Preprint of: Dawn Lawrie, Tim Finin, James Mayfield and Paul McNamee, Comparing and Evaluating Semantic Data Automatically Extracted from Text, AAAI 2013 Fall Symposium on Semantics for Big Data, Arlington, VA, Nov. 2013

Comparing and Evaluating Semantic Data Automatically Extracted from Text

Dawn Lawrie
Computer Science Department
Loyola University Maryland
Baltimore, MD, USA

Tim Finin
University of Maryland
Baltimore County
Baltimore, MD, USA

James Mayfield, Paul McNamee
Johns Hopkins University
Human Language Technology Center of Excellence
Baltimore, MD, USA

Abstract

One way to obtain large amounts of semantic data is to extract facts from the vast quantities of text that is now available online. The relatively low accuracy of current information extraction techniques introduces a need for evaluating the quality of the knowledge bases (KBs) they generate. We frame the problem as comparing KBs generated by different systems from the same documents and show that exploiting provenance leads to more efficient techniques for aligning them and identifying their differences. We describe two tools: entitymatch focuses on differences in entities found and linked; kbdiff focuses on differences in relations among those entities. Together, these tools support assessment of relative KB accuracy by sampling the parts of two KBs that disagree. We explore the usefulness of the tools through the construction of tens of different KBs built from the same 26,000 Washington Post articles and identifying and analyzing the differences.

1 Introduction

One way to obtain large amounts of semantic data is to extract knowledge from on-line text, which is copiously available. Machine reading systems, like NELL (Carlson et al. 2010) and TextRunner (Etzioni et al. 2008), have demonstrated the ability to learn concepts and schema-level knowledge, particular facts about entities and events and the properties and relationships between them. This is a challenging problem that requires unsupervised or self-supervised learning and whose perfection is still a long term basic research goal.

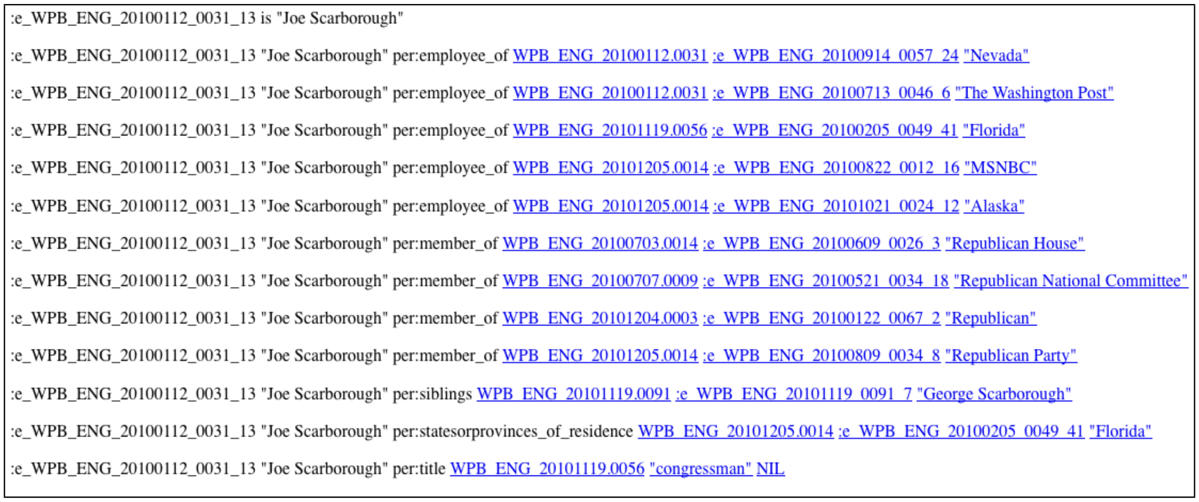
Information extraction (IE) systems, like those developed over the past 25 years for the MUC, ACE and TAC conferences, take a simpler and more pragmatic approach. They start with a knowledge base schema that uses a fixed ontology appropriate for a set of potential applications, and find only those facts that are consistent with the schema to populate the knowledge base (KB). These facts represent the entities discovered (typically people, organizations, places and events) and their properties and the relations between them. While the technology behind these systems has also not yet been perfected, their reliability and practicality are increasing. Properly configured, they can be used today to generate large amounts of potentially useful semantic data.

Moreover, the ontologies underlying such systems can be mapped to equivalent ones in OWL, allowing them to be aligned with other, popular OWL ontologies. The entities found can also be linked to well known entities in common linked open data collections such as DBpedia (Bizer et al. 2009) and Freebase (Bollacker et al. 2008). These connections work both to improve information extraction, providing evidence for entity disambiguation and entity linking (Mayfield et al. 2009) and to enrich the Linked Open Data (LOD) collections with new facts.

