Identifying Interesting Assertions from the Web

Thomas Lin
Turing Center
Computer Science & Engineering
University of Washington
Seattle, WA 98195
tlin@cs.washington.edu

Oren Etzioni
Turing Center
Computer Science & Engineering
University of Washington
Seattle, WA 98195
etzioni@cs.washington.edu

James Fogarty
Computer Science & Engineering
DUB Institute
University of Washington
Seattle, WA 98195
jfogarty@cs.washington.edu

ABSTRACT

How can we cull the facts we need from the overwhelming mass of information and misinformation that is the Web? The TextRunner extraction engine represents one approach, in which people pose keyword queries or simple questions and TextRunner returns concise answers based on tuples extracted from Web text. Unfortunately, the results returned by engines such as TextRunner include both informative facts (e.g., "*the FDA banned ephedra*") and less useful statements (e.g., "*the FDA banned products*").

This paper therefore investigates filtering TextRunner results to enable people to better focus on interesting assertions. We first develop three distinct models of what assertions are likely to be interesting in response to a query. We then fully operationalize each of these models as a filter over TextRunner results. Finally, we develop a more sophisticated filter that combines the different models using relevance feedback.

In a study of human ratings of the interestingness of TextRunner assertions, we show that our approach substantially enhances the quality of TextRunner results. Our best filter raises the fraction of *interesting* results in the top thirty from 41.6% to 64.1%.

Categories and Subject Descriptors

H.0 [**Information Systems**]: General

General Terms

Management, Design

Keywords

Information Extraction, Data Pre- and Post-Processing

1. MOTIVATION

Information extraction (IE) is a subfield of natural language processing that seeks to obtain structured information from unstructured text. IE can be used to automate the tedious and error prone process of collecting facts from the Web. Open IE is a relation-independent form of IE that scales well to large corpuses. Figure 1 presents the output of the TextRunner Open IE system [3] in response to the question "*What has the FDA banned?*". TextRunner homes in on such answers as "*ephedra*" and "*most silicone implants*" and frees people from sifting through many Web pages to find the desired answers.

CIKM'09, November 2–6, 2009, Hong Kong, China.

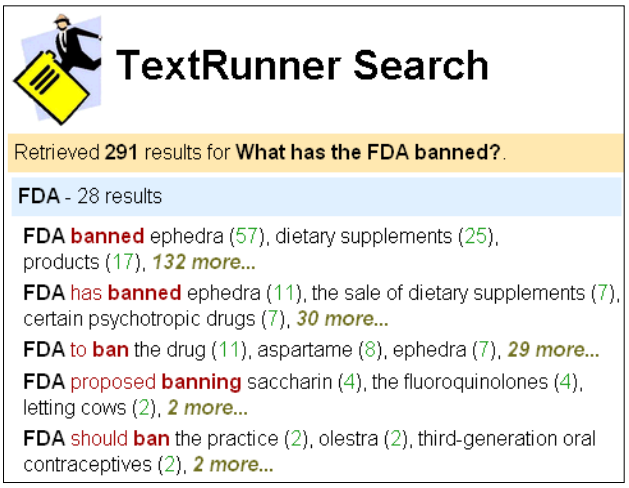

Figure 1. TextRunner results for the question "*What has the FDA banned?*". This paper examines the filtering of such results to focus on *interesting* assertions.

Unfortunately, extraction engines, like search engines, intermix relevant and irrelevant information. This problem is exacerbated in IE systems because they use heuristic methods to extract phrases meant to denote entities and relationships. Thus, in response to the above question, an extraction engine like TextRunner also returns uninformative answers like "*products*".

Extraction engines therefore could be improved by filtering based on models of which extracted assertions are of interest. This problem is especially challenging because *interestingness* can be subjective, personal, and context specific.

This paper proposes the problem of '*interestingness*' for question/answer systems and Web extraction engines, explains why it is particularly acute in the case of extraction, and articulates its connection to previous work. We then introduce and evaluate several practical models of *interestingness* that offer substantial improvements over an assertion frequency baseline.

