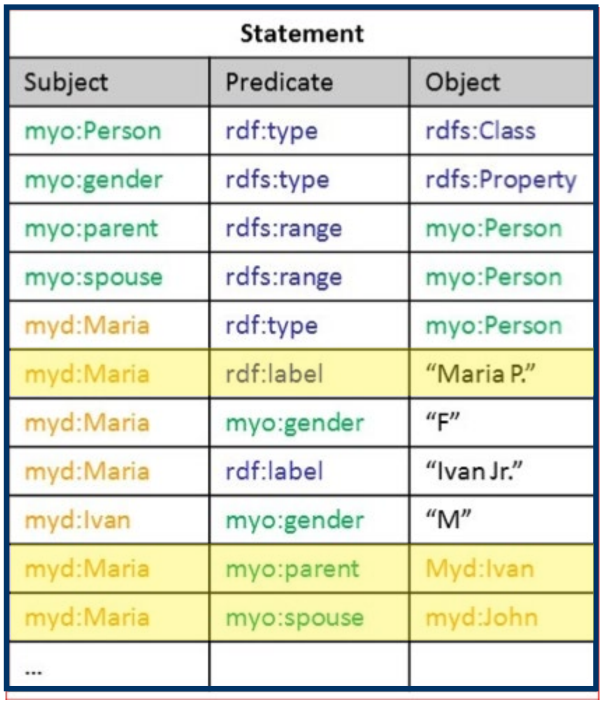
Schema Flexibility – Triplestores are flexible in many ways. They don't require complex schema development the way relational databases do. Therefore data can be easily loaded into triplestores. There are tools that can be used to translate your data into RDF format if needed. Because triples and URIs all exhibit the same formats, you can load massive amounts of triples about different subjects into the same triplestore. It's one very big bucket of knowledge that can continuously grow.

Easy to Query - Triplestores allow querying for general or even unspecified relationship types, releasing the application developers from the need to know the specifics of each data source and to change queries each time one of the sources changes. You can have as many languages of facts stored inside the same triplestore as you like. Some vendors have figured out ways to query both full text and the structured data at the same time. These hybrid queries provide even greater flexibility and additional insights.

Standards – While NoSQL databases are all usually different from one another, all triplestores share the same standards, allowing organizations to swap one triplestore for another. Moving data from one triplestore to another is also an easy task.

Provenance – Tracking where data came from has so many important use cases. Triplestores can maintain this level of detail. Users not only know the semantic facts but they know data sources, dates, levels of trust and other Metadata about the triples. Are you looking for facts about New York City provided in the early part of the last century from specific sources? That's not a big problem for a triplestore to answer.

Query Expressivity – When you query a relational database, you usually join data from different tables. When you set up the tables, you model how the data is joined. Is it a left join or a right join? This refers to the fields in the respective tables that are being joined. But the data inside a triplestore is in one table. It looks like this:



Statement		
Subject	Predicate	Object
myo:Person	rdf:type	rdfs:Class
myo:gender	rdfs:type	rdfs:Property
myo:parent	rdfs:range	myo:Person
myo:spouse	rdfs:range	myo:Person
myd:Maria	rdf:type	myo:Person
myd:Maria	rdf:label	“Maria P.”
myd:Maria	myo:gender	“F”
myd:Maria	rdf:label	“Ivan Jr.”
myd:Ivan	myo:gender	“M”
myd:Maria	myo:parent	Myd:Ivan
myd:Maria	myo:spouse	myd:John
...		

Figure 1: RDF Triples – Subject, Predicate, Object

Notice some of the triples: Maria is the spouse of John. Maria is a parent of Ivan. Also notice that classifications of the data are expressed as triples. Parent is a type of person. A spouse is a type of person. Joining all of this data is easy and universal across schema and data from any source.

Relationships in this table can be represented visually as well. This relationship (and others) are very important because now we can infer new facts that are implied in the data. For example, we can infer that Maria is the parent of Ivan and from there that Maria is a relative of Ivan.

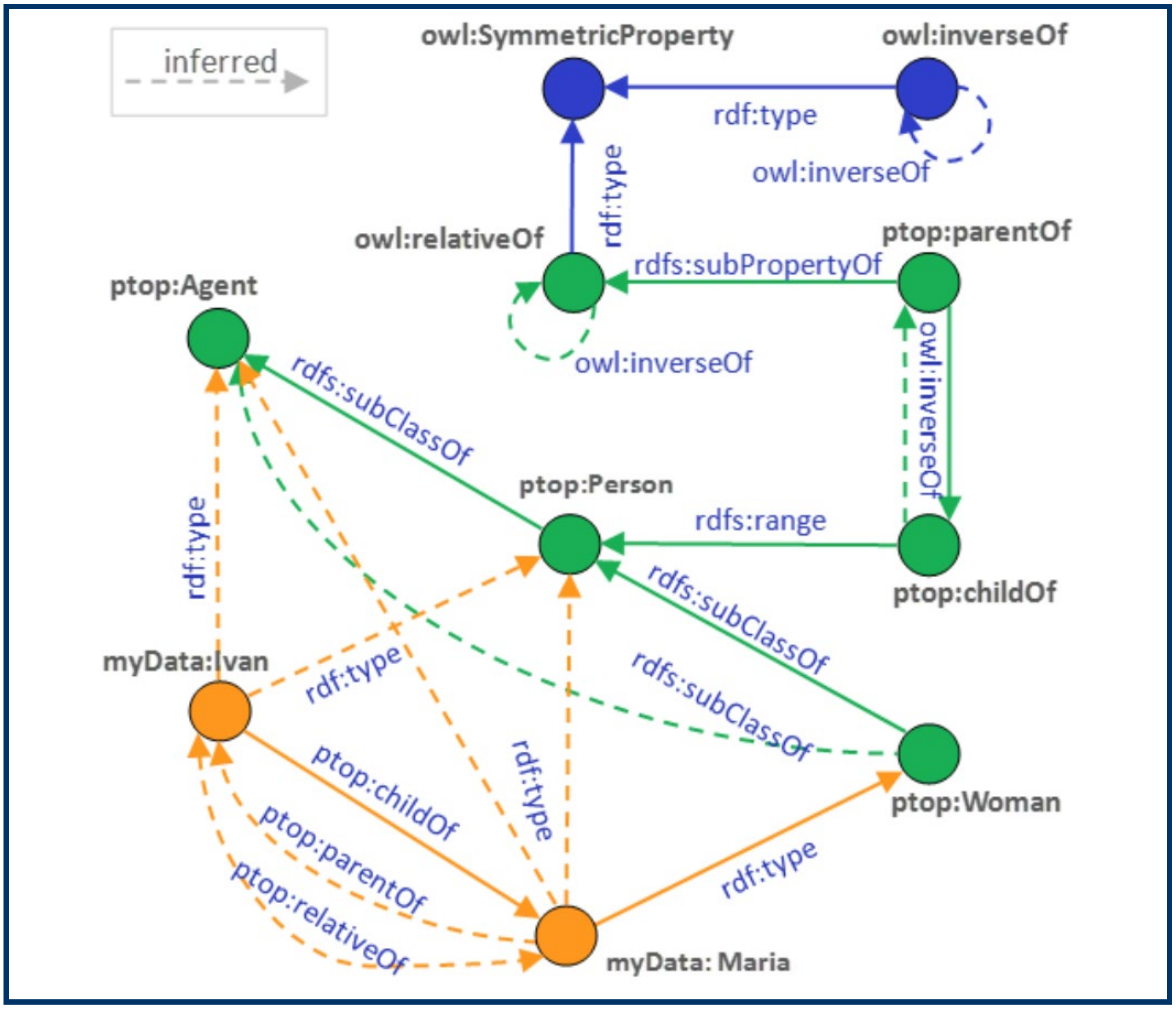


Figure 2: Visual Representation of Triples with RDF Types

In the diagram above, the actual triples, that were explicitly inserted in the database, are expressed with solid lines and direction. For example, Maria is a woman. Classes of data types are also represented – "woman" is a subclass of "person" and a person is a type of "agent." Other relationships between classes exist. For example, the term "parent" is a more specific relationship than "relative." Collectively, this represents a knowledge base including your data, your data classified and relationships between types of terms.

