SOCIAL NETWORK ANALYSIS

A Survey on Semantic Web and Big Data Technologies for Social Network Analysis

Sercan Kulcu • Erdogan Dogdu • A. Murat Ozbayoglu

Computer Engineering Department, TOBB University of Economics and Technology, Ankara, Turkey

2016 IEEE International Conference on Big Data | Washington D.C.

Outline

01 **Motivation & Challenges**

Why social network analysis needs both Semantic Web and Big Data

02 **Semantic Web Technologies**

RDF, RDFS, OWL, SPARQL, triplestores & vocabularies

03 **Big Data Architectures**

Parallel processing frameworks & NoSQL database technologies

04 **Open Research Directions**

Privacy, profile matching, scalable RDF processing

05 **Conclusion**

Synthesis and outlook

Motivation

Two grand challenges in analyzing large-scale social networks

1

Scale

Social networks generate massive, fast-growing, unstructured or semi-structured data that must be processed in a reasonable time.

2

Integration

Distinct datasets from heterogeneous sources (Facebook, Twitter, Google+...) must be merged into a semantically consistent whole.

“Friends are likely to have similar interests” — SNA aims to uncover hidden associations and structural knowledge from social data by combining graph algorithms, Semantic Web modeling, and Big Data-scale processing.

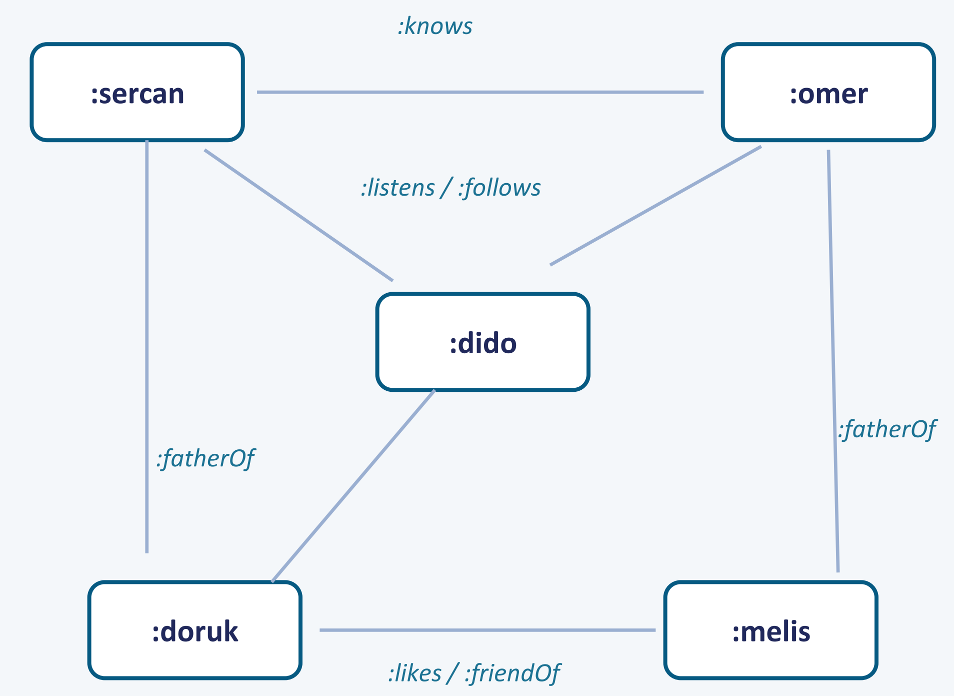
The RDF Data Model

Representing social relationships as a graph of triples

Triple =

⟨subject, predicate, object⟩

- Each triple = a labeled directed edge
- A set of triples forms an RDF graph
- Naturally matches social network structure (people = nodes, relations = edges)
- Queried using SPARQL, the standard graph pattern-matching language



Adapted from Fig. 1 in the source paper

Semantic Web Building Blocks

From raw triples to query-able social knowledge

RDF
Graph data model: subject–predicate–object triples linking resources

→ **RDFS**
Vocabulary to describe classes and properties / relationships between them

→ **OWL**
Richer ontology language for machine-processable class & property semantics

→ **SPARQL**
Standard query language for pattern-matching over RDF graphs

→ **Triplestores**
Storage & SPARQL-endpoint systems: Jena TDB, Virtuoso, AllegroGraph, RDFox...

