Smart-Aleck: An Interestingness Algorithm for Large Semantic Datasets

Kavi Mahesh and Pallavi Karanth

Centre for Ontological Engineering, Department of Computer Science,
PES Institute of Technology, Bangalore 560085 INDIA
drkavimahesh@gmail.com, karanthpallavi@gmail.com

Abstract:

Not every fact in a large semantic dataset is of interest to an application. In the *Smart-Aleck* project, we have designed and implemented an interestingness algorithm that filters facts and joins them to generate new facts with higher levels of interestingness. The algorithm defines different levels of interestingness based on the semantic operations involved in generating interesting facts. The application of the algorithm is a Web site that presents a new interesting fact, rendered in English, each time users visit or refresh the page. The facts are generated from an integration of over half a billion triples from large semantic datasets including YAGO, Dbpedia, DataHub and Timbl. The uniqueness of the *Smart-Aleck* algorithm lies in its ability not merely to select interesting facts from the datasets but to generate new facts by joining two or more facts, possibly from different sources, by applying several comparison, chaining, grouping, aggregation and quantification operations on RDF triples. The implementation of *Smart-Aleck* on the web site is useful to everyone on the net to satisfy their curiosity, acquire general knowledge and design quizzes. It also has business potential as a feed for "fact-of-the-day" applications on cell phones and tablets.

Keywords: Interestingness, fact generation, semantic web, algorithm, dataset.

1 Introduction

Large semantic datasets are becoming available through recent developments in Semantic Web technologies. These datasets are typically represented as RDF triples or quads, along with suitable ontologies, and have immense potential for creating a variety of semantic applications. However, not all the facts in these datasets are of interest to a particular application. Despite the continuing increase in processing speeds and network bandwidths, the enormous size of the datasets poses significant limitations on building useful applications. As such, there is an immediate need to devise effective filters that can select a significantly smaller but relevant subset of a large semantic dataset for a given application.

In the *Semantic Smart-Aleck* project, we have attempted to design and implement a filter for large semantic datasets that selects *interesting* facts, leaving out the vast majority of un-*interesting* facts while also generating new facts through various semantic operations. The *Smart-Aleck* application on the Web renders the selected interesting facts in English and presents them, one at a time in a random order, to arouse the interest or curiosity of a visitor to the site.

In the rest of this paper, we present prevailing notions of interestingness and propose an algorithm for generating interesting facts. The algorithm is based on our proposal that there are (at least) six levels of interestingness that result in sets of facts that evoke corresponding levels of emotional responses from the reader:

0. Useless →1. Boring → 2. Notable → 3. Interesting → 4. Fascinating → 5. Captivating → 6. Amazing → 7. Mindboggling → ..

The *Smart-Aleck* application is available on the Web through our landing page at **http://sites.google.com/site/semanticsmartaleck** (or directly at http://119.82.126.184/SemanticSmartAleck/FactGenerator.php).

2 Interestingness

Various senses of the term interestingness are prevalent in several disciplines:

1. In data mining, interestingness is used as an objective criterion to select certain patterns or rules over many others to address the problem of over-generation [1].
2. On the same lines, interestingness is used as a measure of unexpectedness in knowledge discovery algorithms [2, 3].
3. It is used in proprietary methods to rank media, such as photographs on flickr.com [4] (although Yahoo seems to own a patent on this [5]).
4. In artificial intelligence, estimates of interestingness have been used to control the combinatorial explosion of logical inferences in a reasoning engine [6].
5. In cognitive science, interestingness refers to a measure of the potential to evoke an emotional response from a reader [7].
6. In discourse processing theory, interestingness has been proposed as an important factor in determining discourse-level semantics of a piece of text [8].
7. Games such as quizzes and trivia contents also employ interestingness as a key notion, e.g., Mental Floss http://mentalfloss.com/amazingfactgenerator [9].

We propose, in this paper, a new algorithm for filtering, joining and generating interesting facts from large semantic datasets. We call the algorithm *Smart-Aleck*. It is being implemented and a prototype running system is available on our server which may be accessed through our landing page http://sites.google.com/site/semanticsmartaleck

This site displays a new interesting fact each time a user (re)-visits (or refreshes) the page. These facts are automatically generated from multiple, very large semantic datasets through an implementation of the *Smart-Aleck* algorithm with substantial amounts of pre-processing and indexing.