2. BACKGROUND

2.1 Open Information Extraction

Traditional IE requires pre-specifying a set of relations of interest and then providing training examples for each. Open IE [2] is relation-independent, and instead extracts all relations by learning a set of lexico-syntactic patterns. TextRunner uses conditional random fields to learn a model of how binary relationships are expressed in English. Open IE is highly scalable in that it only needs to make a single pass over the corpus instead of one pass per relation, and this makes it particularly suitable for extracting the knowledge from a massive corpus such as the Web [3].

TextRunner crawls the Web and maps sentences on Web pages into triples of strings of the form **(subject, relation, object).** The *relation* string is meant to denote the relationship between the two *entities*. For example, if the sentence *"Franz Kafka was born in Prague, now in the Czech Republic but then part of Austria"* were found on a webpage, then one extraction would be (*"Franz Kafka", "was born in", "Prague"*). TextRunner has been run on 500 million high-quality webpages yielding over 800 million extractions. These can be queried by entity or relation, or can be used to answer simple questions through pattern matching. Results are returned ranked by frequency because, all other things being equal, extractions that appear frequently on high-quality Web pages are more likely to be correct (the *KnowItAll hypothesis*) [5]. Yet, this method alone will not filter out many assertions that are not *interesting* to people.

2.2 Related Work

2.2.1 Traditional Information Extraction

A key aspect of our study is that, in order to better scale to the full Web, we are studying models that can improve the interestingness of Web extractions in a domain independent and relation independent way. This is important because lexical rules (e.g. "all assertions about what companies Microsoft has bought are interesting") might work well for particular domains or relations but not apply more generally.

In traditional IE, system developers pre-specify a set of relations of interest. For example, the NAGA system has considered methods for evaluating quality of web extractions [7], but their work is grounded in a graph representation based on the specific set of relationships they chose to extract. This limited set of relationships meant they could only evaluate 12 of 50 queries for one of their benchmarks.

2.2.2 Interestingness in Related Domains

The concept of interestingness as a metric has been applied and studied in other related domains. For instance, Flickr recently launched a new feature for identifying "Interestingness" in photos on its site[1]. The Flickr notion is based on social feedback such as click data and comments, supporting the idea that people care about what is interesting and leave indirect clues to where interesting content can be found. We use a similar concept later in learning from how people populate Wikipedia infoboxes.

Similarly, automated mathematical discovery programs require a notion of interestingness in order to identify which potential conjectures and concepts will be of interest to people. Colton and Bundy's survey identifies several key concepts these programs tend to use in deciding what is interesting, including plausibility, novelty, surprisingness, comprehensibility, and complexity [4]. Varying concepts like these have also been occasionally proposed by psychologists to help explain what is interesting [11].

In the area of databases and data mining, Liu et al. found the notion of interestingness helpful for deciding which of a huge number of discovered association rules to present to users [8]. Among other things, they studied the effectiveness of various forms of unexpectedness and successfully applied their ideas to a number of applications.

3. What's Interesting?

At the most general level, we define interesting assertions to be those that a person may find useful or engaging. For any particular query (e.g., "*Einstein*"), the extent to which possible assertions are interesting may vary greatly. For example, a good set of results might include a mix of biographical facts like "*Einstein was born in Germany*" and other interesting facts like "*Einstein's favorite color was blue*". On the other hand, "*Einstein turned 15*" or "*Einstein wrote the paper*" might be less interesting because they express little useful information.

3.1 Specific Assertions

One quality of interesting assertions is that they tend to provide more *specific* information. For example, "*Albert Einstein taught at Princeton University*" is more interesting than "*Albert Einstein taught at a university*" because identifying Princeton as the university is informative. We hypothesize this is one characteristic that can make assertions interesting more broadly in TextRunner.

To operationalize this quality, we define a *specific* assertion as an assertion that either relates multiple proper nouns or an assertion that contains a year. If an assertion relates multiple proper nouns, it is specific because it expresses information about one specific entity relative to another. Similarly, an assertion that contains a year is specific because it contains specific temporal information.

3.2 Distinguishing Assertions

Another quality of interesting assertions might be providing *distinguishing* information about an object. For example, "*Einstein was a man*" is not interesting because the same thing could be said for many people, but *"Einstein was offered the Presidency of Israel"* is interesting because it sets him apart. This is also fairly similar to the earlier notions of surprisingness and unexpectedness.