Since everything is interconnected (and oftentimes also linked to the originating documents from which they were derived), users can ask questions in a flexible way with relevant results returned. For example, if you ask to see "all the relatives of Maria," Ivan would be returned as an answer. The explicit relationship that was stored is "Ivan *is the child* of Maria." In this case, the triplestore also knows that there is an inverse relationship between parents and children. This relationship (and others) are very important because now we can infer new facts that are implied in the data. For example, we can infer that Maria is the parent of Ivan and from there that Maria is a relative of Ivan.

Having a complete knowledge base with entity classifications, data, provenance and linkages between facts and documents, creates a very powerful set of forces allowing you to be very expressive in your queries. Your analysis can be thorough and accurate.

Reasoning – As we just mentioned, some triplestores can infer new knowledge from existing facts. This is called inferencing. Let's suppose you have these facts: "Endocrinologists prescribe synthroid" and "Dr. Smith prescribes synthroid" but you don't have a have fact that says "Dr. Smith is an endocrinologist." And let's also suppose you query the triplestore with "show me all of the endocrinologists?" The inferencing engine can return "Dr. Smith" using a simple form of inference. Why is this important? You can create new facts from existing facts. Your queries run faster. Your results are more accurate. Not all triplestores support this capability and some apply different techniques to infer new semantic facts. We will explore this topic in more detail.

Costs – The best triplestores can hold 30-50 billion facts running on one commodity database server. They don't shard data. They don't need to. The cost of a triplestore is typically less than a relational database. The costs of the hardware and queries will also be less. Lower costs, time savings, and higher levels of accuracy are why many organizations have integrated triplestores into their enterprise architecture.

Knowledge Base – It's great to have billions of facts discoverable in your triplestore. But what if you also had classifications and further information for every entity referenced in those facts? What if you know that "Maria" is a person, "Leesburg" is the name of a city and "Bank of America" is a financial institution? Let's take this one step further. What if you also know that cities exist as parts of states in North America but in Europe, cities are part of countries? This knowledge base of rules and classifications can be expressed in the form of triples and stored in the database. You see some of this in the example above. There are thousands of public datasets in the Linked Open Data cloud that are published as RDFs and are already interconnected with one another. An example is GeoNames which provides extensive and accurate information for all geographic objects on Earth. This intelligence is rocket fuel for a triplestore. You can answer many more queries much faster. The results of the queries are highly relevant to the search.

Experts in this field will probably provide more reasons why triplestores are important. Now let's transition to another topic – How should you evaluate triplestores?

THE TOP 8 THINGS YOU SHOULD KNOW WHEN CONSIDERING A TRIPLESTORE

Since the very first incarnations of triplestores, customers across a variety of industries have repeatedly requested the following requirements leading to enterprise production deployments. As a buyer, when evaluating triplestore vendors, we believe you should look for these eight attributes.

1. INFERENCING

Generating new facts and proving facts using reasoning and a semantic knowledge base

As we mentioned, triplestores are designed to store semantic facts in an atomic form – subject, predicate and object. For example, “Flipper is a dolphin” can be one semantic fact. “A dolphin is a mammal” is another fact. Storing massive amounts of triples improves search. Organizations can analyze all of the facts. Customers receive highly relevant results. Web pages can be dynamically built based on search history, profiles and facts about unstructured text.

Moreover, hundreds of billions of these facts are available for free in the Linked Open Data world about music, places, subjects of interest and products. When applied correctly, semantic facts can enhance your knowledge management and data discovery applications.

Having all of these facts about people, places, organizations and events can be very powerful, but what if you could combine these facts to create new facts derived from the original sets? That’s called “inferencing” or “reasoning” and, when available in a triplestore, is one of the most powerful and important aspects of this type of database. Because we know that Flipper is a dolphin and because we have another fact that states a dolphin is a mammal, we can infer a new fact: Flipper is a mammal.

The applications of inferencing span industries and use cases. Knowing that two people are connected through a series of other factual relationships can be helpful in identifying networks for a variety of purposes – for example social networks, fraud networks or terrorist networks. In physician referrals and clinical trials research, the ability to infer a doctor’s specialty based on the drugs prescribed can help you find doctors that you require. When analyzing economic markets, the ability to infer trading price points for commodities using weather and regional data may provide you with a competitive advantage.

Does the lack of inferencing mean you are at a disadvantage? The answer is usually yes. Queries won’t return all of the facts, only the explicit ones. Analysts can’t always prove (or disprove) a fact of interest, slowing down decision making. Web site visitors and users seeking relevant, accurate results won’t always get a complete picture of the truth. This capability extends knowledge discovery far beyond the explicit data to which you have access today. Let’s look at an example of inferencing represented in a graph.

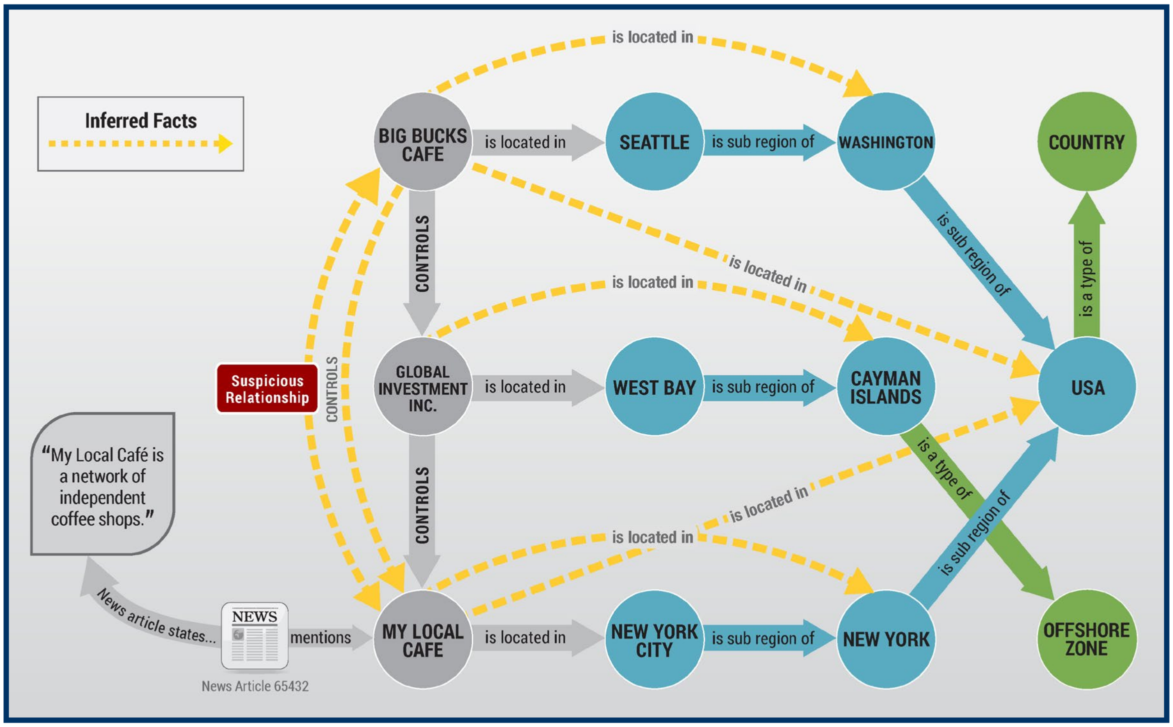


Figure 3: A visual example of inferencing

The graph above contains a series of triples which can be stored in the database. "Big Bucks Café is located in Seattle" and "Global Investment Inc. is located in West Bay" are two of these triples. Also notice that classifications of entities are in the database. For example, "Cayman Islands is a type of Offshore Zone" and the "USA is a type of Country."