We understand that interestingness is both application specific and user specific and requires significant customization, localization and personalization. Smart-Aleck is an initial attempt to demonstrate a reasonable model of interestingness, albeit generic, to build a platform on which better interestingness algorithms can be developed for filtering semantic datasets for other semantic applications and generating complex facts from the triples in the available datasets.

The approach taken in *Smart-Aleck* to filter and generate facts is very different from those in earlier attempts such as Shortipedia [10] which is an aggregator of assertions about a single entity from the web of data. *Smart-Aleck* generates interesting facts from facts in datasets which may not be about a single entity. Moreover, it applies other semantic operations, including chaining, comparison, grouping and quantification to derive new facts which were not present in the datasets.

3 Levels of Interestingness

The *Smart-Aleck* algorithm proposes that there are (at least) six levels of interestingness as follows:

1. **Level 1 – Boring/mundane:** simply select a random fact from the data set and render it in natural language (English) for human consumption. For example (note that these are actually generated by *Smart-Aleck*),

"Hey, by the way, United States Occupation of Dominican Republic ended in September of 1966."

2. **Level 2– Notable:** Select popular facts (based on user ratings, where available) or facts about famous people and render them. For example,

   "Did you know that Albert Einstein was born in Ulm?"

   These two levels merely filter and select existing facts. The higher levels generate new facts from the datasets.

3. **Level 3 – Interesting**: Apply various quantifiers, grouping operators, and minimum-maximum operators to select facts which are interesting. For example,

   "Wow, Cecil R Richardson has won prizes more than anybody else!"

4. **Level 4 – Fascinating** (Join Interestingness): Perform a relational join operation on two or more different relations, possibly from disparate datasets (or data sources) to generate interesting facts. Note that earlier criteria of Popularity can also be superimposed on the Join criterion. For example,

   "Hey, Ecuador, American Samoa and British Virgin Islands have the same currency United States Dollar!"

   Join operations also result in interesting **chaining** of the same (i.e., transitivity) or different relations. For example,

   "By the way, Alan Turing, computer scientist, mathematician, and cryptographer, had academic advisor Alonzo Church whose academic advisor was Oswald Veblen."
   "Did you know that, Albert Einstein who won Nobel Prize in Physics was born in Ulm?"

5. **Level 5 – Captivating** (Join-Quantified): Combining quantifying and grouping operations with joins and then filtering based on comparison and cardinality operators generates new nontrivial facts involving quantitative comparisons (e.g., percentage, majority, more, most, very few, only, etc.). For example,

   "Did you know there are more actors born in Boston than Houston?"
   "Wow, the number of actors born in New York City is 2 times the number of actors born in Los Angeles!"

6. **Level 6 – Amazing** (Complex and Logical): Surprising discoveries are made by taking two or more closely related entities participating in the same or similar relations and applying logical operations on them. For example, *Smart-Aleck* generated this fact:

   "Hey, how come The Beach Boys won Grammy Lifetime Achievement Award but never won a Grammy Award?"

We recognize that further levels of interestingness as well as several other combinations of relational, comparative and other semantic operations are possible in generating interesting facts from large semantic datasets. We intend to extend and improve the *Smart-Aleck* algorithm continually to produce better results.

4 *Smart-Aleck* Algorithm

The algorithm has three phases which are presented in the form of pseudo-code below:

***Smart-Aleck* algorithm: Phase 1 – Filtering**

```
Load and index triples from all datasets;
Filter based on the Smart-Aleck OWL ontology of relation types;
      Map selected relations to Smart-Aleck interestingness operations;
      Mark all facts retained by filters with the selecting operation;
Store and index filtered facts in the database;
```

In order to avoid hard-coding of relations in the algorithm, an OWL Ontology of relation types was developed, a snapshot of which is shown in Fig. 1. Relations and their combinations are selected based on their types in this ontology. The ontology may be modified or enhanced to change the behavior of *Smart-Aleck* to suit a particular application or user community.