We operationalize this notion of *distinguishing* using a technique similar to TF-IDF weighting [10]. For our TF component, we define *AssertionFrequency* as the total number of times an assertion occurs. For our IDF component, we define *ObjectFrequency* as the number of times the object (e.g., "*a man*") appears in a sample of ten million random TextRunner assertions. We define an *AFOFRatio(Extraction)* as follows[2]:

$$AFOFRatio(E) = \frac{AssertionFrequency(E)}{ObjectFrequency\big(object(E)\big) + 1}$$

For assertions, the *AFOFRatio* compares how often the assertion appears with how often we would expect the assertion to appear given its object. If the object has extremely high *ObjectFrequency* (e.g., a common word like "*a man*"), the *AFOFRatio* will be very low. If the object has extremely low *ObjectFrequency* (e.g., a misspelling or obscure term), then the *AFOFRatio* could be very high. In the case of average *ObjectFrequency*, the *AFOFRatio* will reflect whether the assertion appears more often than one would normally expect. We chose a middle range ($1 < AFOFRatio \leq 10$) that seemed to generally yield interesting assertions from the *distinguishing* perspective.

[1] http://www.flickr.com/explore/interesting/

[2] We add 1 in the denominator to prevent possible division by 0.

3.3 Basic Assertions

Another type of interesting assertion is *basic* facts. These are definitional assertions that, for example, might be interesting to a person learning about an object. A person learning about Einstein, might look up such facts as "*Einstein was a physicist*" or "*Einstein was born in Ulm, Germany*". Interest in such *basic* assertions is evident in the emphasis on this type of information in dictionaries and encyclopedias. We thus operationalize *basic* assertions by learning a classifier to identify assertions most similar to the *basic* facts that human editors include in Wikipedia infoboxes.

Training such a classifier requires examples of TextRunner assertions likely to reflect infobox knowledge (positive training examples) and assertions unlikely to reflect infobox knowledge (negative training examples). Starting with the DBPedia Wikipedia infobox database [1], we applied a series of filters and isolated a set of 872 notable people with good infobox coverage. Text matching on infobox values (allowing for small edit distance) produced a set of 1,584 TextRunner assertions that reflected knowledge expressed in those infoboxes. This is comparable to how Kylin matches infobox data to statements [13], but our matching is stricter and thus achieves higher precision at lower recall. We then sampled 3,000 TextRunner assertions about the same people that did not match infobox values.

characters, # words, # capital letters, # numeric digits, presence of years, assertion ends on stop word, proper nouns in arguments, argument frequencies in corpus

Table 1. Features used to train the *basic* classifier.

Table 1 lists the features used to train the *basic* classifier. Because we are interested in a generally applicable classifier, we picked simple low-level features likely to generalize across domains. We tested features that were more lexical (e.g. presence of certain keywords or relations), but found they did not generalize. For our classifier, we use Weka's J48 Decision Tree [9] [12].

4. Evaluating Human Ratings of Interesting

To evaluate our *specific*, *distinguishing*, and *basic* models, we used them as the basis of filters that discard TextRunner results that fail to satisfy each model. To assess their quality, we conducted a study to collect human ratings of assertion interestingness.

4.1 Method and Procedure

We first selected a set of ten study query terms, including famous people (*Albert Einstein, Bill Gates, Thomas Edison*), other proper nouns (*Beijing, Brazil, Microsoft, Diet Coke*), improper nouns (*sea lions*), and relationship queries (*invented, destroyed*). This query set is meant to provide a varied sample of the sorts of queries for which TextRunner can provide interesting results. Our analyses are based on the top thirty assertions resulting from each of these queries, approximately the number of results that can be seen at a glance on a single results page. As a baseline for comparison, we use *AssertionFrequency*, which examines the thirty most frequently occurring assertions. We next obtain assertions for our *specific*, *distinguishing*, and *basic* conditions by applying each of our filters in order of assertion frequency, discarding results that fail the filter until we obtain thirty results that satisfy the filter. This section therefore focuses on 1200 assertions (10 queries * 4 conditions * 30 assertions).