Assume that an article appeared in a major newspaper that mentioned a company called "My Local Café" and in that news article My Local Café was referred to as "a network of *independent* coffee shops." If you had the database referenced above, you could ask questions to prove or disprove this reference. Based on the relationships that exist (the explicit semantic triples) you could reveal that My Local Café is actually controlled by Global Investment who is controlled by Big Bucks Café. Because of the transitive properties of graph databases, one can easily infer that Big Bucks Café is located in the state of Washington and that Big Bucks Café controls My Local Café. In other words, My Local Café is not *independently* owned at all. What's also very important is that you can infer a suspicious relationship between Big Bucks and My Local Café based on a number of other facts which are are true and reasoning.

This is the key: As new facts are added to the database, inferred facts can be created. It's these inferred facts that add extra explanatory power to queries and search results. Inferencing can be one of the most useful weapons in your knowledge discovery arsenal.

When evaluating inferencing engines, check to see if these core features exist:

Automation and Configuration - This process of inferring meaning should be iterative and require little configuration. In other words, as new facts are generated, they in turn can be used with inferencing rules to create more facts. Queries are answered accurately at high speeds because inferred facts have been created. Since a large part of data discovery is about having all of the facts, getting this up and running should be an automatic process, requiring very little configuration and therefore saving you time.

Complete inferencing and the impact on query resolution - There are two major types of inferencing strategies that should be supported: forward chaining and backward chaining. Forward chaining applies rules to known facts in support of generating new facts. This process iterates until no new facts can be created. One of the advantages of forward chaining is that it puts no burden on query evaluation. Backward chaining starts with a fact and then attempts to prove that fact is true. It does this by first checking to see if the fact is present and, if not, uses existing facts and rules to gain proof. Backward chaining can have serious repercussions on query performance. Some vendors have combined these approaches to optimize inferencing.

Standard compliance – There are several standard profiles,or languages, for reasoning. These include RDFS and the three profiles of OWL 2 – RL, DL and QL. Only RDFS OWL 2 RL and OWL 2 QL are appropriate for applications that need to deal with large volumes of data. Investigate which profile best fits the needs of your domain, data and application. Check whether the vendor provides full support for the standard profiles, particularly the one that best fits the needs of your application. Check whether the engine passes standard compliance tests and whether there is proof from independent evaluations of its reasoning support.

Life cycle inferencing - Reasoning support needs to apply to the entire lifecycle of data including loading, querying and updates. This is an essential capability required since massive volumes of data and updates need to take place in real time. For example, there are many instances when you need to remove an explicit statement in the triplestore. When this statement is removed, the triplestore needs to identify and retract any inferred statements that were created from the now removed explicit statement. This advanced form of inferencing management simply does not exist in most triplestores today. Consider the implications without this capability; the entire inferencing process would need to be rerun consuming valuable computational resources and time. Lack of support for retraction means that your query results will return misleading information.

Scale and Performance - When considering a triplestore, look for an efficient, compliant and scalable solution. The triplestore should be proven to have met all inference requirements on very large sets of triples, measured in at least tens of billions of triples. It must have a flexible rules engine. Are inferred facts derived at load time automatically? How does this impact query performance? Has the triplestore been proven to work against massive amounts of triples in production?

2. INTEGRATION WITH TEXT MINING PIPELINES

Continuously enrich content and the triplestore using text mining.

Most valuable information is stored in unstructured human-readable content. To make use of the information in a scalable, cost-effective way, it is necessary to extract the information from the content into a machine-readable format. This is text mining. The process is not unlike the refinement of oil. The raw resource, human-readable text, is put through a number of text-enrichment steps until it produces a number of refined products that have a calculable value to the organization. The refinement of text includes the identification of the parts of speech, the identified entities (e.g. people, places, organizations, etc.), the extraction of relationships between entities and algorithms for disambiguation. The output are the facts stored in the triplestore. These facts drive new information, improved discoverability and a finer-grained understanding of the content the organization produces.

This also enables organizations to make use of the raw text that exists out in the 'wilds' of the Internet.

How does text mining typically work? Figure 4 below shows a typical example of a text mining pipeline for a new article.

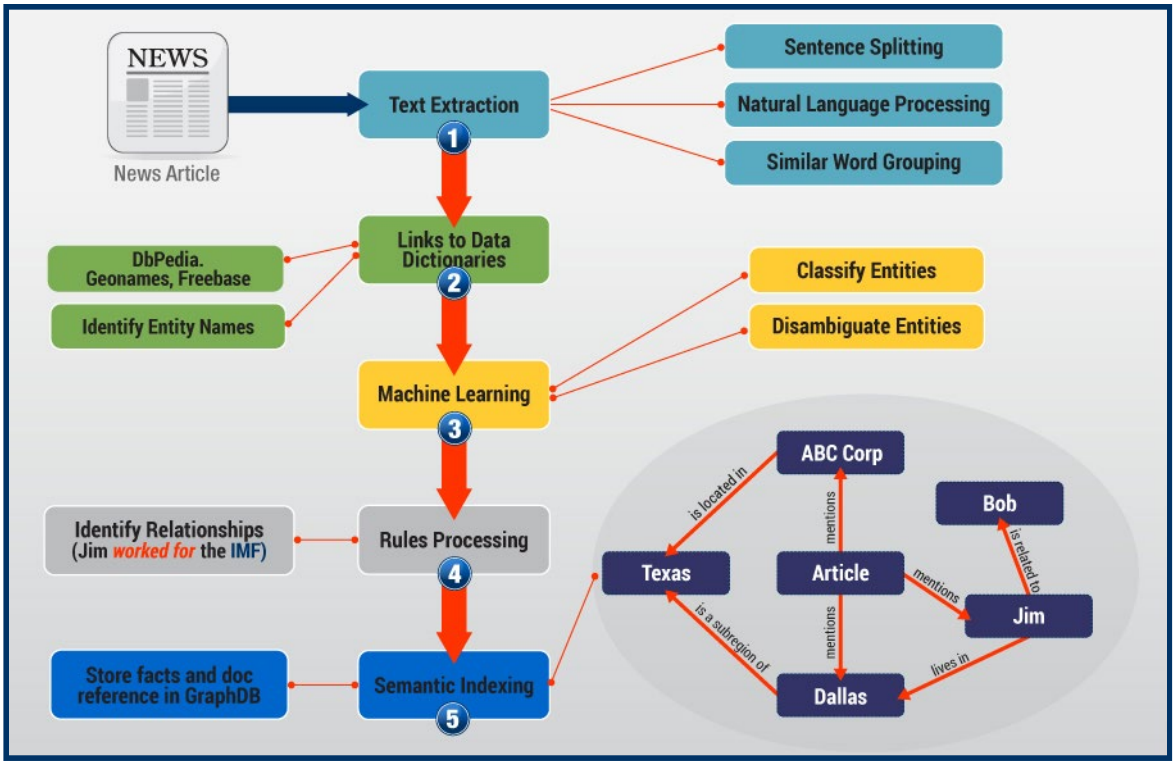


Figure 4: A Typical Text Mining Pipeline

1. Text is extracted from articles, documents or any form of unstructured data. Sentences are split and broken down into parts of speech. The different forms of the same words are grouped together.