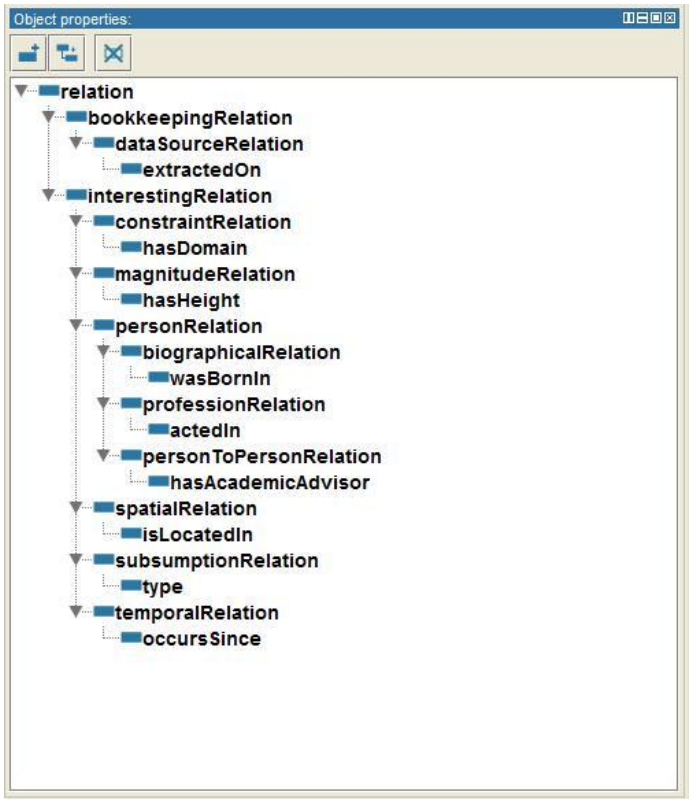

Fig. 1. Snapshot of Relation Types in *Smart-Aleck* OWL Ontology

***Smart-Aleck* algorithm: Phase 2 – Fact generation**

```
for level 1 to level 6 do
      Select relations and relation combinations based on their types
      in the ontology;
```

```
Apply join, grouping and aggregation/quantification operations to
generate new facts for the level of interestingness;

Store and index generated facts in the iFacts table along with
the generating operator and comparison operators;
```

To improve run-time performance, *Smart-Aleck* pre-computes interesting facts at all the levels and stores them in a single de-normalized database table that is ideal for selecting a fact randomly from the set. The table called iFacts can store up to three triples from which the new fact is generated. It also stores the operators used to join the individual facts and the corresponding comparators to be used to render the fact in Phase 3. The schema (i.e., the columns) of the iFacts table is shown below:

seqNum	createdOn	NT1Arg1	NT1Rel	NT1Arg2	NT2Arg1	NT2Rel	NT2Arg2
NT3Arg1	NT3Rel	NT3Arg2	rating	lastUsed	NT1ID	NT2ID	NT3ID
Comparator	iLevel						

***Smart-Aleck* algorithm: Phase 3 – Fact rendering (at run-time)**

```
Use the id of the previous fact (for any user) as a seed to generate a
new random number;

Select random fact from iFacts using the random number, desired iLevel
and feedback rating of facts;

Apply data conversions (e.g., from strings to numbers or vice-versa);

Apply the BNF grammar (shown below) to generate the fact in English
from one or more triples stored in the fact;

Add descriptions to persons from Dbpedia "description" relation;

Record the timestamp of when the fact was last shown;

// to avoid sending the same fact again)

Provide option for user to rate the fact and give feedback;
```

Grammar for Fact Rendering (in BNF-like notation):

```
FACT ::= PREFACE NT [CONN NT [CONN NT]]
PREFACE ::= "Did you know that" | "Hey, by the way" | "Wow," | ..
NT ::= ARG1 REL ARG2
ARG1 ::= cleanUp(NT.arg1)
ARG2 ::= cleanUp(NT.arg3)
REL ::= getText(NT.rel)
CONN ::= getText(CONN)
```

The function cleanUp performs simple string manipulations on the three parts of a triple to make them readable. For example, it removes underscores, normalizes whitespace characters and deletes unrecognized Unicode characters. The function getText maps relations names in the triples to corresponding string patterns to generate readable relation names. It also does clean up operations such as splitting the relation name at each uppercase character.

5 Datasets and Implementation

Smart-Aleck started with the YAGO2 dataset [11, 12, 13]. YAGO (Yet Another Great Ontology, http://www.mpi-inf.mpg.de/yago-naga/yago/index.html, Max Planck Institute for Informatics, Germany) is a highly accurate algorithmic aggregation of WordNet [14] and Wikipedia along with other datasets such as GeoNames. The version of YAGO currently used by Smart-Aleck, YAGO2, contains over 447 million facts (thus approaching half the size of the "Billion Triples" benchmark). The convertor utility provided by the developers of YAGO2 was used to load the triples into an RDF database (using MySQL) after making some modifications to the loading process to speed it up. The *Smart-Aleck* ontology of relation types was used to exclude certain triples that contain merely bookkeeping information (which does not generate any interesting fact in *Smart-Aleck*).