Current information extraction systems are far from perfect, often failing to detect entities, missing attested facts, and hallucinating incorrect relations. Additionally, KBs constructed from IE output can both over- and under-merge entities. These diverse sources of error create a need to compare KBs, either an automatically extracted KB with a "ground truth" KB, two KBs extracted by different systems, or the KBs produced by different versions or configurations of the same system. Comparing two independently generated graph KBs, even when they share a fixed ontology, has some practical approaches (Berners-Lee and Connolly 2004; Carroll 2003) but is made very difficult by the potentially exponential problem of aligning nodes that lack globally unique identifiers (e.g., RDF's blank nodes) (Berners-Lee and Connolly 2004; Zeginis, Tzitzikas, and Christophides 2011). For KBs extracted from the same collection of text documents, however, we can capture provenance information and use it for alignment.

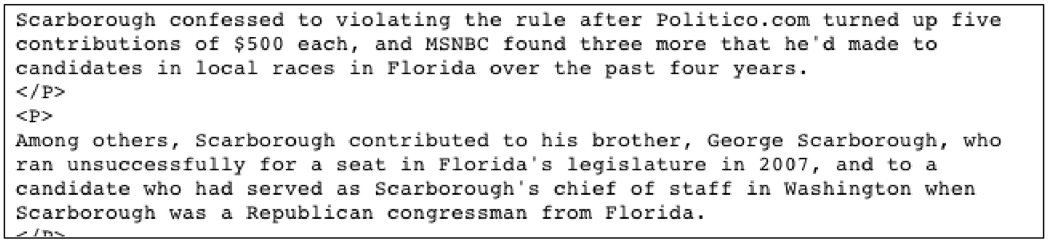
In this paper, we present an approach to KB comparison that we developed as part of our submission to the 2012 TAC Cold Start track (McNamee et al. 2012). The key to this approach is to exploit provenance, which is the link between the entities and the strings that mention them in a document. In the Cold Start scenario, each assertion is tied to strings in a particular document that give evidence for it. These ties can be used to align the entities and relations of two KBs build from the same text, eliminating the exponential cost of alignment.

While we have applied the approach and specific techniques to KBs whose data are extracted from text documents, we believe that same approach can be applied to other use cases as well; examples include extracting semantic data from tables (Mulwad, Finin, and Joshi 2012), social media streams and calendar entries. The basic tasks of entity



```
:e_WPB_ENG_20100112_0031_13 is "Joe Scarborough"
:e_WPB_ENG_20100112_0031_13 "Joe Scarborough" per:employee_of WPB_ENG_20100112.0031 :e_WPB_ENG_20100914_0057_24 "Nevada"
:e_WPB_ENG_20100112_0031_13 "Joe Scarborough" per:employee_of WPB_ENG_20100112.0031 :e_WPB_ENG_20100713_0046_6 "The Washington Post"
:e_WPB_ENG_20100112_0031_13 "Joe Scarborough" per:employee_of WPB_ENG_20101119.0056 :e_WPB_ENG_20100205_0049_41 "Florida"
:e_WPB_ENG_20100112_0031_13 "Joe Scarborough" per:employee_of WPB_ENG_20101205.0014 :e_WPB_ENG_20100822_0012_16 "MSNBC"
:e_WPB_ENG_20100112_0031_13 "Joe Scarborough" per:employee_of WPB_ENG_20101205.0014 :e_WPB_ENG_20101021_0024_12 "Alaska"
:e_WPB_ENG_20100112_0031_13 "Joe Scarborough" per:member_of WPB_ENG_20100703.0014 :e_WPB_ENG_20100609_0026_3 "Republican House"
:e_WPB_ENG_20100112_0031_13 "Joe Scarborough" per:member_of WPB_ENG_20100707.0009 :e_WPB_ENG_20100521_0034_18 "Republican National Committee"
:e_WPB_ENG_20100112_0031_13 "Joe Scarborough" per:member_of WPB_ENG_20101204.0003 :e_WPB_ENG_20100122_0067_2 "Republican"
:e_WPB_ENG_20100112_0031_13 "Joe Scarborough" per:member_of WPB_ENG_20101205.0014 :e_WPB_ENG_20100809_0034_8 "Republican Party"
:e_WPB_ENG_20100112_0031_13 "Joe Scarborough" per:siblings WPB_ENG_20101119.0091 :e_WPB_ENG_20101119_0091_7 "George Scarborough"
:e_WPB_ENG_20100112_0031_13 "Joe Scarborough" per:statesorprovinces_of_residence WPB_ENG_20101205.0014 :e_WPB_ENG_20100205_0049_41 "Florida"
:e_WPB_ENG_20100112_0031_13 "Joe Scarborough" per:title WPB_ENG_20101119.0056 "congressman" NIL
```