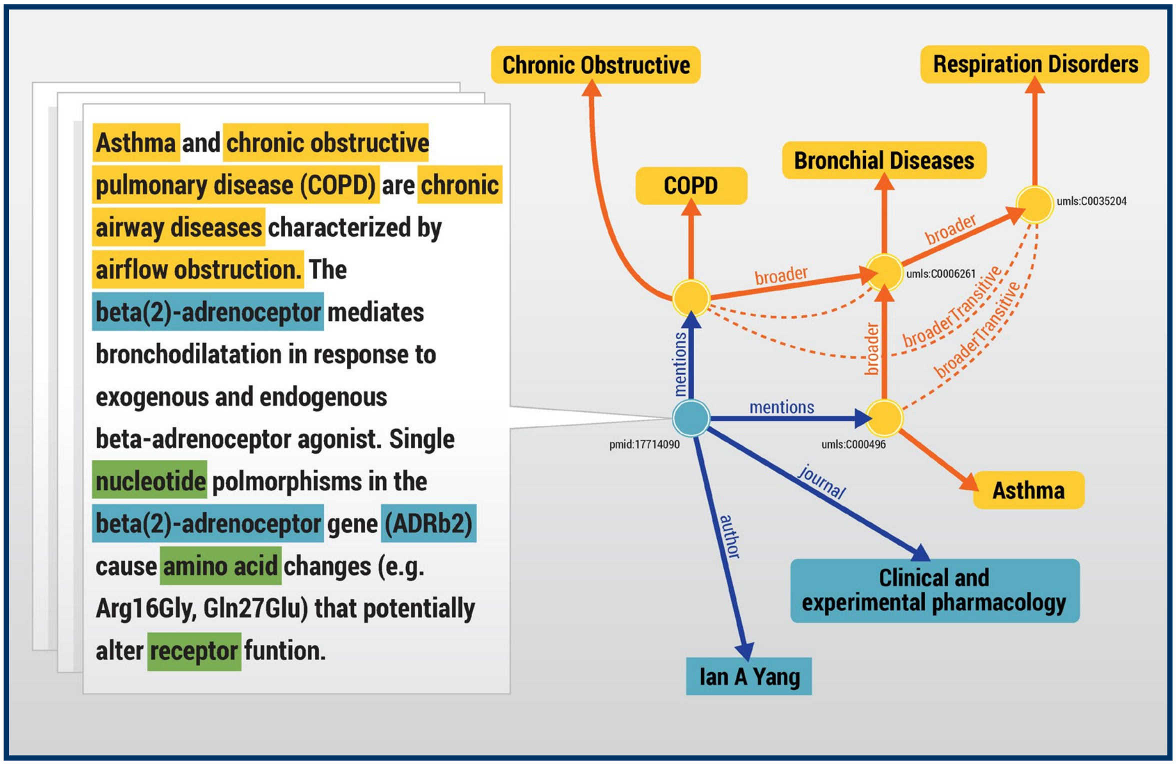
2. After sentences are split, the important concepts and entities (i.e. the proper nouns) are identified through dictionary word lists.

3. Machine learning algorithms classify and disambiguate the identified entities. When it come to classification, the algorithms, for example can classify "Lionel Messi" as a "soccer player". When it comes to disambiguation, the algorithms determine the correct entity being referenced. For example, the text might just mention the word "Paris," but, through the context the algorithm can figure out if it is Paris, France, Paris Hilton or Paris, Texas being referenced.

4. Relationships between the entities are also identified. Through rules processing, relationships can be identified between the subjects and objects. "Twitter acquires Vine," is a simple sentence that expresses an important relationship between two companies and text mining can identify this.

5. There's more. The facts and the original reference to the articles are indexed and stored with corresponding classifications and relationships in the triplestore. All of this becomes available for queries and can be returned in search results. In the example above, the article is indexed in the triplestore with explicit facts mentioned in the article. Collectively, this powerful blend of relationships, classifications, explicit & inferred facts and unstructured data allow organizations to understand and interrogate their content and data at a much finer grain of detail.

An industry where text mining can perform well using complex unstructured data is life sciences. Figure 5 shows a technical document that has been mined for classified entities and relationships. All of this data can be stored in the triplestore.



Asthma and chronic obstructive pulmonary disease (COPD) are chronic airway diseases characterized by airflow obstruction. The beta(2)-adrenoceptor mediates bronchodilatation in response to exogenous and endogenous beta-adrenoceptor agonist. Single nucleotide polmorphisms in the beta(2)-adrenoceptor gene (ADRb2) cause amino acid changes (e.g. Arg16Gly, Gln27Glu) that potentially alter receptor funtion.

Figure 5: Domain Specific Text Mining

The research article above mentions Asthma and COPD. Broader classifications of these diseases are also represented in the triplestore. This allows users to query using the broader terms with the specific disease states returned. The transitive properties of the database allow inferred relationships to be determined. For example, COPD is a type of respiratory disorder. Metadata from the article including the title and author are also retained in the triplestore. This complete knowledge base of organized relationships allows users to ask questions of the data that could not be answered without the existence of these important links.

By leveraging text mining with a semantic database, organizations create a virtuous cycle of enriched content, improved precision and recall and increased wealth of data within the knowledge base. The result is an increasingly automated system performing knowledge tasks that previously required the attention of expensive knowledge management experts. This expertise can now be redirected to adding value to content rather than spending time on maintenance and curation.

Without a triplestore, the output from text mining is left up to the user to determine where facts are stored. Without text mining, organizations are left trying to figure out how to extract meaning from text. Vendors that support both technologies equate to lower integration costs, reduced maintenance, increased savings and maximized ROI.

One of the most important considerations when evaluating this technology is the process of synchronizing information in real time. We know that documents are updated frequently. Edits take place by various writers, new content is added and much more. If a text mining pipeline existed which pumped updates into the triplestore, wouldn't that make everyone's life simpler? As the document changes, triples could be updated. The reverse is also true. If triplestores have pointers back to the originating documents from which they were derived, as new triples were added to the database, real time checks could be made with the documents to see if any new facts needed to be created. Could new inferences also be made? In essence, this is a way to synchronize your text and data, a monumental challenge that many organizations have today.

Anyone who has experimented with text mining or used it in production knows that processing language to create meaning is no small feat. Complex terms and real life meaning discerned from ground truth are sometimes hard to extract from text. As a result, organizations are focused on enhancing text mining to go from 90% accuracy to 100% accuracy. Content curators read articles and classify "hard to classify" entities. They store the results in a triplestore. Text mining algorithms can be trained and validated to achieve these higher levels.