Frequently Used RDF Vocabularies for SNA

FOAF
Friend-of-a-Friend — user profiles & friendships

SIOC
Semantically-Interlinked Online Communities — forums, blogs, wikis

SKOS / MOAT
Concept organization & tag semantics

SemSNI
Tracks member visits & private messages

RELATIONSHIP
Vocabulary describing relation types between people

OntoSNA / MUTO
General-purpose SNA & tagging ontologies

Triplestores: Centralized vs. Distributed

Storage & query performance is critical for real-time SNA applications

Centralized (single machine)

- Jena TDB
- RDF4J (Sesame)
- RDFSuite
- AllegroGraph
- RDF-3X
- Hexastore
- RDFox

Distributed (multi-machine)

- Virtuoso
- EAGRE
- TriAD
- 4Store
- TripleRush

Key finding (Bizer & Schultz): Virtuoso performs best on large datasets; Sesame performs better on smaller datasets — no single triplestore dominates across all workloads.

Big Data Architectures for SNA

The 5 Vs and the two dominant architectural families

Volume | Velocity | Variety | Value | Veracity

Parallel Processing

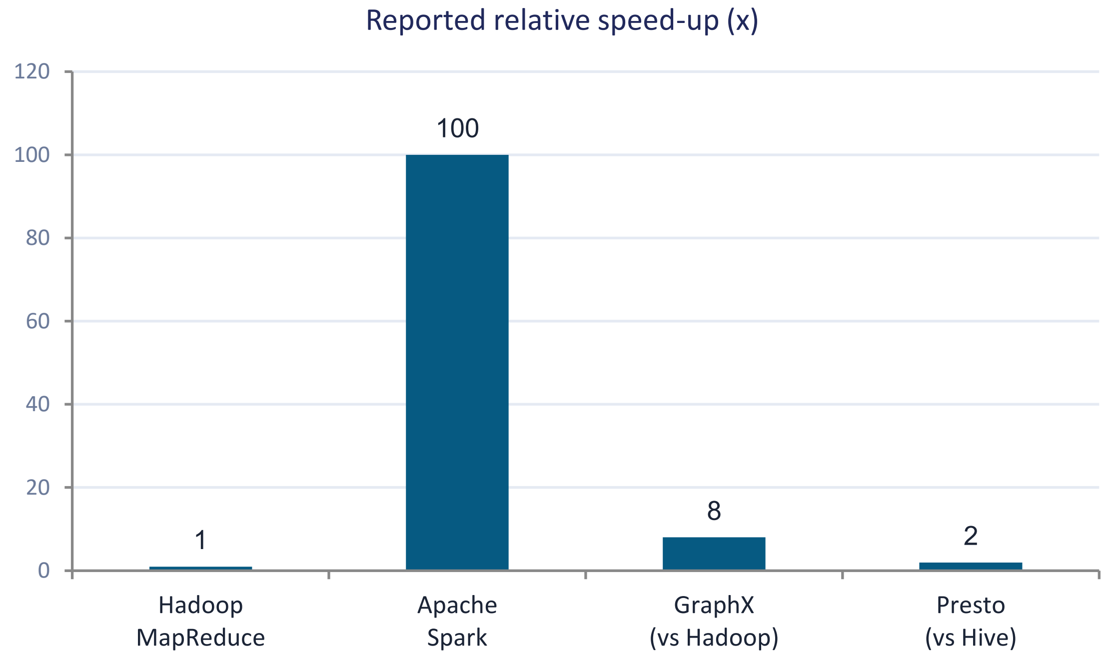
- MapReduce / Hadoop (HDFS)
- Apache Spark (in-memory RDD)
- Graph engines: Pregel, Giraph, GraphX, GraphLab
- Scripting: Pig Latin, HiveQL

NoSQL Databases

- Tabular: HBase, Cassandra, MongoDB, Accumulo
- Graph DBs: Neo4j, DEX, Titan, HyperGraphDB
- Graph query languages: Cypher, Gremlin, GraphQL

Spark vs. Hadoop MapReduce

Why the field is porting RDF & graph workloads away from batch MapReduce



Figures are illustrative, aggregated from claims cited across the surveyed literature.