Figure 1: Simple rendering of extracted facts about former Florida congressman Joe Scarborough. Many are correct – he lived in and was employed by the State of Florida; he has a brother George; he was a member of the Republican House of Representatives; and, he is employed by MSNBC.



```
Scarborough confessed to violating the rule after Politico.com turned up five
contributions of $500 each, and MSNBC found three more that he'd made to
candidates in local races in Florida over the past four years.
</P>
<P>
Among others, Scarborough contributed to his brother, George Scarborough, who
ran unsuccessfully for a seat in Florida's legislature in 2007, and to a
candidate who had served as Scarborough's chief of staff in Washington when
Scarborough was a Republican congressman from Florida.
</P>
```

Figure 2: Supporting text for some assertions about Mr. Scarborough. Source documents are also viewable by following hyperlinks.

recognition, entity linking or merging, property and relation extraction, and inference are common across many information domains. All data has some provenance and in many cases, knowing the provenance of entity mentions or property or relation assertions can be used to align nodes in two or more semantic graphs. Thus the key insight in this work is the use of provenance. This leads to the research contributions of this paper:

- the ability to compare multiple KBs; and
- a way to evaluate KBs based on the comparison.

The next section of the paper provides background on the TAC Cold Start task. This is followed by a description of the evaluation tools we developed, how they were used in validating the KBs we created, and concluding remarks.

2 Cold Start Task and KB Evaluation

The Text Analysis Conference (TAC) Knowledge Base Population (KBP) Cold Start task (TAC KBP Web site 2012) requires systems to extract from a set of documents a comprehensive set of triples that encode relationships between and attributes of the named entities that are mentioned in the corpus. Systems are evaluated based on the fidelity of the constructed KB. For the 2012 evaluation, a fixed schema of 42 relations (or slots), and their logical inverses was provided. Targeted relations include:

- X:Person is-married-to Y:Person
- X:Organization employs Y:Person
- X:Person has-job-title *title*
- X:Organization headquartered-in Y:Location

Cold Start differs from previous KBP tracks in that it assumed an initial KB containing only a schema with no entities or facts. In knowledge representation terms, the schema is entirely composed of TBOX entries; there are no ABOX entries. Moreover, the documents to be processed are assumed to be largely about entities that are not "famous," so using a strategy of linking entities to those in an existing reference KB (*e.g.,* DBpedia, Freebase) would, in general, be of little help. This places more emphasis on developing strategies for clustering entities as coreferent over linking a new document entity to an existing KB entity about which we already know a great deal.

Multiple layers of NLP software are required for this undertaking, including at the least: detection of named entities, intra-document coreference resolution, relation extraction, and entity disambiguation. The tools created for this task rely on the way in which a KB is defined for the TAC Cold Start task. A Cold Start KB is created from a large collection of documents; currently the collection comprises on the order of 30,000 documents. Entities are identified as people, GPEs, or organizations. An entity will have one or more mentions in the documents. These mentions are tied to the knowledge base node representing that entity; such ties form the provenance for the entity.

Cold Start Evaluation. A central problem in evaluating generated KBs is aligning the entities in the KB with known ground truth. For example, if we had a reference ground truth KB, we could try to evaluate the created KB by aligning the nodes of the two KBs, then looking for structural differences. Aligning entities is a complex task that, in the worst case, can have exponential complexity in the number of entities involved. Cold Start's approach avoids this problem by using known *entry points* into the KB that are defined by a document and an entity mention string (Mayfield and Finin 2012a). For example, an entry point could be defined as "the entity that is associated with the mention *Bart Simpson* in document D014." This requires each entry point to be aligned with a node in the KB by the KB constructor. Doing so is straightforward if the KB is being constructed from the documents that contains the entry point mentions.

A Cold Start submission is evaluated by applying a set of *evaluation queries* to the KB and manually assessing the results. An evaluation query starts at an entry point, finds the corresponding entity in the KB (if it exists), then traverses zero or more relations. A gloss of such a query might be "find all schools that the siblings of the 'Bart Simpson' mentioned in Document 42 attended." All entities that satisfied the query would be assessed by a human judge to determine whether they were supported by the documents cited. Cold Start uses a simplified graph path notation for evaluation queries to make constructing them easier; this notation is then automatically compiled into corresponding SPARQL-like queries. For example, one pattern starts with an entry point (a mention in a document) and continues with a sequence of properties. The general form of such a path expression is $MDP_1...P_n$ where M is a mention string, D is a document identifier, and each P_i is a property from the target ontology. All of the properties in the path except the final one must go from entities to entities. The final one can have a range that is either an entity or a string. For example, to generate a query for *"The ages of the siblings of the entity mentioned as "Bart Simpson" in document D012"* the path expression `"Bart Simpson" D012 sibling age` was used.