The point here is that having a system and technology which allows for this type of enrichment, (typically referred to as Semantic Annotation), provides a big advantage in eliminating false positives and negatives. Real world experience can be applied to a set of documents, the results of which are used to improve the knowledge base and in so doing, provide a type of learning system that takes advantage of experiential knowledge.

Text mining plays a significant role in knowledge discovery today. Organizations can automatically extract facts about important events, entities and relationships using text mining. What's also essential is that this semantic pipeline is synchronized with your triplestore.

Other questions worth considering when evaluating text-mining solutions include:

- **Can the text mining pipeline be tuned for your specific domains?**
- **Can it make use of existing and new classification systems (ontologies, taxonomies, thesauri)?**
- **Does it use proprietary formats?**
- **How hard is it to make use of the output and integrate it within your technical stack?**

Here's the key: Organizations need to look for instant synchronizations between the text mining process and the triplestore. This should be offered out-of-the-box including an end-to-end solution that includes semantic authoring, enrichment, annotation and publishing. If you implement these pieces in an ad hoc way, years of resources will be consumed and the end results will suffer.

3. SCALABILITY

Will the solution handle your data needs today and in years to come?

Consider this scenario – all types of data are streaming into your company from a variety of sources and you need to process them right away. Updates and queries against your data are happening in real time. How do you reconcile the results? How do you ensure consistency of data?

Let's examine the problem in more depth. All of this content – structured and unstructured data - is enriched with metadata used to describe various aspects of the data. This data may be linked to other reference bases of knowledge. For example, the content could be linked to a taxonomy or ontology that includes classifications for your entities and concepts. If they can be used together, the streaming data, metadata and knowledge base provide a very rich set of information. They can be used in applications as diverse as semantic search, the dynamic assembly of web pages, creating new financial information products, personalized reader recommendations, improved learning systems and analysis of text based documents in the cloud. The possibilities are infinite.

Can your triplestore handle all of this complexity? Can it load billions of triples while providing status reports on the progress? Will it crack under the pressure of hundreds of queries per second while high volume updates are being written to the database and new inferred facts are being created?

When done correctly, the entire processing stream and resulting data allow organizations to efficiently produce highly relevant and enriched answers using powerful hybrid queries. The challenge lies in the handling of these interactions – inserts, deletes, updates, queries – simultaneously and at scale. Organizations need to find RDF triplestores that update and query the reference knowledge base and metadata at high speeds, especially when the queries aggregate data to produce a specific result.

Remember, updates are taking place all the time. The data is streaming in from any number of sources. As the database is updated, the resulting aggregation queries need to be immediately resolved. This produces database consistency allowing your organization to run operational applications in real time. Buyers interested in triplestores should look at vendors who have proven to run in production grade enterprise environments when complex processes are running simultaneously.

4. EXTENSIBILITY

Adding user defined logic through API Plug-Ins allowing for more powerful queries and flexibility

Triplestores should come with an API plug-in framework and a set of public classes and interfaces that allow developers to extend the triplestore in many useful ways. When you initialize your triplestore, the extensions should be automatically discovered during initialization. All of the query processing tasks should be delegated. By providing low-level access to the triplestore, the extensions are enabled to do their job efficiently.

Why is this so important? Users can embed custom logic in the triplestore engine and run SPARQL queries against it. This may include binding taxonomies and ontologies into the triplestore. Additionally, plug-ins should be able to use other plug-ins allowing them to take advantage of each other's functionality. For example, if you needed full-text search, you might plug-in Lucene and have the search results ranked using values provided by a second plug-in that facilitates query result scoring and ordering.

These extensions to your triplestore provide powerful processing capabilities. For example, plug-ins can:

- **participate in request processing**
- **interpret patterns in the data**
- **conduct processing (i.e. modify or filter results)**
- **process update operations for statement patterns containing specific predicates**
- **access other plug-ins**

Powerful triplestore APIs allow you to do more with your data. When buying a triplestore, investigate the breadth and depth of the vendors API and SPARQL support.

5. ENTERPRISE RESILIENCE

Proven to work in enterprise implementations

Most are well aware of the database management issues associated with big data. To handle massive volumes and extremely complex data of various types, triplestores must be enterprise-ready. Here's a small set of core features that you need to look for:

- **Supporting enterprise clusters organized as one or more master nodes managing worker nodes.**
- **Fail-over and load balancing between the worker nodes should be automatic and prevent performance hits.**
- **Multiple master nodes operating in stand-by mode that are used to load balance worker nodes ensure that clusters continue to perform even in the case of a master node failing.**
- **The cluster deployment can be modified when it's running, allowing worker nodes to be added during peak times, or released for maintenance and backup.**
- **Enterprise configurations guarantee high performance, an "always on" service capable of handling millions of query requests per hour.**

When evaluating triplestores, you should be focused on databases that have matured through the years, databases that are proven in mission critical installations across industries. Look for triplestores that are proven to handle ad hoc queries at scale against tens of billions of statements without ever going down. Enterprise resilient triplestores translate into higher levels of customer satisfaction, more productive users, optimal revenue levels and loyalty.

6. DATA INTEGRATION & IDENTITY RESOLUTION

Identifying the same entity across disparate data sources

With so many different databases and systems existing inside any single organization, how do companies integrate all of their data? How do they recognize that an entity in one database is the same entity in a completely separate database?

Resolving identities across disparate sources can be tricky. First, they need to be identified and then linked.

To do this effectively, you need two things. Earlier, we mentioned that through the use of text analysis, the same entity spelled differently can be recognized. Once this happens, the references to entities need to be stored correctly in the triplestore. The triplestore needs to support predicates that can declare two different Universal Resource Indicators (URIs) as one in the same. By doing this, you can align the same real-world entity used in different data sources. The most standard and powerful predicate used to establish mappings between multiple URIs of a single object is **owl:sameAs**. In turn, this allows you to very easily merge information from multiple sources including linked open data or proprietary sources. The ability to recognize entities across multiple sources holds great promise helping to manage your data more effectively and pinpointing connections in your data that may be masked by slightly different entity references. Merging this information produces more accurate results, a clearer picture of how entities are related to one another and the ability to improve the speed with which your organization operates.

When identities are resolved and mapped this way, however, exponential growth of the data in the triplestore can take place because a single fact may have many variations. Suppose there is a triple stating that New York City is part of USA. Now imagine there are 10 different identifiers for NYC and another 15 for USA – this means there are 150 different variants of this triple. To handle this, the triplestore needs to be optimized, allowing you to efficiently and correctly deal with this phenomena. This allows you to retrieve such facts using any of the equivalent URIs, but without massive penalty from explicitly storing all the variants. This optimization is essential since the matching process cannot get in the way of currently running queries, database updates or inferences. As your triplestore increases in volume, indices will also increase. The recommendation to buyers is to first look for this functionality and then see how it has been optimized. Triplestore vendors should also offer tools to transform your data into RDF format allowing for easy integration and updating of the graph database.

7. SEMANTICS IN THE CLOUD

Hosted services to perform semantic analysis at lower costs

Small, medium and large organizations and government entities continue to show interest in SaaS and cloud-based solutions. Some semantic technology vendors have deployed cloud based solutions that include text mining, triplestores, an inferencing engine and Linked Open Data enrichment - all running in the cloud. These semantic cloud solutions offer powerful services at a fraction of the cost. There's no need to provide your own hardware and software. The services are managed. And you pay as you go.