In addition to YAGO, *Smart-Aleck* has attempted to include the following data sets from the Semantic Web Billion Triples Challenge-2012 Dataset: Dbpedia, DataHub and Timbl. In particular, the "Description" relation in the *person data* of Dbpedia which is absent in YAGO was used to describe individuals while rendering facts (e.g., see the Alan Turing description in the example above for **Level 4 – Fascinating**).

Smart-Aleck was built primarily on a single HP ProLiant ML110 G7 server with 16 GB RAM and 8TB disks. Some of the software packages used include Ubuntu Server 12.04 Precise Penguin, MySQL, Java, Apache, PHP, Jena, Jena-SDB, YAGO converters, TDB-Loader, Protégé and SPARQL.

6 Results

We believe that *Smart-Aleck* is already able to generate many non-trivial facts which are not obvious from the datasets. Work is currently ongoing to load further datasets and to generate a large number of interesting facts at various levels of interestingness. We will provide the final sizes and other statistics at the conference. We are also developing a comprehensive ontology of relation types based on previous work on ontological engineering [15].

7 Further enhancements

Apart from adding further datasets (such as Freebase) and improving the algorithm to generate more facts, we plan to provide a mechanism for user feedback and rating of facts. In the future, we also plan to investigate the application of machine learning techniques to learn from user feedback and improve the filtering and generation of interesting facts. We also plan to add related (or linked) images and multi-media artifacts to make the rendering of facts more appealing to users. Further, localization and personalization possibilities will be considered to further improve the usability of the application.

8. Design Choices, Lessons and Conclusion

Smart-Aleck addressed an important problem in using semantic datasets, namely, selecting a subset of facts that is of interest to a particular application or user. It proposed a new interestingness algorithm using which it was not only able to select interesting facts but also generate several types of new nontrivial facts which were not obvious in the datasets (without having to compute the deductive closure of the given facts). Its implementation in a public web site presents each visitor with an interesting fact with the potential of providing subscription feeds to cell phones and tablets.

In designing and implementing *Smart-Aleck*, several design choices were dictated by the sizes of the datasets involved. YAGO was chosen initially because of its very high accuracy. Most of the filtering and fact generation had to be pre-computed to obtain quick run-time performance, given our limited computing facilities. We also had to

deal with high degrees of redundancy in the datasets which generates duplicate facts. Apart from the typical learning curve associated with such large datasets, we also learned how to model the data using an ontology of relation types in order to filter and select only those semantic operations that are likely to generate interesting facts. Several initial choices of tools and packages either did not scale up or had no support for the kind of semantic operations we needed to perform (especially some of the SPARQL-based ones). On the positive side, we found that the datasets do contain information from which *Smart-Aleck* was able to generate a number of interesting facts.

References

1. Geng, L., Hamilton, H. J.: Interestingness Measures for Data Mining: A Survey, *ACM Computing Surveys* (CSUR), Association for Computing Machinery(2006)

2. Padmanabhan, B., Tuzhilin, A.: Unexpectedness as a Measure of Interestingness in Knowledge Discovery, *Decision Support Systems*, 27:3, 303-318 (1999)

3. Hidi, S., Baird, W.: Interestingness: A Neglected Variable in Discourse Processing, *Cognitive Science*, 10:2, 179-194 (1986)

4. Flickr.com's algorithm for ranking photographs and other media.

5. Butterfield, D. S., Henderson-Begg, C. J.,Mourachov, S.: Interestingness Ranking of Media Objects, US Patent Application 11/350,981, Assignee: Yahoo! Inc. (2006)

6. Schank, R. C.: Interestingness: Controlling Inferences, *Artificial Intelligence*, 12:3, Nov., 273-297 (1979)

7. Anderson, R. C., Shirey, L. L., Wilson, P. T., Fielding, L. G.: Aptitude, Learning and Instruction: Volume 3 – *Conative and Affective Process Analyses*, Chapter 12, Snow, R.E., Farr, M. J., eds., Lawrence Erlbaum Associates(1987)

8. Vrandecic, D., Ratnakar, V., Krotzsch, M., Gil, Y.: Shortipedia: Aggregating and Curating Semantic Web Data, Semantic Web Challenge at the International Semantic Web Conference, ISWC-2010 (2010)

9. McGarry, K.: A Survey of Interestingness for Knowledge Discovery, *The Knowledge Engineering Review*,20:1, 39-61(2005)

10. Mental-Floss: Where knowledge junkies get their fix, http://www.mentalfloss.com/amazingfactgenerator

11. Suchanek, F. M., Kasneci, G., Weikum, G.: YAGO: A Core of Semantic Knowledge, *Proc. Sixteenth Int. Conf. World Wide Web, WWW-2007*,697-706, Association for Computing Machinery (2007).