A SPARQL query generated for this path expression is shown in Figure 3; when run against a submitted KB, it produces data allowing the assessor to verify that the KB accurately reflects the supported facts:

sibling mention	sib doc	age	age doc
”Lisa Simpson”	D012	”10”	D008
”Maggie Simpson”	D014	”1”	D014

In general, for each entity in the result, a query produces the canonical mention string for that entity in the supporting document (e.g., support for "Lisa Simpson" as Bart's sister is in D012), while for each slot value (e.g., age:10), the query produces the value (10) and the document providing evidence for it (D008). This enables verification that the correct entities are identified and that slot values have explicit support.

```
SELECT ?CN ?SIBDOC ?A ?ADOC WHERE {
  ?P kbp:mention "Bart Simpson".
  ?P kbp:sibling ?SIB.
  ?SIB kbp:canonical_mention ?CN; kbp:age ?A.
  _:x rdf:subject ?P; rdf:predicate kbp:mention; rdf:object "Bart Simpson";
    kbp:source doc:D12.
  _:x rdf:subject ?P; rdf:predicate kbp:sibling; rdf:object ?SIB;
    kbp:source doc:SIBDOC.
  _:x rdf:subject ?SIB; rdf:predicate kbp:canonical_mention; rdf:object ?CN;
kbp:source doc:SIBDOC.
  _:x rdf:subject ?SIB; rdf:predicate kbp:age; rdf:object ?A; kbp:source
doc:ADOC.}
```

Figure 3: This SPARQL query generates data that an assessor can use to evaluate the KB.

Different classes of evaluation queries can assess different capabilities. For example, asking whether two entry points refer to same KB node evaluates coreference resolution. Asking facts about the KB node associated with a single entry point evaluates simple slot-filling. More complicated queries that start with one or more entry points can be used to evaluate the overall result of the extraction process involving entity linking, fact extraction, appropriate prior,s and inference. Note that this approach to KB evaluation is agnostic toward inference. That is, the original KB system may perform sophisticated backward chaining inference or no inference at all; the evaluation mechanism works the same either way.

Design-based Evaluations. While the 2012 Cold Start evaluation methodology is able to compare the output of two systems and say which performed better, it fails to provide direct insight into why one was better than the other. For example, was a system's poor performance due to under merging entities or over-merging them, or perhaps to a single catastrophic decision to merge two important entities? Does one system find more evidence for entity relations than another? How much does drawing plausible inferences about probable relations help in entity matching? Answering such questions requires more elaborate evaluation tools than the simple sample query approach used by the Cold Start evaluation. We describe our first attempt to create such tools in the next section.

3 Approach

We developed two tools to compare knowledge bases, both of which exploit the alignment-by-mentions technique (*i.e.*, provenance). The first, entity-match, identifies differences in entities; the second, kbdiff, identifies differences in relations. Used together, these tools can identify where two KBs differ and why they differ. They can also be used to determine which of two related KBs is more accurate without a full Cold Start evaluation.

Slotname	*Count*
per:employee_of	60,690
org:employees	44,663
gpe:employees	16,027
per:member_of	14,613
org:membership	14,613
org:city_of_headquarters	12,598
gpe:headquarters_in_city	12,598
org:parents	6,526
org:country_of_headquarters	4,503
gpe:headquarters_in_country	4,503

Slotname	*Count*
per:title	44,896
per:employee_of	39,101
per:member_of	20,735
per:countries_of_residence	8,192
per:origin	4,187
per:statesorprovinces_of_residence	3,376
per:cities_of_residence	3,376
per:country_of_birth	1,577
per:age	1,233
per:spouse	1,057

Figure 4: Most frequently occurring slots extracted by SERIF (left) and FACETS (right) from the 26,000 Washington Post articles.

3.1 Detecting Entity Differences: Entity-Match

The entity-match tool defines an entity in a KB as the set of mentions that refer to the same entity node. From the perspective of an entity in one KB, its mentions might be found within a single entity in the other KB, spread among multiple entities, or missing altogether from the other KB. In the first case there is agreement on the what makes up the entity. In the second case, there is evidence either that multiple entities have been conflated in the first KB, or that a single entity has been incorrectly split in the second. In the third case, the entity has gone undetected.

The algorithm assumes that all named references are associated with an entity id. If two named references have the same entity id, then the named references refer to the same entity. To begin the named references of all the entities in the first KB are established. Then when reading the named references of the second KB, the id of the entity for that named reference in the first KB is established by an exact match of provenance (overlapping named references will not be treated as a match). The entity id from the second KB is associated with the aligned entity in the first KB. Finally, for each entity id in the KB, the number of corresponding ids in the second KB establish what case the entity id falls into. A set of size one indicates the first case of agreement. A set containing more than one id indicates the second case of non-agreement. Finally, the empty set indicates the third case of a missing entity in the second KB.

The tool reports for each entity in the KB which case it falls into. If there is disagreement between the KBs, it reports each corresponding entity in the second KB and the number of mentions that map to that entity.

The running time of this algorithm is $O(n^2)$ in the number of entities. Given two KBs, each with approximately 6 million assertions, it takes entity-match about 2 and a half minutes to produce the report. The only barrier to scaling to much larger KBs is that the current implement assumes that both KBs can reside in memory together; however, re-engineering could overcome this limitation.