Specifically related to this white paper, these solutions should include an enterprise triplestore and related tools to build applications, link to open data, and perform text mining. Users should be able to try this for free and assess if the results meet their needs.

8. SEMANTIC EXPERTISE

Deep knowledge and experience working with all aspects of semantics

Semantic technology and services consist of more than a database. This paper has outlined the most important aspects of triplestores. Experience, however, is also critical and can be difficult to find because it encompasses knowledge of inferencing, text mining, a graph database, ontologies, search and integration with content stores – just to name a few topics to consider. Buyers should be working with vendors that have all of these skills and technology as well as a partner ecosystem should deeper knowledge in any single area be needed.

Oftentimes when text mining is involved in the solution, a need for content curation services will be very important. Look at the experience of these curators. Do they know your domain? What projects have they worked on? How are their results communicated to the client?

Development skills deploying custom applications will often come into play. In some cases, you may want your triplestore and your semantic application customized for your work processes and/or embedded inside a larger search, discovery or analytics applications. Since these graph databases represent data in the form of linked relationships, analytic applications may include data visualization. Experience integrating with link analysis tools (relationship graphing) and working with proven partners may be of importance to you.

Oftentimes, triplestores are deployed in production environments along with NoSQL databases (content stores). How much experience does your vendor have working with some of the better known open source and enterprise solutions?

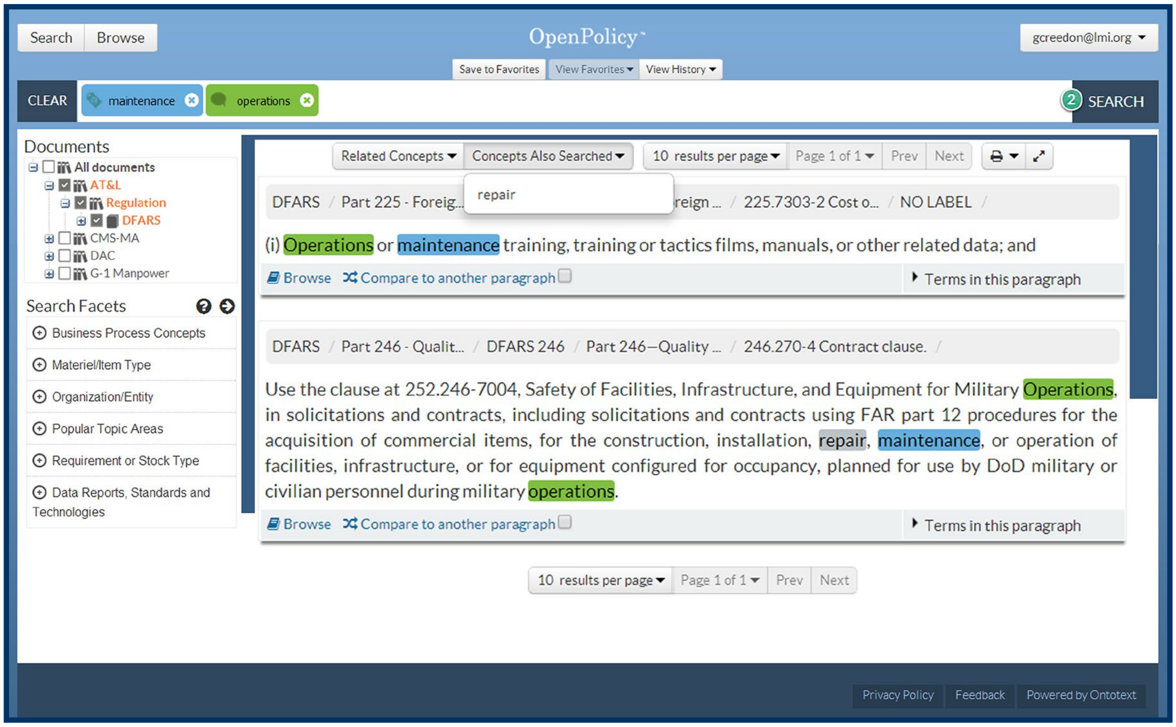
Overall, a knowledge of how to optimize the performance of the triplestore, deep experience with SPARQL, engineering resources, text mining experts, knowledge of linked open data, curation, and data integration are all very important. The best way to find out whether or not your vendor has these skills is to ask them to review successful production deployments. In the next section, we will explore three use cases that may help you understand how this technology can be applied.

USE CASES FOR TRIPLESTORES

There are countless use cases for triplestores that are being explored and deployed worldwide. Some examples are:

- **Search and Discovery**
- **Dynamic Information Products**
- **Personalized Content and Recommendations**
- **Data Visualization**

Search and discovery covers a broad set of applications but generally refers to any use case where organizations want to improve the content delivered through search. The ability to do this with semantic search improves the results. Consider the following example from OpenPolicy™ created by LMI:



DFARS / Part 225 - Foreig... / ...reign ... / 225.7303-2 Cost o... / NO LABEL /

(i) Operations or maintenance training, training or tactics films, manuals, or other related data; and

DFARS / Part 246 - Qualit... / DFARS 246 / Part 246—Quality ... / 246.270-4 Contract clause. /

Use the clause at 252.246-7004, Safety of Facilities, Infrastructure, and Equipment for Military Operations, in solicitations and contracts, including solicitations and contracts using FAR part 12 procedures for the acquisition of commercial items, for the construction, installation, repair, maintenance, or operation of facilities, infrastructure, or for equipment configured for occupancy, planned for use by DoD military or civilian personnel during military operations.

Figure 6: OpenPolicy™ Search and Discovery: Knowledge Drives the Search

OpenPolicy™ allows users to search using key words and concepts that are part of a thesaurus. This is done at the paragraph level across very large sets of pages, all of which have been semantically indexed inside the triplestore. The example above shows a user searching on the words "maintenance" and "operations." The user gains the advantage of having the system automatically also search on "repairs." Related concepts are recommended to the user. The user is also given the option of comparing this paragraph to other paragraphs in other documents, paragraphs that are also relevant.

OpenPolicy™ indexes content in an intelligent way using the thesaurus which has key terms and phrases about specific subject domains. A multi-step process is used to curate and classify content and all of this is hosted by LMI in Tyson's Corner, Virginia. This becomes the knowledge-base, the brains behind the highly targeted search. The entire library of documents is analyzed including words and phrases. A large set of potential thesaurus terms are extracted from the library. Subject matter experts use the Simple Knowledge Organization System (SKOS) to create relationships among the terms in the thesaurus. They create synonyms, acronyms, and other relationships that OpenPolicy™ translates into semantic predicates. These relationships enable OpenPolicy™ to build the intelligent index that guides the user to smarter results. The documents, indexes, thesaurus and semantic facts are all contained within the triplestore. They can be updated, and managed at any time.

A second use case has to do with targeted semantic recommendations made for both internal users and website visitors. Consider the process flow described below.