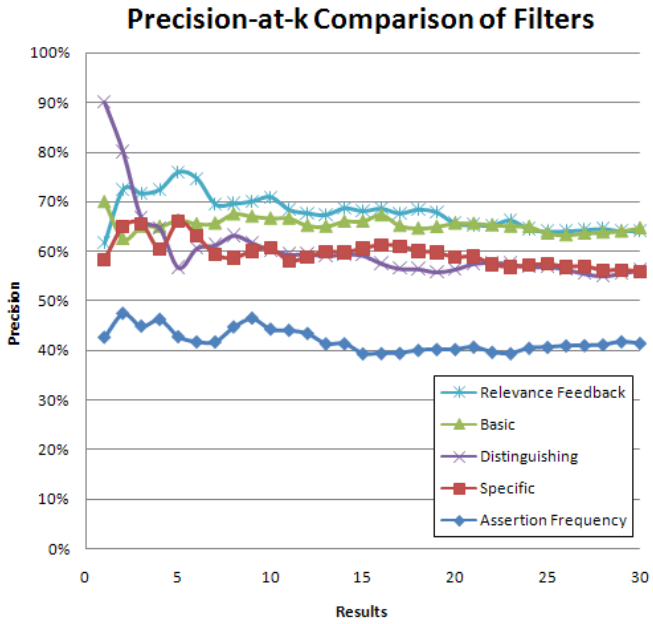

Figure 2. Comparison of performance of our models of interestingness on the top thirty results for our test queries, including an Assertion Frequency baseline.

We recruited 12 study participants (7 female) with a variety of backgrounds including math, marketing, finance, music, and nursing. Participants rated assertions on a scale from 1 (labeled "Least Interesting") to 5 (labeled "Most Interesting"). It would have been onerous to ask participants to rate all 1200 assertions, and so each participant rated 200 assertions. Assertions were presented one at a time, drawn randomly without replacement between participants. Every assertion was therefore rated once before any assertion was rated a second time. We gathered a total of two or three ratings for every assertion, helping to account for individual differences in what people consider interesting.

Defining ratings of *4* or *5* to be *interesting*, ratings of *1* or *2* to be *not interesting*, and discarding ratings of *3*, inter-annotator agreement was 71.1%. This suggests there are assertions that people generally find *interesting* or *not interesting*, but that there is also some variation.

4.2 Relevance Feedback

Given our human labels, we also consider whether a learning-based method, using a classifier to combine information from all three filters, might perform better than any single filter. For features, this classifier used the outputs of the other filters, as well as the features from Table 1. We were careful to evaluate this method using ten-fold cross-validation such that training and testing assertions always came from different queries. Several classifiers we tried had comparable performance, so again we chose to use a Decision Tree classifier [9].

4.3 Results

Defining ratings of *4* or *5* to be *interesting* and *1* or *2* to be *not interesting*, we first compare overall mean average precision for the top thirty results. *AssertionFrequency* had the lowest mean average precision at 41.9% interesting. *Specific* (59.5%) and *distinguishing* (60.3%) were better, *basic* (65.4%) was even better, and *relevance feedback* (67.9%) was the best.

Figure 2 plots *precision at k* for our *specific*, *distinguishing*, *basic*, and *relevance feedback* filters against *AssertionFrequency*. To test for difference between these curves, we conduct an analysis of variance for the precision at each plotted point, treating *condition* and *k* as fixed effects. The omnibus test reveals a significant main effect of *condition* ($F(4, 4) = 285$, $p < .0001$), leading us to investigate pairwise differences. We use Tukey's HSD procedure to account for increased Type I error in unplanned comparisons. This shows *relevance feedback* yields significantly more interesting assertions than *specific* ($F(1,144) = 95.4$, $p < .0001$), *distinguishing* ($F(1,144) = 78.6$, $p < .0001$), *basic* ($F(1,144) = 8$, $p \approx .005$) and *AssertionFrequency* ($F(1,144)=926$, $p <.0001$).

The largest differences in Figure 2 are between our filter-based approaches and TextRunner's original use of *AssertionFrequency*, indicating the advantage of filtering. The classifier filters trained with user labels and user-contributed knowledge (*relevance feedback* and *basic*) performed significantly better than all other approaches, indicating the utility of such data for this task. *AssertionFrequency* achieves a precision at thirty of 41.6%, while *relevance feedback* achieves a precision at thirty of 64.1% (almost comparable to human inter-annotator agreement levels). Additionally, our trained filters achieved higher overall precision at thirty than *AssertionFrequency* in all query term categories we tested (famous people, other proper nouns, improper nouns, and relationship queries).