12. Suchanek, F. M., Kasneci, G., Weikum, G.: YAGO: A Large Ontology from Wikipedia and WordNet, in *Web Semantics: Science, Services and Agents on the World Wide Web*, 6:3, 203-217 (2008).

13. Hoffart, J., Suchanek, F. M., Berberich, K., Weikum, G.: YAGO2: A Spatially and Temporally Enhanced Knowledge Base from Wikipedia, *Artificial Intelligence* (In press)

14. Fellbaum, C.: WordNet: An Electronic Lexical Database, MIT Press (1998)

15. Mahesh, K.: Ontology Development for Machine Translation: Ideology and Methodology, New Mexico State University, Computing Research Laboratory MCCS-96-292 (1996)

Appendix: How *Smart-Aleck* Meets the Open Track Criteria: Minimal requirements

1. The application has to be an end-user application, i.e. an application that provides a practical value to general Web users or, if this is not the case, at least to domain experts.
   Yes, Semantic Smart-Aleck provides a public web site that anyone can visit to obtain general knowledge, satisfy their curiosity or spend quality time browsing interesting facts. It may also be useful for designing questions for quizzes.

2. The information sources used
    - should be under diverse ownership or control
      Semantic Smart-Aleck uses YAGO2 which itself is an integration of Wikipedia and WordNet. It also uses additional facts from Dbpedia and DataHub. Attempts are underway to add Freebase and Timbl as well.
    - should be heterogeneous (syntactically, structurally, and semantically), and
      Wikipedia and WordNet are rather heterogeneous, the latter being a computational lexicon of the English language.
    - should contain substantial quantities of real world data (i.e. not toy examples).
      YAGO2 contains 447 million triples. We are adding many more triples (and quads) from the other sources to approach the billion triple benchmark. However, our initial filters do eliminate more than half of the triples.
3. The meaning of data has to play a central role.
    - Meaning must be represented using Semantic Web technologies.
      All data is represented and processed as RDF triples/quads in addition to the use of the YAGO semantic model and our own ontology for mapping various relations to levels of interestingness.
    - Data must be manipulated/processed in interesting ways to derive useful information and
      Data is manipulated based on our ontology of relation types and various combinations of relational, comparative and other semantic operations are used to generate new facts.
    - this semantic information processing has to play a central role in achieving things that alternative technologies cannot do as well, or at all;
      Only facts at Level 1 and 2 could be generated at best using non-semantic technologies. Even at these levels, without our ontology of relation types, most of the facts selected would be too uninteresting. Facts generated at higher levels would be impossible without semantic processing.

Additional Desirable Features

- The application provides an attractive and functional Web interface (for human users)
  A minimally attractive but fully functional web interface has been provided.
- The application should be scalable (in terms of the amount of data used and in terms of distributed components working together). Ideally, the application should use all data that is currently published on the Semantic Web.
  We are working on including more of the available datasets. We believe we are about half-way there.
- Rigorous evaluations have taken place that demonstrate the benefits of semantic technologies, or validate the results obtained.
  Evaluation of the interestingness of generated facts has not yet been done in a formal way.
- Novelty, in applying semantic technology to a domain or task that have not been considered before
  We believe that our interestingness algorithm is novel. Other fact generators that we are aware of merely select from a small set of manually compiled facts. We are not familiar with any other semantic fact generator that can work with large semantic datasets.
- Functionality is different from or goes beyond pure information retrieval
  Generation of new facts is not possible through mere retrieval.
- The application has clear commercial potential and/or large existing user base
  Apart from ad-revenue for the web site, the application can provide "fact-of-the-day" feeds to cell phone and other subscribers.
- Contextual information is used for ratings or rankings: *Not applicable.*
- Multimedia documents are used in some way
  This is feasible but yet to be implemented. We plan to add images of people mentioned in the facts soon.
- There is a use of dynamic data (e.g. workflows), perhaps in combination with static information
  Not applicable.
- The results should be as accurate as possible (e.g. use a ranking of results according to context)
  Not applicable.
- There is support for multiple languages and accessibility on a range of devices
  It is already accessible from any device running a web browser, including smart phones. All of the code and technologies used are Unicode-compliant. We do not have multilingual data at this time to test in other languages.