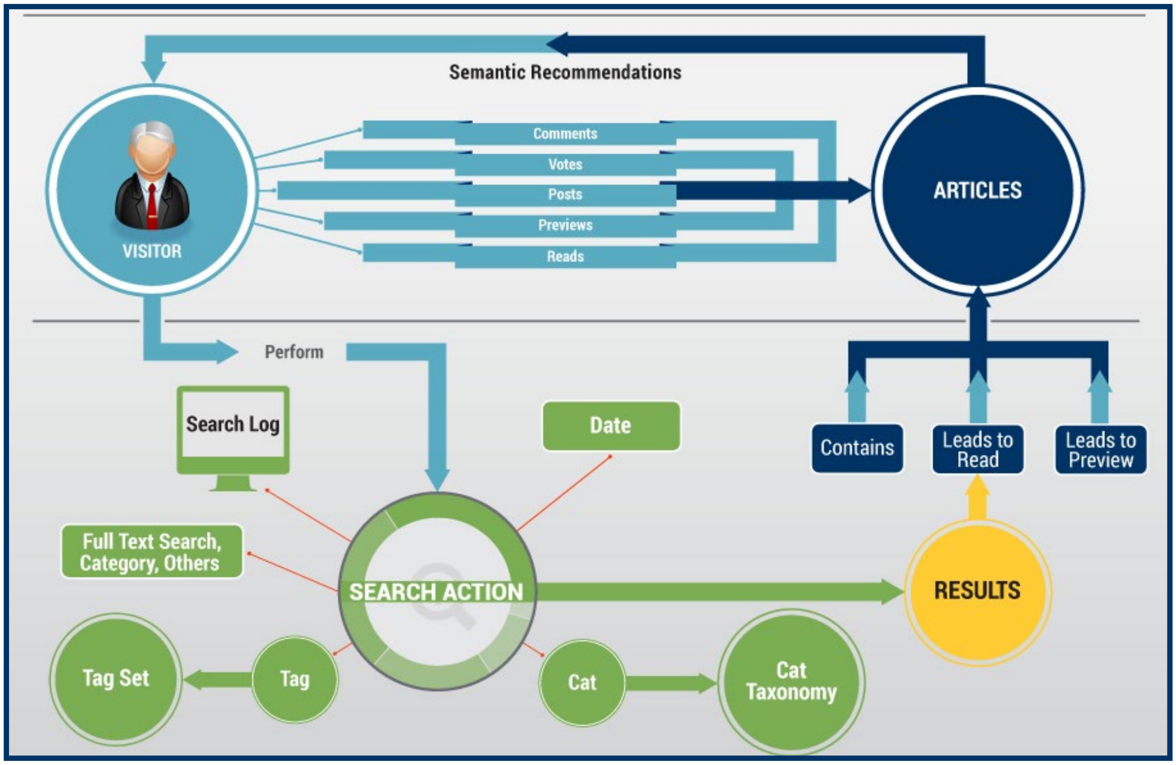
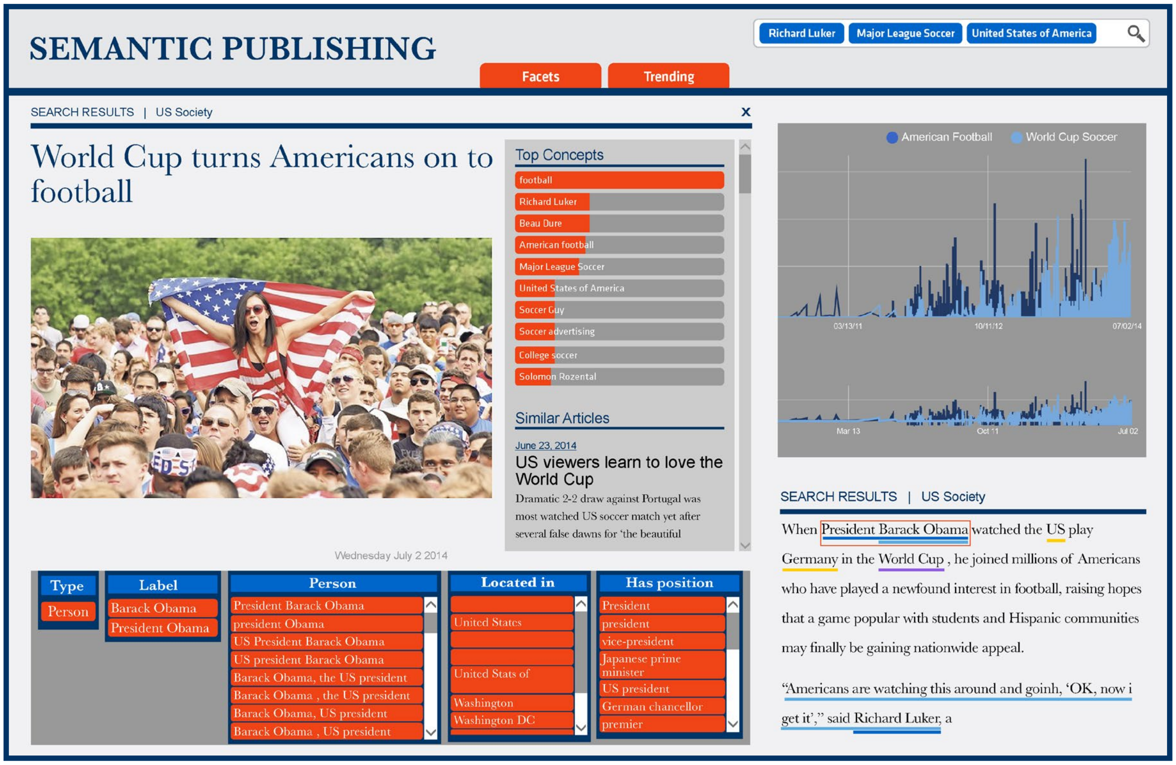


Figure 7: Semantic Recommendation Process Flow

In Figure 7, a website visitor performs search actions over time. The full text search criteria, search categories, date of the search and the log can all be maintained in the triplestore. Search actions are classified and enriched using a taxonomy of terms. In the darker blue, this semantic knowledge is used to deliver related articles which enhance the website experience. This leads to more time on the site, targeted recommendations, increased customer loyalty, revenue and advertising. Since website visitors often comment, vote, post and perform other actions, all of this can also be used in the semantic targeting applications. This process is often part of larger semantic publishing applications. What do they look like?



SEMANTIC PUBLISHING

SEARCH RESULTS | US Society

World Cup turns Americans on to football

Wednesday July 2 2014

Top Concepts

- football
- Richard Luker
- Beau Dure
- American football
- Major League Soccer
- United States of America
- Soccer Guy
- Soccer advertising
- College soccer
- Solomon Rozental

Similar Articles

June 23, 2014

US viewers learn to love the World Cup

Dramatic 2-2 draw against Portugal was most watched US soccer match yet after several false dawns for 'the beautiful

Type	Label	Person	Located in	Has position
Person	Barack Obama	President Barack Obama		President
	President Obama	president Obama	United States	president
		US President Barack Obama		vice-president
		US president Barack Obama		Japanese prime minister
		Barack Obama, the US president	United Stats of	US president
		Barack Obama , the US president	Washington	German chancellor
		Barack Obama, US president	Washington DC	premier
		Barack Obama , US president		

SEARCH RESULTS | US Society

When President Barack Obama watched the US play Germany in the World Cup , he joined millions of Americans who have played a newfound interest in football, raising hopes that a game popular with students and Hispanic communities may finally be gaining nationwide appeal.

"Americans are watching this around and goinh, 'OK, now i get it'," said Richard Luker, a

Figure 8:Internal Use of Semantic Publishing

There are two different application UIs that one should consider when thinking about "Semantic Publishing." Most of us have seen a UI with recommendations on the right. Google is one example. Obviously this can also be achieved for your own website given a triplestore and knowledge base powering the site.