3.2 Detecting Relation Differences: Kbdiff

The kbdiff tool operates at the level of individual assertions. It works in a similar fashion to the standard Unix utility diff (Hunt and Mcillroy 1976) by identifying assertions in one KB that do not appear in the other. The challenge of this task is to identify which entities are held in common between the two KBs. Provenance is again useful here. Two KBs assert the same relationship if the predicates match, and the subject and object have identical provenance.

The algorithm works by first reading all the assertions in both KBs. Assertions are matched based on provenance and type. Then the assertions in the first KB are iterated over. If there is an assertion from the first KB that does not match an assertion from the second KB, that assertion is part of the output and is preceded by a “$<$”. Then the assertions in the second KB are iterated over. If there is an assertion from the second KB that does not match an assertion from the first KB, that assertion is part of the output and is preceded by a “$>$”.

The running time of this algorithm is $O(n^2)$ in the number of assertions. Given two KBs, each with approximately 6 million assertions, it takes kbdiff about 7 and a half minutes to produce its report. Again, the only barrier to scaling to much larger KBs is that the current implement assumes that both KBs can reside in memory together.

4 Validation

To assess the usefulness of these tools, tens of different KBs were constructed over the same text collection. Rather than using different processes to build the KBs, this study generated variability by removing text from the documents that did not meet certain criteria. To keep the provenance information the same, text was replaced by a string of spaces when it was removed.

In the following sections we first briefly describe the system used to the generate the data, then present the experimental setup, and lastly discuss the results that can be extracted from the tools.

4.1 KELVIN

The KELVIN system (McNamee et al. 2013) operates as a pipeline that integrates a number of tools required to perform the Cold Start task. BBN’s SERIF tool[1] (Boschee, Weischedel, and Zamanian 2005) provides a considerable suite of document annotations that are a strong basis for building a TAC KB. The functions SERIF provides are based

[1] Statistical Entity & Relation Information Finding.

largely on the NIST ACE specification,[2] and include: (a) identifying named entities and classifying them by type and subtype; (b) performing intra-document coreference analysis, including named mentions, as well as coreferential nominal and pronominal mentions; (c) parsing sentences and extracting intra-sentential relations between entities; and, (d) detecting certain types of events. SERIF recognizes, but does not normalize, temporal expressions, so we used the Stanford SUTime package to normalize date values.

SERIF output is augmented by FACETS, another BBN tool. FACETS adds role and argument annotations derived from person noun phrases that include relative clauses and appositives to SERIF output. FACETS is implemented using a conditional-exponential learner trained on broadcast news. The attributes FACETS can recognize include general attributes like religion and age (which anyone might have), as well as role-specific attributes, such as medical specialty for physicians, or academic institution for someone associated with an university. FACETS can independently extract some slots that SERIF is also capable of discovering (*e.g.,* employment relations). The tables in Figure 4 show the most common slots SERIF and FACETS extracted from the Washington Post articles.

We used a simple approach to entity conference resolution in our initial experiments. Under the theory that name ambiguity may not be a huge problem, entities are merged across different documents if their primary mentions were an exact string match after some basic normalizations such as removal of punctuation and conversion to lower-case characters.

We performed a small amount of light inference to fill some slots. For example, if we identified that a person P worked for organization O, and we also extracted a job title T for P, and if T matched a set of titles such as *president* or *minister* we asserted that the tuple <O, org:top_members_employees, P> relation also held.

4.2 Experimental Setup

For our evaluation, we created a series of KBs based on documents that were partially ablated, comparing each to a KB that was created from the full text of the documents. This approach is based on the observation that component systems tend to get confused when sentence structure is complicated. If we were simply to remove the most complicated sentences from the document collection, we might see a performance improvement. The underlying collection of documents comprises 26,000 documents from the Washington Post. All the variants are based on attributes of the sentences within the documents. The attributes we selected to mirror sentence complexity include the number of identified names in the sentence, the number of tokens in the sentence, the number of commas, and the number of tokens that are punctuation characters. For each KB constructed from ablated text, a sentence was ablated if it exceeded a maximum number of a selected attribute. Documents were only ablated using one criterion at a time.

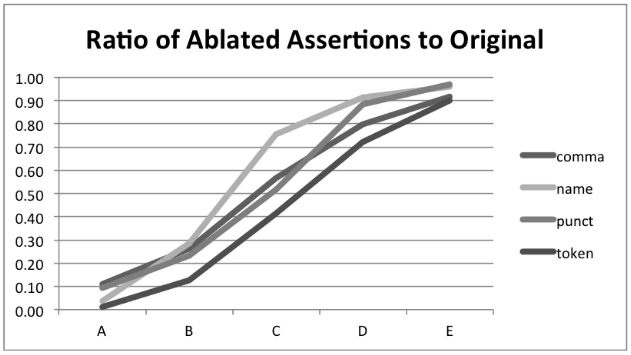


Figure 5: Ratio of assertions in an ablated KB relative to the original KB. Each line represents the different thresholds used for a particular attribute.