We primarily examine precision within the top results because Open IE on the Web can generally return many more results than can be read and so the challenge is in precision more than recall. Even if good assertions are filtered out, there are often other assertions that express similar information that pass through. For queries without many results or applications where it is important to not filter out good results, models of interestingness could also be used to rank rather than filter.

5. CONCLUSIONS

Extraction engines such as TextRunner are a promising avenue towards improving Web search and generating large knowledge bases. However, such systems are currently hamstrung by the fact they often return uninformative results that are vague or uninteresting. Web extraction systems are particularly prone to this problem because of the general methods they use to extract entities and relationships [2].

This paper has developed a filter system to enhance interaction with TextRunner by better focusing on *interesting* assertions. As a part of this task, we have presented three models of *interesting* assertions. These have the virtue of being readily operationalized into filters over TextRunner results. In addition to presenting a study of human ratings of the interestingness of assertions, we combined the filters using a relevance feedback technique that raised the average percentage of interesting results on a sample of queries from 41.6% to 64.1%.

There are several exciting avenues of future work here. First, to exceed inter-annotator agreement levels, we could study how different people may find different assertions interesting, and address how a system might learn, represent, and apply personal preferences. Second, carefully examining interesting assertions that did not pass any filters would help to reveal whether there are additional important aspects to *interestingness*. Leveraging resources such as WordNet [6] could provide us with more complex features like semantic similarity of arguments. Lastly, we would like to study the qualities that make *sets* of assertions more interesting to people (e.g., coverage, variety, and redundancy).

Although the problem of improving result ranking is well-studied for search engines, we believe that this work is the first study of *interestingness* for Web extraction systems, and it will serve as a useful baseline for future work.

6. ACKNOWLEDGMENTS

This research was supported in part by NSF grants IIS-0803481 and IIS-0812590, ONR grant N00014-08-1-0431, and Google and was carried out at the University of Washington's Turing Center. The first author was supported in part by an NDSEG Fellowship.

7. REFERENCES

[1] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, Z. Ives. DBPedia: A Nucleus for a Web of Open Data. In *Proceedings of the International Semantic Web Conference (ISWC)*, 2007.

[2] M. Banko and O. Etzioni. The Tradeoffs Between Open and Traditional Relation Extraction. In *Proceedings of the Association for Computational Linguistics (ACL)*, 2008.

[3] M. Banko, M.J. Cafarella, S. Soderland, M. Broadhead, O. Etzioni. Open Information Extraction from the Web. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI)*, 2007.

[4] S. Colton and A. Bundy. On the Notion of Interestingness in Automated Mathematical Discovery. In *Proceedings of the AISB Symposium on AI and Scientific Discovery*, 1999.

[5] D. Downey, O. Etzioni, S. Soderland, A Probabilistic Model of Redundancy in Information Extraction. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI)*, 2005.

[6] C. Fellbaum. *WordNet: An Electronic Lexical Database*. MIT Press, Cambridge, MA, 1988.

[7] G. Kasneci, F. Suchanek, G. Ifrim, M. Ramanath, G. Weikum. NAGA: Searching and Ranking Knowledge. In *Proceedings of the 24th International Conference on Data Engineering (ICDE)*, 2008.

[8] B. Liu, W. Hsu, S. Chen, Y. Ma, Analyzing the Subjective Interestingness of Association Rules. *IEEE Intelligent Systems 15*, 47-55, 2000.

[9] J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufman, 1993.

[10] G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval, *Information Processing & Management*, 24, 1988.

[11] P.J. Silvia, *Exploring the Psychology of Interest*. Oxford University Press, New York, NY, 2006.

[12] I.H. Witten and E. Frank. *Data Mining: Practical machine learning tools and techniques*, 2nd Edition, Morgan Kaufmann, 2005.

[13] F. Wu, and D. Weld. Autonomously Semantifying Wikipedia. In *Proceedings of the 16th Conference on Information and Knowledge Management (CIKM)*, 2007.