Why Spark wins for RDF/graph workloads

- In-memory RDDs avoid repeated HDFS read/write between iterations
- Native support for iterative computation — critical for graph algorithms
- GraphX (Spark) reported up to 8× faster than Hadoop MapReduce
- Facebook migrated production RDF/graph pipelines from Hive to Spark
- Hadoop MapReduce remains preferred for large batch jobs where the full graph can't fit in memory

What Big Platforms Actually Use

Big data technologies behind major social network services

Platform	Big Data Technologies Used
Facebook	Cassandra, HBase, Giraph, Presto, Neo4j
Twitter	Cassandra, FlockDB, HBase, Neo4j, Pig
LinkedIn	AllegroGraph, HBase, MongoDB, Neo4j
Amazon	DynamoDB, SimpleDB
eBay	MongoDB, Neo4j
Google+ / YouTube	BigTable
Foursquare	Cassandra, CouchDB, InfoGrid, MongoDB, Riak
Instagram / Flickr	Cassandra / MongoDB, Neo4j

Graph Databases for SNA

Why graph storage often beats tabular storage for social data

Graph DB vs. Tabular DB

- Query time is largely independent of graph size
- Nodes are directly connected — no costly join operations
- Better suited for densely connected, complex social data
- Tabular DBs remain simpler for structured, high-volume writes (e.g., timelines, logs)

Popular Graph Databases

Neo4j	RDF TripleStore	DEX	InfiniteGraph
HyperGraphDB	FlockDB	OrientDB	Titan

Benchmark result: Mathew & Kumar compared Neo4j, FlockDB, OrientDB, InfoGrid, and AllegroGraph on SNA workloads — Neo4j outperformed the others.

Open Research Directions

Unsolved problems identified across the surveyed literature

01

Privacy & Trust

De-anonymization risk and preserving privacy while sharing social data remains open.

02

Limited API Access

Platforms rarely expose the full graph — algorithms must work correctly on partial data.

03

Cross-Platform Identity

Matching the same user's profile across multiple social networks is still unsolved.

04

Scalable RDF Processing

Distributed RDF stores (Spark-, Hadoop-, Accumulo-, Cassandra-based) are actively evolving.

Conclusion

Ontologies enable schema-less, heterogeneous data integration — social data maps naturally onto RDF triples.

Architecture choice depends on the workload: Hadoop-based systems suit large-scale batch processing; Spark-based systems suit interactive, real-time queries.

Decentralized RDF triplestores are the key technology for handling web-scale RDF data efficiently, queried via SPARQL translated into parallel jobs.

Big Data ≠ size alone

It is also about diversity, heterogeneity of formats and sources, and decentralized control over distributed data. Combining Semantic Web and Big Data technologies is essential to scale social network analysis to the web.

Selected References

[1] Kulcu, S., Dogdu, E., & Ozbayoglu, A. M. (2016). A Survey on Semantic Web and Big Data Technologies for Social Network Analysis. IEEE International Conference on Big Data.

[2] Malewicz, G. et al. (2010). Pregel: A System for Large-Scale Graph Processing. ACM SIGMOD.

[3] Xin, R. S. et al. (2013). GraphX: A Resilient Distributed Graph System on Spark. GRADES Workshop.

[4] Bizer, C. & Schultz, A. (2009). The Berlin SPARQL Benchmark. Int. J. on Semantic Web and Information Systems.

[5] Mathew, A. B. & Kumar, S. M. (2015). Analysis of Data Management and Query Handling in Social Networks Using NoSQL Databases. ICACCI.

[6] Marcu, O.-C. et al. (2016). Spark versus Flink: Understanding Performance in Big Data Analytics Frameworks. IEEE Cluster.

Note: Full reference list (104 citations) available in the original paper.