Different thresholds were chosen for each of the attributes:

- Name: A=1, B=2, C=4, D=6, E=8
- Tokens: A=10, B=20, C=30, D=40, E=50
- Commas: A=0, B=1, C=2, D=3, E=4
- Punctuation: A=1, B=2, C=3, D=5, E=7

Figure 5 depicts the ratio of the number of assertions made in an ablated KB and the original. It provenance shows how the different thresholds affect the number of assertions. The goal in choosing the thresholds was to span the spectrum of few assertions to many assertions. Then the tools introduced in the prior section could be used to ascertain whether the type of attribute had any impact on the quality of resulting KBs.

5 Experimental Results

The entity-match tool reveals how the entities in one KB align to entities in another. When entities map to multiple entities, they generally match to just over two entities on average. This is consistent over all the ablations. Figure 6 shows that the more heavily ablated text are more consistent with the original KB. This is not surprising since the most ablated KBs have many fewer entities. These missing entities are by definition consistent across the two KBs. It is interesting to note the difference in consistency decay: token based ablation has the highest consistency, while name-based ablation has the lowest consistency with the original. Finally, when considering the entities' mention sets, only 31% of the ablated entities are proper subsets of the original entities. This indicates that ablation has a big impact on entity linking. Unfortunately, determining whether the ablation is revealing entities that should not have been merged or missing entities that should have been merged requires human judgments.

From a brief inspection of the mentions, about 86% of entities in the original KB appear to be a single entity. In the instances where the original KB combined multiple entities, the ablation was able to separate the entities 43% of the time.

[2] The principal types of ACE named entities are persons, organizations, and geo-political entities (GPEs), see (NIST and ACE 2007).

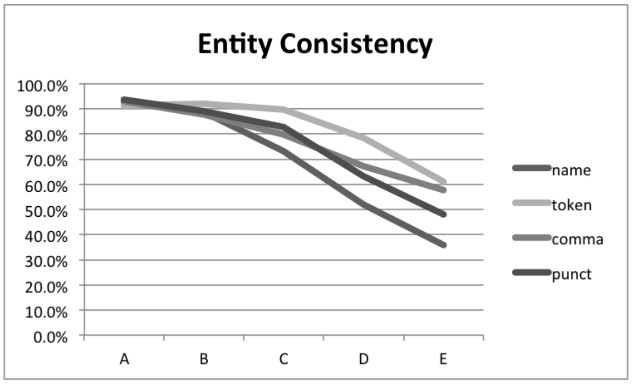


Figure 6: Percentage of entities in original KB that are not divided among multiple entities in the ablated KB.

Considering the case of a single mention chain in the original knowledge base, 15% of the ablated entities conflated multiple entities. This indicates that ablation can affect the cohesion of entities.

The kbdiff tool easily supports gaining an understanding of the impact of a change on the resulting KB in terms of the relations asserted. This includes the number of assertions that are in one KB and not in another and the number of assertions that have changed. Figure 7 shows that quantity of change peaks in the middle of the range of ablation. This behavior makes intuitive sense given that roughly half the assertions are present by volume. This indicates that sufficient data is available to find many assertions over the same data. The question is which set of assertions is better.

Turning to the assertions that are in the original KB but not in the ablated version, it is no surprise that Figure 8 shows that this occurs predominately in the most ablated KBs. This is due to the fact that there are simply not many assertions in the the heavily ablated KBs. When comparing this figure to Figure 5 where the number of assertions is counted, the figures are almost exact inverses of each other. On average the difference between the ratio of assertions and the inverse of the percentage of assertions only in the original KB is 2%.

Finally, considering the assertions that are only in the ablated KB relative to the original KB, Figure 9 reveals that the greatest proportion of new assertions comes from the most ablated KBs. This indicates that some information may be obscured by the presence of more complicated text (although it also could be that the system as a whole is less accurate when so little information is present).

To get a sense of the impact ablation has on accuracy, we made about twenty judgments on the assertions for each KB relative to the original non-ablated one. Assertions were binned into the same three categories outlined above: those representing a change from the original to the ablated KB, those only in the original, and those only in the ablated one. Figure 10 shows the accuracy differences between an ablated and original KB.

In total we made 414 judgments over the KB assertions that were part of the difference set of an ablated knowledge base and its original. By using all the judgments that are present in a knowledge base, most KBs have over one hundred judgments. Figure 11 shows that eliminating sentences based on the number of tokens may have a positive impact on accuracy. The other noteworthy conclusion is the difficulty of processing sentences that include only a single name. To have a relation, a reference to a second entity must be present in the sentence. With so few sentences in ablated text, the correct reference to the second entity is often missing, leading to particularly poor performance.