The example in Figure 8 is what a publishing analyst would see to better understand the articles trending on their site based on key words and phrases that are part of the search. In the upper right, the analyst sees how the phrases "American Football" and "World Cup Soccer" are trending during the 2014 World Cup. They have visibility into related articles that are displayed to the readers. An analysis of the article with highlighted words and phrases is available for review. The analyst can also look at faceted search results, and quickly modify the search using related words and concepts. This provides intelligence about site performance in a very timely way. As you would imagine, since all other articles are semantically analyzed and indexed, the relevancy for the website visitor reaches heights never achieved before. All of the data is stored inside a triplestore.

As we have learned, graph databases store data in the form of relationships. For examples "Kurt Heiland specializes in otolaryngology" and "Dr. Heiland prescribes fluticasone propionate" are two statements that can be expressed as RDF statements in a triplestore. These RDF statements can also be used to draw a visual graph with interconnected nodes linked by the predicate of the triple. An example of a graph like this is shown in Figure 9.

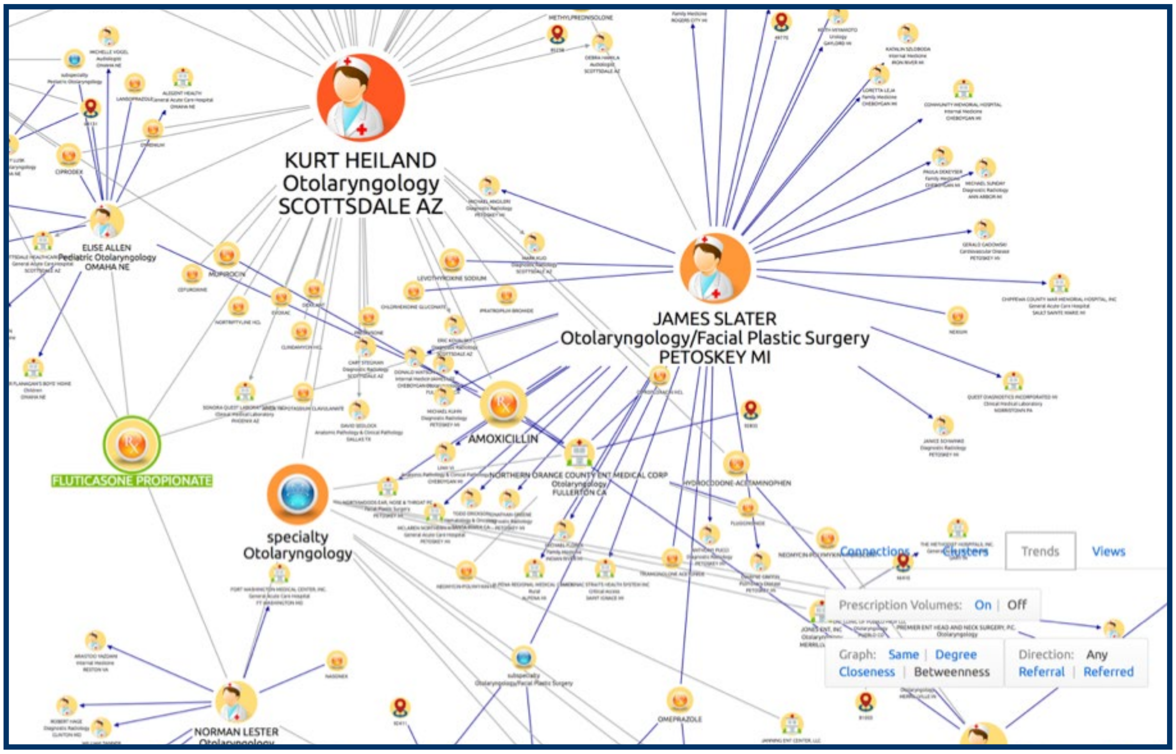


Figure 9: Triples shown in a visual graph. Relationship graph courtesy of Clarté

This type of analysis is especially useful in pattern detection, anomaly identification and network analysis. It shows entities linked together that simply cannot be accurately expressed in the form of charts. In some cases, it is also important to show these relationships geospatially where linked entities are drawn on a map. Similarly, showing when the relationships were formed using a timeline can also expose important connections made at critical times. These relationship graphs are very useful in social network analysis, homeland defense, fraud analysis, adverse event analysis, supply chain analysis and much more.

CONCLUSIONS

Triplestores offer businesses the ability to make sense of text and data and easily ingest data in a schema agnostic manner. Customers gravitate to this technology because it substantially lowers the efforts to query diverse and evolving data coming from different sources. Since triplestores speak the same language from one store to the next, it's usually easy to swap the data if needed.

This paper discussed many other advantages of this new approach to database management including expressive queries, scalability and provenance (the ability to track where every piece of data came from). But all triplestores do not have the same level of maturity. Advanced functions exist in some and are absent from others.

Newer players in the market may not support SPARQL 1.1 and therefore are missing a lot of the capabilities that are a must for a mature, structured query language, e.g. sub-queries, aggregates, and federation. Property paths, for example, allow users to retrieve all the nodes that are connected to a given person through a chain of relationships of a specific type. Users should look for mature and efficient support of inferencing since it holds the power to identify important facts, run queries faster and deliver more comprehensive, accurate results in the form of search, analytics and personalized recommendations.

Triplestores are being used today to power some of the most progressive knowledge management applications in the world. For those who have not worked with this technology, these databases can dramatically increase productivity by delivering semantically enriched search results. They deliver highly scalable solutions that promise to transform the web experience by delivering content based results aligned with user preferences, historical search patterns and new content that is continuously adapted and enriched in real time.

Do not underestimate the need for tight integration with text mining pipelines. Your unstructured data holds the facts needed to drive knowledge based search. When these two technologies (text mining and graph databases/triplestores) are tightly coupled, they will produce breathtaking results that business executives have largely only envisioned at this point in time. The ability to perform research faster, accurately analyze a complete picture of your customers, drive more traffic to your website, uncover hidden relationships across all of your data – these applications and many more can be addressed with a full complement of technology.

For further guidance on this topic and additional detail, contact us at **info@ontotext.com**. You can learn more about this special form of database at **www.ontotext.com** or reach Ontotext directly by calling **1-866-972-6686.**

ABOUT ONTOTEXT

Ontotext has been providing semantic technology solutions since 2000. Headquartered in Sofia, Bulgaria, Ontotext maintains offices in London and near Washington, DC. The Ontotext portfolio of products and solutions is anchored by GraphDB™, an RDF triplestore that has been proven to scale in enterprise resilient production environments. GraphDB™ was formerly known as OWLIM. Many of the capabilities cited in this paper were explicitly requested by Ontotext customers and needed to power enterprise semantic implementations. To assist companies in creating world class semantic solutions, Ontotext expanded it's portfolio to include Semantic Enrichment, Semantic Data Integration and Semantic Curation products. This stack has been used to deliver solutions to the BBC, Euromoney, The Financial Times, AstraZeneca, Logistics Management Inc. (LMI), The Getty Trust, The House of Parliament, IET and many other companies throughout the world. You can learn about Ontotext Media & Publishing, Ontotext Life Sciences and Ontotext Compliance & Document Management by visiting **www.ontotext.com**. Ontotext regularly holds educational webinars to help both business and IT better understand the challenges and solutions in applying semantic technology. **Sign up today.**

ontotext

ontotext

www.ontotext.com | 1 866-972-6686