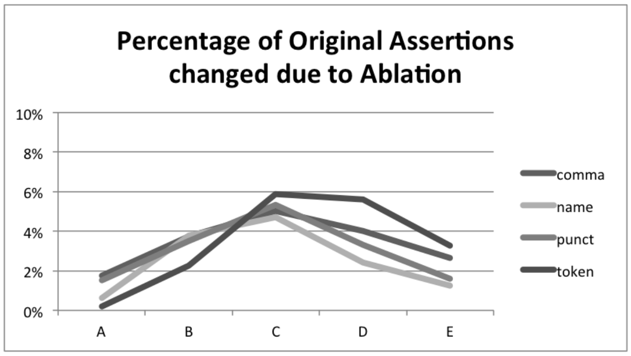


Figure 7: Ratio of assertions that were modified in the ablated KB relative to the original. Each line represents the different thresholds used for a particular attribute.

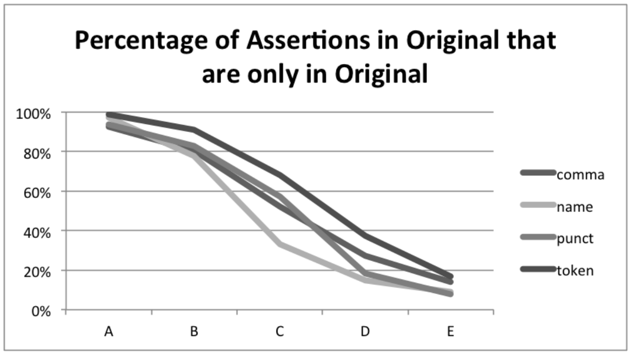


Figure 8: Ratio of assertions only in the original KB relative to the original. Each line represents the different thresholds used for a particular attribute.

6 Conclusions and Ongoing Work

Automatically extracting information from text and expressing it as RDF linked data is a promising way to generate and augment large semantic KBs. Assessing the quality of the knowledge produced is essential for evaluating the utility of such systems and also extremely useful as part of their development methodology. We described an approach to comparison and initial versions of two general systems, kbdiff and entity-match, that we developed to support our work on knowledge base population. Both tools are based on the idea that we can exploit document provenance in compar-

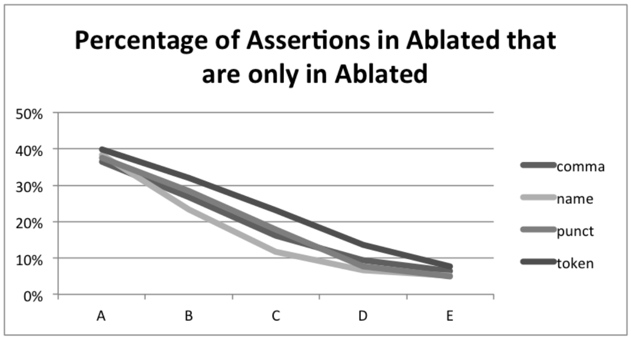


Figure 9: Ratio of assertions only in an ablated KB relative to the original. Each line represents the different thresholds used for a particular attribute.

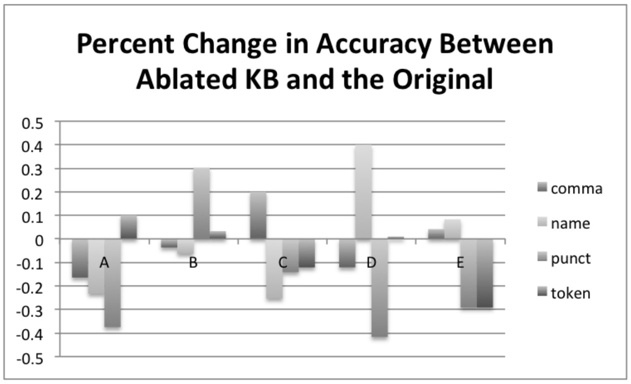


Figure 10: The difference in accuracy of the ablated KB relative to the original one. A positive difference occurs when the ablated KB has higher accuracy.

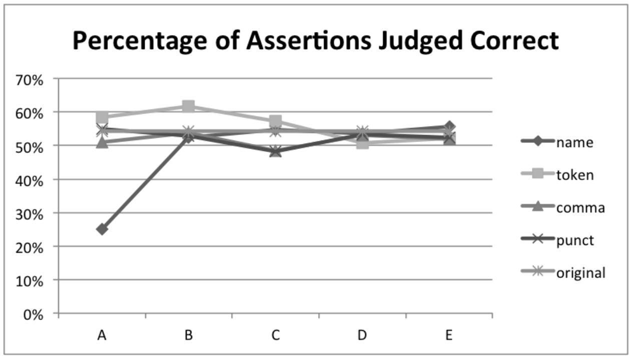


Figure 11: The accuracy of each KB using all judgments on assertions present.

ing two KBs that are generated from a common set of text documents.

We have developed an OWL ontology (Mayfield and Finin 2012b) that corresponds to the knowledge base scheme used in the 2012 and 2013 TAC KBP Cold Start tasks, and wrote simple programs to generate an RDF graph from the Cold Start submissions format. The submission format's inclusion of provenance data and certainty metrics requires the use of reification for RDF encodings and introduces some overhead in storage and search in most triple stores. While these issues may be important in developing a production system intended to process large volumes of text and generate huge KBs, they are less problematic in an evaluation context where speed and scaling are not a focus. We found the RDF linked data versions generated by our 2013 Cold Start system very useful for exploration and analysis via Pubby (Cyganiak and Bizer 2007) and SPARQL queries. We plan to experiment with the linked data version of the extracted knowledge to support augmentation and integration with other linked data resources and as a store for a large-scale streaming version of our system. The tools presented in this paper will thus help us evaluate the quality of RDF KBs.

7 Acknowledgments

Partial support for this work was provided by NSF grant CCF 0916081 and AFOSR grant FA9550-08-1-0265.

References

Berners-Lee, T., and Connolly, D. 2004. Delta: an ontology for the distribution of differences between rdf graphs. http://www.w3.org/DesignIssues/Diff (2006-05-12).

Bizer, C.; Lehmann, J.; Kobilarov, G.; Auer, S.; Becker, C.; Cyganiak, R.; and Hellmann, S. 2009. Dbpedia - a crystallization point for the web of data. *Journal of Web Semantics* 7(3):154–165.

Bollacker, K.; Evans, C.; Paritosh, P.; Sturge, T.; and Taylor, J. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proc. ACM Int. Conf. on Management of Data*, 1247–1250. ACM.

Boschee, E.; Weischedel, R.; and Zamanian, A. 2005. Automatic information extraction. In *Proc. Int. Conf. on Intelligence Analysis, McLean, VA*, 2–4.

Carlson, A.; Betteridge, J.; Kisiel, B.; Settles, B.; Jr., E. R. H.; and Mitchell, T. M. 2010. Toward an architecture for never-ending language learning. In *Proc. 24th Conf. on Artificial Intelligence*.

Carroll, J. J. 2003. Signing rdf graphs. In *The Semantic Web-ISWC 2003*. Springer. 369–384.

Cyganiak, R., and Bizer, C. 2007. Pubby: a linked data frontend for SPARQL endpoints. http://wifo5-03.informatik.-uni-mannheim.de/pubby/.

Etzioni, O.; Banko, M.; Soderland, S.; and Weld, D. S. 2008. Open information extraction from the web. *Commun. ACM* 51(12):68–74.

Hunt, J. W., and Mcillroy, M. D. 1976. An algorithm for differential file comparison. *Communications of The ACM*.

Mayfield, J., and Finin, T. 2012a. Evaluating the Quality of a Knowledge Base Populated from Text. In *Joint Workshop on Automatic Knowledge Base Construction and Web-scale*

Knowledge Extraction. Association for Computational Linguistics. held in conjunction with 2012 NAACL-HLT.

Mayfield, J., and Finin, T. 2012b. TAC KBP Ontology in OWL. http://ebiq.org/o/tackbp/2012/tackbp.owl.

Mayfield, J.; Alexander, D.; Dorr, B.; Eisner, J.; Elsayed, T.; Finin, T.; Fink, C.; Freedman, M.; Garera, N.; Mayfield, J.; McNamee, P.; Mohammad, S.; Oard, D.; Piatko, C.; Sayeed, A.; Syed, Z.; and Weischedel, R. 2009. Cross-Document Coreference Resolution: A Key Technology for Learning by Reading. In *Proc. AAAI Spring Symposium on Learning by Reading and Learning to Read*. AAAI Press.

McNamee, P.; veselin Stoyanov; Mayfield, J.; Finin, T.; Oates, T.; Xu, T.; Oard, D.; and Lawrie, D. 2012. HLT-COE Participation at TAC 2012: Entity Linking and Cold Start Knowledge Base Construction. In *Proc. 5th Text Analysis Conference*. NIST.

McNamee, P.; Mayfield, J.; Finin, T.; Oates, T.; Lawrie, D.; Xu, T.; and Oard, D. 2013. KELVIN: a tool for automated knowledge base construction. In *Proc. NAACL-HLT*. Association for Computational Linguistics. (demo. paper).

Mulwad, V.; Finin, T.; and Joshi, A. 2012. A Domain Independent Framework for Extracting Linked Semantic Data from Tables. In *Search Computing - Broadening Web Search*. Springer. 16–33. LNCS volume 7538.

NIST, and ACE. 2007. Automatic content extraction 2008 evaluation plan – assessment of detection and recognition of entities and relations within and across documents. Technical report, NIST. http://bit.ly/aceEval.

TAC KBP Web site. 2012. Cold start knowledge base population at TAC 2012 task description. http://www.nist.gov/-tac/2012/KBP/ColdStart/. National Institute of Standards and Technology.

Zeginis, D.; Tzitzikas, Y.; and Christophides, V. 2011. On computing deltas of RDF/S knowledge bases. *ACM Trans. Web* 5(3):14:1–14:36.